

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. В.Ф. УТКИНА»

КАФЕДРА СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ  
ВЫЧИСЛИТЕЛЬНЫХ СРЕДСТВ

## **ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

по дисциплине

**Б1.В.01 «Встроенные компьютерные системы электронных средств »**

Направление подготовки

11.04.03 Конструирование и технология электронных средств

Направленность подготовки магистратуры

«Конструирование и технология электронно-вычислительных средств»

Квалификация (степень) выпускника — магистр

Форма обучения — очная, очно-заочная

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной профессиональной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной профессиональной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общекультурных, общепрофессиональных и профессиональных компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости проводится с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины (модуля), организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков, приобретенных обучающимися в ходе выполнения индивидуальных заданий на практических занятиях и лабораторных работах. При оценивании результатов освоения практических занятий и лабораторных работ применяется шкала оценки «зачтено – не зачтено». Количество лабораторных и практических работ и их тематика определена рабочей программой дисциплины, утвержденной заведующим кафедрой.

Результат выполнения каждого индивидуального задания должен соответствовать всем критериям оценки в соответствии с компетенциями, установленными для заданного раздела дисциплины.

Промежуточный контроль по дисциплине осуществляется проведением экзамена.

#### *Паспорт фонда оценочных средств по дисциплине*

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
1	Введение во встроенные системы	ПК-3	текущий контроль, зачет
2	Архитектуры процессоров для встроенных систем стохастической структуры.	ПК-3	текущий контроль, зачет
3	Операционная система Linux	ПК-3	текущий контроль, зачет
4	Инструментальные средства разработки ПО для встроенных систем	ПК-3	текущий контроль, зачет
5	Встроенные системы на основе Linux	ПК-3	текущий контроль, зачет
6	Курсовой проект	ПК-3	защита курсового проекта

### Общепрофессиональные компетенции выпускников и индикаторы их достижения

Категория (группа) общепрофессиональных компетенций	Код и наименование общепрофессиональной компетенции	Код и наименование индикатора достижения общепрофессиональной компетенции
	ПК-3 Способен производить разработку функциональной схемы изделий "система в корпусе".	<p>ИД – 1 ПК-3 Знать: основные принципы построения встроенных систем ЭВС и изделий "система в корпусе".</p> <p>ИД – 2 ПК-3 Уметь: разрабатывать функциональные схемы встроенные компьютерные системы и изделий "система в корпусе" на основе открытого программного обеспечения.</p> <p>ИД – 3 ПК-3 Владеть: навыками работы с распространенными средствами разработки встроенных компьютерных систем и изделий "система в корпусе".</p>

#### *Типовые контрольные задания или иные материалы*

#### **Вопросы к лабораторным занятиям по дисциплине**

- 1) Что собой представляет программа на языке Shell? В каких случаях лучше не использовать язык Shell?
- 2) В чем особенность переменных в языке Shell? Каким образом переменные определяются в программе? Как осуществляется присвоение значения переменной? Как выполняются операции ввода-вывода?
- 3) Что собой представляют позиционные и специальные параметры, переменные окружения?
- 4) В чем особенности использования различных скобок и кавычек в языке Shell? Как можно вычислить значение арифметического выражения в языке Shell?
- 5) Для чего предназначены команды if и test языка Shell? Перечислите основные параметры команды test.
- 6) Для чего предназначены команды for, while и until языка Shell? В чем особенности циклов языка Shell по сравнению с другими языками программирования?
- 7) Как реализованы функции в языке Shell? Каким способом функциям передаются параметры?
- 8) Опишите устройство аппаратной составляющей маршрутизатора D-Link Dir-300 и основные выполняемые им функции.
- 9) Что собой представляет файловая система Proc? Какие данные об устройстве маршрутизатора D-Link Dir-300 можно получить с помощью этой файловой системы?
- 10) Опишите структуру исходного кода программного обеспечения маршрутизатора D-Link Dir-300.
- 11) Опишите методику сборки программного обеспечения маршрутизатора D-Link Dir-300.
- 12) Каким образом можно добавить другие файлы в корневую файловую систему маршрутизатора?
- 13) Что собой представляет программа BusyBox? Какую роль она играет во встроенных системах? Каким образом происходит работа с программой?

- 14) Синтаксис каких команд и программ Linux может поддерживаться программой BusyBox? Перечислите основные апплеты.
- 15) Что собой представляет процесс конфигурации программы BusyBox? Перечислите основные команды для конфигурации программы BusyBox?
- 16) Какие параметры компиляции программы BusyBox могут быть заданы на этапе ее конфигурации?
- 17) Какой командой выполняется сборка программы BusyBox? Как можно удалить созданные на этапе компиляции временные файлы?
- 18) Что собой представляет процесс установки программы BusyBox?
- 19) Для чего предназначена система сборки корневой файловой системы Buildroot?
- 20) Перечислите основные команды конфигурации Buildroot.
- 21) Опишите назначение основных файлов и подкаталогов Buildroot.
- 22) Что понимают под термином «пакет» в рамках системы сборки корневой файловой системы Buildroot?
- 23) Какие файлы нужно создать для создания пакета в Buildroot?
- 24) Опишите назначение и синтаксис файла «Config.in».
- 25) Перечислите основные этапы обработки пакета в Buildroot. Какие из этих этапов выполняются системой сборки автоматически? Какие типы пакетов могут обрабатываться в Buildroot?
- 26) Как в Buildroot обрабатываются пакеты общего вида?
- 27) Как в Buildroot обрабатываются пакеты типа «autotools-package»?

### **Вопросы к курсовому проектированию**

- 1) Что такое «встроенная вычислительная система»? Приведите примеры встроенных систем.
- 2) Какие основные компоненты включает в себя программное обеспечение встроенных систем на основе Linux?
- 3) В чем состоит метод кросс-компиляции программ при разработке ПО для встроенных систем?
- 4) Какие языки программирования используются при разработке ПО для встроенных систем на основе Linux?
- 5) Что содержит типичная корневая файловая система для встроенных Linux-систем?
- 6) Приведите основные характеристики маршрутизатора D-Link DIR-300 (аппаратная ревизия В7).
- 7) Что содержится в основных каталогах исходного кода ПО маршрутизатора D-Link DIR-300?
- 8) Как производится сборка ПО маршрутизатора D-Link DIR-300? Какие основные этапы выполняются и какие файлы создаются при сборке?
- 9) Перечислите основные команды конфигурации ядра Linux. Чем отличаются запускаемые ими процедуры конфигурации ядра?
- 10) Перечислите основные пункты меню конфигурации ядра Linux.
- 11) Для чего предназначена программа BusyBox? Перечислите основные апплеты BusyBox.
- 12) Перечислите основные команды конфигурации программы BusyBox. Чем отличаются данные команды друг от друга?
- 13) В чем заключается разработка пользовательского апплета для программы BusyBox?
- 14) Для чего предназначена система сборки корневой файловой системы Buildroot?
- 15) Перечислите основные команды конфигурации Buildroot.
- 16) Какие основные стандартные пакеты доступны в Buildroot?
- 17) Что при разработке пользовательского пакета содержится в файле Config.in?
- 18) Какие основные типы пользовательских пакетов поддерживаются в Buildroot? Какие основные действия выполняются при обработке пакета в Buildroot?
- 19) Что содержит файл пакет.mk для пользовательского пакета общего вида?

- 20) Что содержит файл пакет.mk для пользовательского пакета, основанного на системе сборки GNU Autotools?

### Вопросы к практическим занятиям по дисциплине

- 1) Какие типы файлов существуют в Linux? Охарактеризуйте каждый тип файла.
- 2) Как устроено дерево каталогов Linux? Каково назначение основных каталогов?
- 3) Какие команды в Linux предназначены для просмотра содержимого текущего каталога и перехода в другой каталог? Перечислите основные опции данных команд.
- 4) Какие команды в Linux предназначены для создания и удаления каталогов, для удаления файлов?
- 5) Какие команды Linux предназначены для копирования и перемещения файлов и каталогов?
- 6) Как устроена система прав доступа к файлу в Linux?
- 7) Каким образом задаются права доступа к файлам в команде chmod в текстовом и цифровом виде?
- 8) Что такое процесс? Что собой представляют родительские, дочерние, сестринские процессы? Какое место в иерархии процессов занимает процесс init?
- 9) В каких состояниях может находиться процесс? Что такое PID и PPID?
- 10) Чем отличается запуск процесса в фоновом режиме от запуска процесса на переднем плане? Как запустить процесс в фоновом режиме? Для чего предназначены команды bg, fg, jobs?
- 11) Для чего предназначена команда ps? Перечислите основные опции и режимы данной команды.
- 12) Что такое сигнал? Для чего предназначена команда kill?
- 13) Что собой представляет компилятор GCC? Какие основные аппаратные платформы и языки программирования им поддерживаются?
- 14) Что собой представляет процесс компиляции программы на языке C компилятором GCC? Каковы основные этапы данного процесса? Какие получаются промежуточные и конечные результаты?
- 15) С помощью какой команды выполняется компиляция простейшей программы на языке C? Как выполняется компиляция программы, состоящей из нескольких файлов? Как выполняется компиляция программы, использующей статическую библиотеку?
- 16) Как выполняется компиляция статических и динамических библиотек?
- 17) Как при компиляции задать используемый стандарт языка C? Как включить в исполняемый файл отладочную информацию? Какие существуют режимы оптимизации кода в компиляторе GCC? Какие существуют параметры выдачи предупреждений компилятора?
- 18) Что собой представляет система сборки? Какие основные задачи она решает?
- 19) Откуда берет исходные данные система сборки Make? Какой используется формат данных? Какой командой запускается сборка?
- 20) Что означают понятия «цель» и «зависимости» в терминологии утилиты Make?
- 21) Что такое «абстрактная цель»?
- 22) В соответствии с каким алгоритмом работает утилита Make?
- 23) Перечислите основные стандартные имена целей в Make. Для чего предназначена каждая из указанных целей?
- 24) Что собой представляют переменные в Make? Какие существуют ограничения на их объявление и использование?

### Задания к курсовому проектированию

Разработайте программное обеспечение для встроенной системы (маршрутизатора заданной модели) в соответствии с исходными данными приведенными ниже. Проверьте работоспособность и полноту реализации всех заданных функций программного обеспечения встроенной системы.

**Исходные данные, общие для всех вариантов**

Параметр	Значение
Рекомендуемая версия Linux для рабочей машины	32-разрядная версия дистрибутива Linux Mint MATE
Встроенная система	Маршрутизатор D-Link DIR-300 Hardware revision B7
Путь к исходному коду программного обеспечения маршрутизатора по GPL	<a href="https://cloud.dlink.ru/owcl/index.php/s/DUJ9hd0OiONTiFU">https://cloud.dlink.ru/owcl/index.php/s/DUJ9hd0OiONTiFU</a>
Версия ядра Linux	2.6.32
Версия программы Busybox	1.19.2
Имя тестового скрипта	about.sh
Выполняемые тестовым скриптом действия	<p>Вывод информации о курсовом проекте:</p> <ol style="list-style-type: none"> <li>2. ФИО студента,</li> <li>3. номер учебной группы,</li> <li>4. дата сборки прошивки.</li> </ol> <p>Демонстрация выполненных модификаций в программном обеспечении маршрутизатора:</p> <ul style="list-style-type: none"> <li>• выполненных изменений в ядре Linux через вывод соответствующих значений из файловой системы Proofs,</li> <li>• запуск программы BusyBox для выполнения добавленных команд и результатов их работы,</li> <li>• запуск программы BusyBox для демонстрации работы разработанного апплета,</li> <li>• запуск программы для разработанного пакета Buildroot.</li> </ul>

**Исходные данные по вариантам**

№ вар-та	Параметр	Значение
1	Модификация ядра Linux	Включить поддержку сетевого протокола CAN. Включить поддержку файловой системы ext2.
	Добавляемые команды BusyBox	Команды cal и sort
	Разрабатываемый апплет BusyBox	Разработайте апплет «not», реализующий функцию логического отрицания. Апплет имеет один параметр, задаваемый как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
2	Модификация ядра Linux	Включить поддержку инфракрасной передачи данных (IrDA). Включить поддержку файловой системы ext3.
	Добавляемые команды BusyBox	Команды chmod и grep
	Разрабатываемый апплет BusyBox	Разработайте апплет «and2», реализующий функцию логического И. Апплет имеет два параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
3	Модификация ядра Linux	Включить поддержку сетевого протокола Bluetooth. Включить поддержку файловой системы RaiserFS.
	Добавляемые команды BusyBox	Команды adduser и deluser

	Разрабатываемый апплет BusyBox	Разработайте апплет «or2», реализующий функцию логического ИЛИ. Апплет имеет два параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
4	Модификация ядра Linux	Включить поддержку сетевого протокола I2C. Включить поддержку файловой системы для CD ISO 9660.
	Добавляемые команды BusyBox	Команды gzip и gunzip
	Разрабатываемый апплет BusyBox	Разработайте апплет «xor», реализующий функцию логического исключающего или. Апплет имеет два параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
5	Модификация ядра Linux	Включить поддержку сетевого протокола SPI. Включить поддержку файловой системы MS-DOS.
	Добавляемые команды BusyBox	Команды find и pwd
	Разрабатываемый апплет BusyBox	Разработайте апплет «impl», реализующий функцию логической импликации. Апплет имеет два параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
6	Модификация ядра Linux	Включить поддержку сетевого протокола CAN. Включить поддержку файловой системы NTFS.
	Добавляемые команды BusyBox	Команды patch и cut
	Разрабатываемый апплет BusyBox	Разработайте апплет «nand2», реализующий логическую функцию И-НЕ. Апплет имеет два параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
7	Модификация ядра Linux	Включить поддержку инфракрасной передачи данных (IrDA). Включить поддержку файловой системы ext2.
	Добавляемые команды BusyBox	Команды ps и traceroute
	Разрабатываемый апплет BusyBox	Разработайте апплет «nor2», реализующий логическую функцию ИЛИ-НЕ. Апплет имеет два параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
8	Модификация ядра Linux	Включить поддержку сетевого протокола Bluetooth. Включить поддержку файловой системы ext3.
	Добавляемые команды BusyBox	Команды cal и sort
	Разрабатываемый апплет BusyBox	Разработайте апплет «and3», реализующий функцию логического И. Апплет имеет три параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
9	Модификация ядра Linux	Включить поддержку сетевого протокола I2C. Включить поддержку файловой системы RaiserFS.
	Добавляемые команды BusyBox	Команды chmod и grep
	Разрабатываемый апплет BusyBox	Разработайте апплет «or3», реализующий функцию логического ИЛИ. Апплет имеет три параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».

10	Модификация ядра Linux	Включить поддержку сетевого протокола SPI. Включить поддержку файловой системы для CD ISO 9660.
	Добавляемые команды BusyBox	Команды adduser и deluser
	Разрабатываемый апплет BusyBox	Разработайте апплет «nand3», реализующий логическую функцию И-НЕ. Апплет имеет три параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
11	Модификация ядра Linux	Включить поддержку сетевого протокола CAN. Включить поддержку файловой системы MS-DOS.
	Добавляемые команды BusyBox	Команды gzip и gunzip
	Разрабатываемый апплет BusyBox	Разработайте апплет «nor3», реализующий логическую функцию ИЛИ-НЕ. Апплет имеет три параметра, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
12	Модификация ядра Linux	Включить поддержку инфракрасной передачи данных (IrDA). Включить поддержку файловой системы NTFS.
	Добавляемые команды BusyBox	Команды find и pwd
	Разрабатываемый апплет BusyBox	Разработайте апплет «and», реализующий функцию логического И. Апплет имеет до пяти параметров, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
13	Модификация ядра Linux	Включить поддержку сетевого протокола Bluetooth. Включить поддержку файловой системы ext2.
	Добавляемые команды BusyBox	Команды patch и cut
	Разрабатываемый апплет BusyBox	Разработайте апплет «or», реализующий функцию логического ИЛИ. Апплет имеет до пяти параметров, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
14	Модификация ядра Linux	Включить поддержку сетевого протокола I2C. Включить поддержку файловой системы ext3.
	Добавляемые команды BusyBox	Команды nc и traceroute
	Разрабатываемый апплет BusyBox	Разработайте апплет «nand», реализующий логическую функцию И-НЕ. Апплет имеет до пяти параметров, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».
15	Модификация ядра Linux	Включить поддержку сетевого протокола SPI. Включить поддержку файловой системы RaiserFS.
	Добавляемые команды BusyBox	Команды chmod и grep
	Разрабатываемый апплет BusyBox	Разработайте апплет «nor», реализующий логическую функцию ИЛИ-НЕ. Апплет имеет до пяти параметров, задаваемых как «0», «false», «FALSE», «1», «true», «TRUE». Результат: «0» или «1».

**Список вопросов к экзамену по дисциплине**

1. Понятие встроенной системы. Примеры встроенных систем. Понятие системы реального времени.
2. Особенности аппаратной реализации встроенных систем.
3. Особенности программной реализации встроенных систем.
4. Общие сведения о Linux. История создания ОС Linux и Unix.
5. Дистрибутивы Linux. Области применения Linux.
6. Графический интерфейс Unix-подобных систем. Графическая система X Window. Среды рабочего стола GNOME и KDE. Командные оболочки Linux.
7. Командная оболочка Bash. Структура команды Linux.
8. Компиляция простейшей программы в компиляторе GCC. Структура компилятора GCC: препроцессор, компилятор, ассемблер, компоновщик.
9. Компиляция простейшей программы в компиляторе GCC. Компиляция программ на языке C.
10. Структура компилятора GCC.
11. Понятие о системах сборки. Утилита Make. Введение в написание Make-файлов.
12. Алгоритм работы утилиты Make.
13. Понятие о системах сборки. Утилита Make. Стандартные имена целей.
14. Отладчики в Unix и Linux. Отладчик GDB. Общее описание и основные команды отладчика GDB. Пример отладки программы в отладчике GDB.
15. Основные программные компоненты встроенных систем на основе Linux.
16. Общие сведения о ядре Linux, версии ядра.
17. Обзор конфигурации ядра Linux. Команды конфигурации. Файл .config.
18. Компиляция ядра Linux. Установка ядра Linux и модулей ядра. Кросс-компиляция ядра Linux.
19. Программа BusyBox. Компиляция и конфигурация программы BusyBox.
20. Использование системы сборки корневой файловой системы Buildroot.
21. Разработка пакета в Buildroot.

Составил  
доцент кафедры САПР ВС  
к.т.н., доцент

В.А. Шибанов

Заведующий кафедрой САПР ВС,  
д.т.н., профессор

В.П. Корячко