

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственно правильный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматизации».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственно правильный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «средняков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
[2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственно правильный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствующая на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо

На основе предложенной методики для преобразователя с перестраиваемым преселектором разработаны программные средства оптимизации параметров частотного распределения «Расчет и оптимизация параметров частотного распределения преобразователя с перестраиваемым преселектором» (FAM with ITF. Frequency Allocation Mixer with Input Tunable Filter. Адрес разработчика: E-mail: Loginov@aquasci-nnov.ru). Применение идеализированной модели фильтрующих элементов позволяет получить предельные параметры частотного распределения.

Программа FAM with ITF работает в двух режимах: расчета и оптимизации параметров частотного распределения. Основное окно используется для отображения результатов расчета в абсолютном масштабе частотной области на трех уровнях: частотное распределение входных сигналов, промежуточная частота и частотное распределение ближайших четырех комбинационных составляющих к полосе выходного фильтра. Второе окно номограммы комбинационных частот отображает область фильтруемых частот в относительном частотном масштабе на номограмме комбинационных частот.

Список литературы

- [1] Шарапов Ю.И., Крылов Г.М., Пантелеев Ю.П. Преобразование сигнала без комбинационных частот. – М.: ИПРЖР, 2001. – 288 с.
- [2] Логинов В.И., Маркова С.А. Номограмма комбинационных частот – алгоритмический подход. – ж. Радиотехника. – 1989, №1. С. 44–46.

Т.И. Гаврилова, Р.Н. Гурьяшова
ФБОУ ВПО «ВГАВТ»

ИЗУЧЕНИЕ ОСНОВ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ СИ В РАМКАХ ДИСЦИПЛИНЫ «ИНФОРМАТИКА» ДЛЯ ИНЖЕНЕРНЫХ СПЕЦИАЛЬНОСТЕЙ

Рассматриваются проблемы преподавания языка программирования Си. Обсуждаются трудности освоения языка в условиях ограниченного количества лекционных занятий. Предлагаются пути решения ряда проблем.

Язык программирования Си – основа таких популярных на сегодняшний день языков как Си++ и Си#. Эти языки широко востребованы в настоящее время и используются для разработки современных программных средств как системного, так и прикладного характера. Язык Си преподается на младших курсах инженерных специальностей электромеханического факультета совсем недавно: два года для студентов специальности «Техническая эксплуатация транспортного радиооборудования», и впервые в этом году – для студентов, обучающихся по специальности «Эксплуатация судового электрооборудования и средств автоматики».

Цель курса – приобретение студентами умения алгоритмически мыслить, самостоятельно ориентироваться в среде разработки Visual Studio 2008, а также опыта написания простейших пользовательских программ. Эти навыки будут в дальнейшем использоваться студентами при изучении таких дисциплин, как «ЭВМ в системах управления», «Микропроцессорные системы управления» и ряда других.

Новые учебные планы электромехаников предусматривают изучение дисциплины «Информатика» в течение двух семестров. При этом на весь курс запланировано всего 26 часов лекций, что меньше, чем количество лекционных часов ранее приходившихся только на один семестр (34 часа). В то же время, количество лабораторных работ в

одном семестре осталось примерно прежним (30 часов против бывших ранее 34), а в следующем увеличено до 4 часов в неделю.

В этих условиях лекционный процесс приходится либо вести в режиме «галопом по Европам», либо посвящать лекции базовым вопросам, объясняя многие сложные вещи «на пальцах». Становится чрезвычайно важно своевременно выявить, насколько студенческая аудитория усвоила лекционный материал. Это можно легко сделать, организовав тестирование по пройденной теме. При этом важно, чтобы тесты были составлены качественно. На большую часть вопросов студент должен дать единственный самостоятельный ответ, что исключает возможность угадать из нескольких вариантов-подсказок. Если такие опросы проводить регулярно, аудитория либо начнет готовиться к занятиям, и мы получим пример эффективно организованного учебного процесса, либо выберет тактику «избегания», то есть начнет прогуливать лабораторные работы, последствием чего станет рост количества задолженностей.

Такая жесткая система способствует выявлению действительно работоспособных, талантливых студентов, которые, безусловно, достойны получить диплом специалиста. Но, в то же время, мы рискуем потерять на младших курсах «среднячков», не вписавшихся в интенсивный процесс обучения, и «слабаков», не приученных напрягаться еще со школьных времен.

Какие еще сложности создает преподавателю и студенту такая ситуация? Анализ результатов работы по-новому показал, что положение с успеваемостью кардинально не изменилось, однако выявился ряд проблем.

Весьма небольшое количество лекций способствует тому, что студенты, имеющие мотивацию учиться, стараются не пропускать их и использовать полученную информацию на практике максимально эффективно. Гораздо тяжелее ситуация со студентами, для которых учебный процесс в силу ряда причин не входит в круг первоочередных интересов и задач. Отсутствовавшие на лекциях дезорганизуют работу остальных членов учебной группы, заимствуя у них тетради с конспектами и лишая тем самым возможности эффективно работать на лабораторных занятиях. Также хронической становится ситуация, когда студенты не приносят конспекты лекций на лабораторные работы. Чтобы преподаватель не был безоружен в этом случае, необходимо, чтобы в локальной сети в свободном доступе была электронная версия лекционного материала. При этом удается избежать неэффективного расходования учебного времени. Также дисциплинирует студентов регулярный контроль посещаемости на лекциях.

Другой проблемой, как показала практика, является достаточно высокая сложность изучаемого языка программирования. В сети Интернет не так давно развернулась широкая полемика по вопросу, можно ли язык Си использовать как первый язык для изучения основ программирования. Мы готовы согласиться с тем, что лучше изучать язык Си, имея уже достаточный опыт программирования на таких алгоритмических языках как Паскаль или Бейсик. К сожалению, далеко не все студенты такой опыт имеют.

Структурно процесс обучения разделяется на два этапа: освоение типовых алгоритмических конструкций и программирование на языке Си. Первому этапу отводится примерно полторы – две лекции, причем уже на второй лекции приходится показывать первый пример написания программы. Такое построение лекций является осознанной необходимостью, так как к моменту прочтения третьей лекции проходит уже целый месяц, то есть четыре лабораторных занятия, на которых, кроме составления схем алгоритмов, студенты уже успевают познакомиться со средой программирования и жаждут получить первые результаты.

Теперь о трудностях освоения языка. Во-первых, основной структурной единицей при программировании на языке Си, в отличие от ряда других языков, является функция. Таким образом, буквально на первой лекции приходится объяснять, что это такое, показывать структуру простейшей программы, вводить понятие «директивы препроцессора», говорить о тонкостях описания типов функций и переменных, о воз-

возможностях вывода результатов на экран, рассказывая о спецификаторах формата, что является совсем неочевидным для первого занятия моментом. И все это желательно совместить с понятием разветвленного алгоритма и демонстрацией решения одного или нескольких примеров на эту тему.

Любой преподаватель информатики скажет, что доходчиво объяснить все это в рамках одной лекции практически невозможно, если пытаться «разжевать» рассматриваемый материал по общепринятым канонам. Но мы поставлены в условия, когда это необходимо сделать. Далее кратко рассматриваются некоторые библиотеки стандартных функций языка, в частности `stdio` и `math`, без чего объяснение следующих примеров становится затруднительным, поясняется назначение функции `getch()` библиотеки `conio`.

Безусловно, перед тем, как обучаемые приступят к решению задач в среде Visual Studio 2008, требуется объяснить, что такое процесс компиляции, компоновка программы, какие промежуточные файлы при этом получаются, прежде чем будет получена выполняемая программа. После этого студенты могут приступить к лабораторным работам, следуя призыву преподавателя «делай, как я».

На практике у отдельных студентов возникают трудности с использованием операторных скобок. Стандартной является ошибка, когда требуется сравнить два значения на равенство и вместо используемого в Си двойного знака равенства (`a == b`) указывается более привычный один знак (`a = b`).

Безусловно, наибольшую трудность в понимании вызывают понятие указателя, операция взятия адреса и разыменования. Не объяснять их нельзя, так как основная изюминка языка заключается в их использовании и ряд моментов, которые обязательно должны быть доведены до сведения аудитории, требуют понимания этих вопросов. Таким образом, на лекционных занятиях, проводимых, как уже было сказано, в цейтноте, приходится «на картинках» быстро пояснять идею. И уже в ходе лабораторных работ требовать от студентов осознания этих сложных вещей, давая им задания на использование в решении задачи нескольких функций, кроме `main()`, что требует осмысления технологии передачи данных. Очень полезно эту тему проработать на лабораторных занятиях с помощью тестов.

При решении задач на массивы считаем целесообразным лишь упомянуть, что они не являются в Си самостоятельным типом данных. К сожалению, в столь жестко ограниченных временных рамках не представляется возможным познакомить студентов с мощным аппаратом адресной арифметики, используемым в Си для работы с массивами. Но, так как профессиональное программирование не входит в сферу профессиональных компетенций студентов-электромехаников, достаточно ограничиться стандартными приемами работы с массивами, благо, язык Си позволяет сделать это.

Наиболее интересной, по мнению самих студентов, является тема «работа с символьными строками». Здесь раскрываются особенности Си, которые называются нестрогой типизацией. Студенты с удивлением открывают для себя тот факт, что в Си символ и его код суть одно и то же. Также они, наконец, начинают понимать, что логические значения являются, оказывается, константами целого типа. Такое проникновение в тонкости языка стимулирует студентов к более глубокой самостоятельной работе, и лучшие из них с удовольствием решают задачи на эту тему.

Цикл лабораторных работ включает в себя перечень следующих заданий: задачи на разветвления, табулирование функции, два задания на обработку одномерных массивов, двумерные массивы, причем одна из задач выдается как зачетная, а также задачи на обработку символьных строк и на структуры. Лабораторный практикум предусматривает написание схемы алгоритма решения задачи, осознания студентом, с какими данными какого типа предстоит работать программе, что является в Си достаточно тонким вопросом, и дальнейшее получение работающей программы.

Для получения зачета по дисциплине требуется выполнить все лабораторные работы, представить отчет, содержащий постановку задачи, список имен переменных,

используемых в задаче с пояснением их смысла, схему алгоритма решения задачи, листинг программы, распечатанный результат. Также требуется решить зачетную задачу и выполнить мини-тест на понимание кода, написанного на языке Си. Под тестом имеется в виду фрагмент программы, который тестируемый студент должен проанализировать, демонстрируя понимание порядка действий, и указать, какой должен получиться результат и почему. Успешное прохождение этого теста свидетельствует о глубоком усвоении материала, что является предпосылкой успешного освоения дисциплин, так или иначе связанных с необходимостью составления программ.

Г.А. Гора
ФБОУ ВПО «ВГАВТ»

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩЕЙ СИСТЕМЫ НА СУДНЕ С УЧЕТОМ ПРИОРИТЕТА УПРАВЛЯЕМЫХ ОБЪЕКТОВ

Аннотация: Рассматриваются вопросы создания многоуровневой локальной сети для судовой информационно-управляющей системы (СЛВС).

В настоящее время локальные вычислительные сети (ЛВС) являются обязательными компонентами информационной инфраструктуры любого крупного предприятия.

При проектировании информационно-управляющей системы на судне решают следующие задачи:

- 1) информационное обеспечение систем управления;
- 2) анализ и выбор методов синтеза математических моделей управления комплексом;
- 3) разработка и обоснование алгоритма управления с использованием методов «мягких» вычислений, включающих в себя теорию нечетких множеств, искусственных нейронных сетей и генетических алгоритмов.

В работе рассмотрена одна из задач проблемы управления многосвязными объектами.

При построении информационно-управляющей сети предлагается использовать многоуровневую архитектуру:

- сеть разделяется на несколько уровней, каждый уровень выполняет определенные функции;
- уровни строятся на основе модулей, каждый модуль представляет собой функционально законченную единицу, выполняющую функции соответственно своего уровня.

Проблема создания СЛВС имеет свои особенности, и, соответственно, проблемы, связанные с передачей не только информационных пакетов, но и управляющих команд, например, на перекоммутацию энергопотоков.

Информационные пакеты в данном случае будут носить только рекомендательный характер, пакеты с управляющими командами должны быть недоступными. Создание серверного места (в рулевой рубке) предлагается создать с использованием концентраторов, что создает возможность дальнейшего подключения новых объектов.

При создании судовой СЛВС будем учитывать приоритет управляемых объектов. Разделим судовые управляемые объекты на три категории: объекты высокого приоритета, среднего и низкого.

Принцип подключения устройств управления и наблюдения, для обеспечения постоянной видимости и управляемости объектов энергосистемы судна, необходимо