

Направление подготовки 11.03.04 «Электроника и наноэлектроника»
Профиль «Промышленная электроника»
Методическое обеспечение дисциплины Б1.В.03 «Встраиваемые системы»



**Филиал федерального государственного бюджетного образовательного учреждения
высшего образования
«Национальный исследовательский университет «МЭИ»
в г. Смоленске**

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБЕСПЕЧЕНИЯ ОБРАЗОВАТЕЛЬНОГО
ПРОЦЕССА ПО ДИСЦИПЛИНЕ «ВСТРАИВАЕМЫЕ СИСТЕМЫ»**

Направление подготовки (специальность): **11.03.04 «Электроника и наноэлектроника»**

Профиль: **«Промышленная электроника»**

Уровень высшего образования: **бакалавриат**

Нормативный срок обучения: **4 года 11 месяцев**

Форма обучения: **заочная**

Год набора: **2020**

Смоленск 2020

Направление подготовки 11.03.04 «Электроника и нанoeлектроника»
Профиль «Промышленная электроника»
Методическое обеспечение дисциплины Б1.В.03 «Встраиваемые системы»



Методические материалы составил:

Доцент кафедры
электроники и микропроцессорной техники

Образцов Сергей Александрович
ФИО

«24» июня 2020 г.

Заведующий кафедрой электроники и микропроцессорной техники

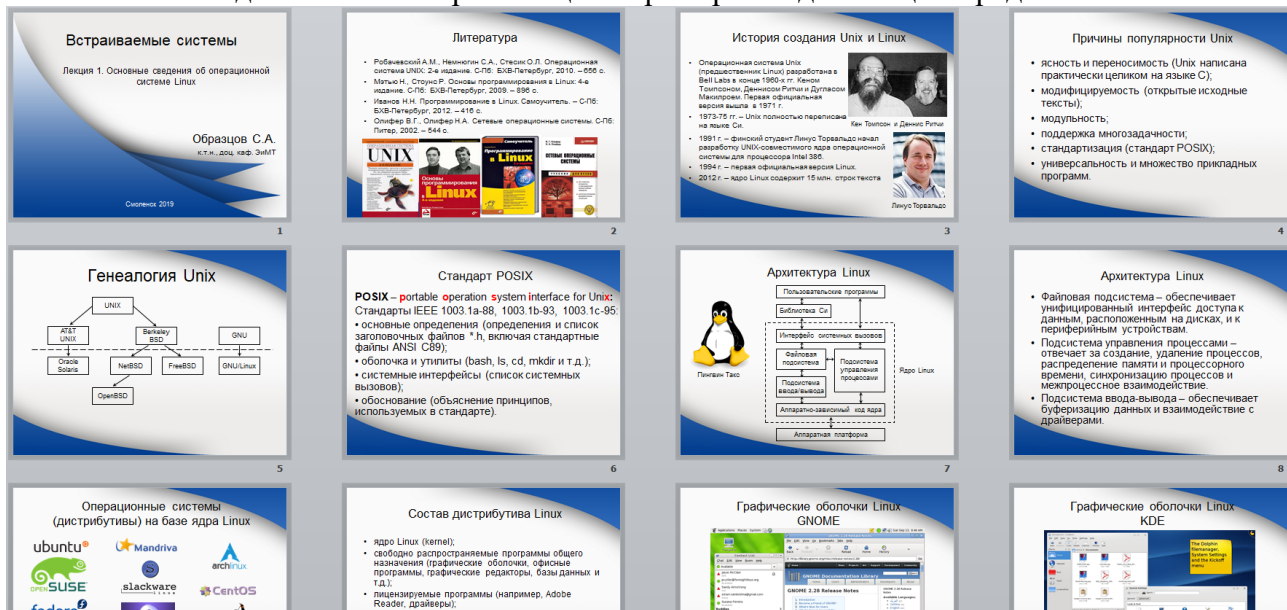
подпись

Якименко Игорь Владимирович
ФИО

«02» июля 2020 г.

1. Лекционный курс.

Лекции по дисциплине «Встраиваемые системы» проводятся в мультимедийном формате с использованием подготовленных презентаций. Пример слайдов лекций представлен ниже.



Ссылка для скачивания лекций размещена на странице дисциплины на сайте кафедры электроники и микропроцессорной техники: <https://www.eimt.ru/bakalavriat-2019/it>.

2. Практические занятия (упражнения).

Во время практических занятий предусмотрено решение практических задач по администрированию операционной системы GNU/Linux с участием и под руководством преподавателя. Практикум по решению задач размещен на странице дисциплины на сайте кафедры электроники и микропроцессорной техники: <https://www.eimt.ru/bakalavriat-2019/vs>.

Примеры заданий из практикума представлены ниже:

Практическое занятие №1

Основы работы в командной оболочке bash

Цель работы: изучение командной оболочки **bash** и приобретение навыков работы с ее основными командами.

1 Теоретические сведения

Командная оболочка bash

Bash — это наиболее распространенный командный интерпретатор в операционных системах семейства Linux, с его помощью пользователь может либо вводить команды

операционной системе по отдельности, либо запускать скрипты, состоящие из списка доступных команд.

Работа с командной оболочкой **bash** осуществляется в консоли, доступ к которой можно получить с помощью виртуального терминала или эмулятора терминала. Для перехода в виртуальный терминал следует нажать комбинацию клавиш **Ctrl+Alt+Fi** (вместо *i* могут быть числа от 1 до 6), запуск эмулятора терминала выполняется в графической оболочке через главное меню. В результате на экране будет отображаться однотонный текстовый экран (в случае эмулятора терминала — часть экрана), вверху которого будет выведено приглашение. По виду окончания приглашения можно определить, является ли текущий пользователь суперпользователем-**root** («#» — для **root** и «\$» — для остальных пользователей). Все манипуляции в консоли совершаются с помощью команд, набранных с клавиатуры.

Поиск нужной команды можно осуществить, набрав предполагаемое начало команды и нажав дважды клавишу **Tab**, в результате чего будет получен список команд, начинающийся с введенных символов.

Для каждой доступной команды можно получить описание (справку). Рекомендуется вызывать его всякий раз перед первым вызовом команды. Справка вызывается командой вида:

```
man <имя команды>
```

либо

```
<имя команды> --help
```

У команды **man** также имеются дополнительные опции, о которых можно узнать, набрав в командной строке «**man man**» или «**man --help**». Выход из режима чтения справки **man** происходит по нажатию клавиши **q**.

При работе в консоли на экране уместается ограниченное количество строк, для просмотра остальных строк необходимо воспользоваться прокруткой с помощью сочетания клавиш **Shift+PageUp** (прокрутка вверх) и **Shift+PageDown** (прокрутка вниз).

В течение каждого сеанса работы с **bash** в файле **.bash_history** сохраняется история команд (имя файла установлено в переменной **\$HISTFILE**). Файл находится в домашнем каталоге пользователя, их вызвавшего. Количество команд, записываемых в файл истории, ограничено и установлено в переменной **\$HISTSIZE**. Значение этих и других системных переменных можно узнать командой

```
echo $HISTFILE $HISTSIZE
```

Для просмотра истории команд используется команда **history**. Для просмотра определенного количества команд, необходимо указать их число в качестве аргумента

```
history 5
```

Перемещение между предыдущими командами осуществляется нажатием клавиш «вверх» и «вниз».

2 Подготовка к работе

1. Изучите команды работы с файлами: создание **touch**, перезапись файла **echo** "текст" > file.txt, запись в конец файла **echo** "текст" >> file.txt, просмотр **cat**, **page**, **more**, **less**, редактирование **nano**, копирование **cp**, перемещение/переименование **mv**, удаление **rm**.
2. Изучите команды работы с каталогами: создание **mkdir**, просмотр **ls**, переход между каталогами **cd**, удаление **rmdir**.
3. Изучите средства перенаправления ввода-вывода в **bash** и способы использования специальных файлов устройств терминалов /dev/tty.
4. Изучите способы получения справки по использованию команд.

3 Рабочее задание

1. Очистите экран (команда **clear**).
2. В домашнем каталоге пользователя создайте одной командой заданную структуру: каталоги main_n, An, Bn, Cn, Dn, где n — номер варианта по журналу (рисунок 1).

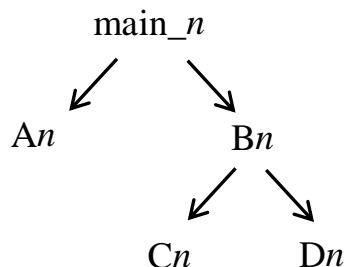


Рисунок 1 — Структура каталогов

3. В каталоге main_n создайте текстовый файл name.txt со своей фамилией с помощью команды **echo** и средств перенаправления ввода-вывода.
4. Допишите в файл name.txt из каталога main_n название группы с помощью команды **echo** и средств перенаправления ввода-вывода.
5. В каталоге An создайте текстовый файл date.txt с текущим временем в формате ДД.ММ.ГГГГ ЧЧ:ММ:СС (например, 12.10.2014 14:15:19), используя команду **date** и средства перенаправления ввода-вывода (обратите внимание на символ «+» перед описанием пользовательского формата даты).
6. Скопируйте файлы name.txt и date.txt в каталоги Bn и Cn соответственно.
7. Файл name.txt в каталоге main_n переименуйте в myname.txt.
8. Файл date.txt из каталога Cn переместите в каталог Dn.
9. Введите с клавиатуры файл list.txt со списком фамилий одноклассников (5–6 фамилий) путем копирования из специального файла устройства терминала /dev/tty.
10. Отредактировать текстовым редактором **nano** файл myname.txt, добавив в него свой номер варианта.
11. Выведите на экран содержимое файлов name.txt из каталога Bn и date.txt из каталога Cn.
12. На место файла name.txt из каталога Bn скопируйте файл date.txt из каталога Cn.

13. На место файла date.txt из каталога Cn скопируйте файл myname.txt из каталога main_n.
14. Выведите на экран содержимое файлов name.txt из каталога Вn и date.txt из каталога Cn.
15. Удалите файл name.txt из каталога Вn.
16. Выведите список всех созданных файлов и каталогов с помощью одной команды.
17. Сохраните историю команд в файл history.txt в каталоге Dn.
18. Очистите экран и покажите созданные структуры и содержимое файлов преподавателю.
19. Удалите все созданные структуры одной командой.

4 Контрольные вопросы

1. Как получить справку по функции printf языка программирования C?
2. Приведите практические примеры использования средств перенаправления ввода-вывода.
3. Чем отличаются команды вывода содержимого файлов на экран **page**, **less** и **more**?
4. Объедините два текстовых файла в один с помощью команды **cat**.
5. Как организовать чат между пользователями двух терминалов?
6. Что такое переменная окружения?
7. Как определить полное имя домашнего каталога текущего пользователя?
8. Приведите все способы записи пути к файлу myname.txt (текущий каталог — Dn).
9. Как скопировать файл без использования команды **cp**?
10. Как создать скрытый каталог?

3. Лабораторные работы.

Лабораторные работы проводятся в компьютерных классах. Задания на лабораторные работы представлены на странице дисциплины на сайте кафедры электроники и микропроцессорной техники: <https://www.eimt.ru/bakalavriat-2019/vs>.

Примеры лабораторных работ:

Лабораторная работа №1

Работа с файловой системой в Linux

Цели работы:

1. Получить практику работы с компилятором gcc и средой разработки Eclipse в операционной системе Linux.
2. Научиться использовать функции стандартной библиотеки для работы с файловой системой.

Задание:

1. Разработать алгоритм и программу согласно варианту задания с использованием языка Си. Программа должна принимать аргументы командной строки. Имена файлов должны храниться в динамическом списке.
2. Проверить функционирование разработанной программы на наборе тестов.

3. Провести анализ этапов выполнения задачи.
4. Подготовить и оформить отчет по выполнению лабораторной работы. Отчет должен содержать титульный лист, техническое задание, блок-схемы алгоритмов, исходный текст программы с комментариями, тесты, результаты их выполнения и выводы по работе.

Порядок выполнения работы

1. Создайте файл исходного текста программы с помощью редактора **nano**.
2. Разработайте алгоритм программы в соответствии с вариантом задания.
3. Разбейте алгоритм на несколько логических блоков.
4. Создайте исходный текст программы, реализующей разработанный алгоритм.
5. Скомпилируйте проект командой `gcc program.c -o program`, где *program* — название программы.
6. Устраните синтаксические ошибки, если такие имеются.
7. Разработайте набор тестов с учетом особенностей реализации алгоритма.
8. Проверьте правильность функционирования программы с использованием набора тестов.
9. Устраните семантические ошибки в программе, если такие имеются.

Методические указания

Для считывания аргументов командной строки используйте следующее объявление функции `main`:

```
int main(int argc, char* argv[])
```

В переменной `argc` содержится количество аргументов командной строки (один аргумент существует всегда — имя исполняемого файла программы). В массиве строк `argv[]` содержатся аргументы командной строки, например, `argv[0]` — имя исполняемого файла программы, `argv[1]` — первый аргумент и т.д.

Пример. Написать программу вывода списка файлов из текущего каталога с указанием их размера. Формат вызова программы: `list <путь>`.

```
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <sys/stat.h>

void main(int argc, char* argv[])
{
    DIR* d;
    struct dirent * entry;
    struct stat st;
```

```
// проверка количества аргументов if
(argc != 2)
{
    printf("Формат вызова: list путь\n");
    return;
}

// открытие каталога d =
opendir(argv[1])
if (d == NULL) // если каталог не найден
{
    printf("Каталог не найден\n");
    return;
}

// чтение каталога
while((entry = readdir(d)) != NULL)
{
    // получение информации о файле
    stat(entry->d_name, &st);
    // проверка, обычный ли это файл if
    (S_ISREG(st.st_mode))
    {
        // вывод на экран имени и размера файла
        printf("%20s\t%d", entry->d_name, st.st_size);
    }
}

closedir(d); // закрытие каталога
}
```

Индивидуальные задания

1. Написать программу поиска всех файлов в указанном каталоге, содержащих заданное слово. Формат вызова программы: wordsearch <путь> <искомое слово>. Программа должна выводить имена всех файлов в один столбец.
2. Написать программу поиска файла по имени в указанном каталоге и во всех вложенных. Формат вызова программы: filesearch <путь> <имя файла>. Программа должна вывести путь к искомому файлу.
3. Написать программу вывода списка имен всех файлов из указанного каталога и всех вложенных в алфавитном порядке с указанием размера файла. Формат вызова программы: filelist <путь>. Программа должна выводить список файлов в два столбца: имя файла, размер файла.

4. Написать программу вывода списка имен всех файлов из указанного каталога и всех вложенных в порядке возрастания или убывания размера файла. Формат вызова программы: `sizelist <путь> <-a по возрастанию или -d по убыванию>`. Программа должна выводить список файлов в два столбца: имя файла, размер файла.
5. Написать программу вывода иерархического списка всех вложенных каталогов, начиная с указанного. Формат вызова программы: `catlist <путь>`. Программа должна выводить иерархический список каталогов по строкам: в первой строке — корневой каталог, во второй — его подкаталоги, далее — их подкаталоги и т.д.
6. Написать программу вывода списка имен всех файлов из указанного каталога в алфавитном порядке с группировкой по расширению (список сортируется по возрастанию расширения, внутри каждой группы с одинаковым расширением — в порядке возрастания имени файла) с указанием размера файла. Формат вызова программы: `alphalist <путь>`. Программа должна выводить список файлов в два столбца: имя файла, размер файла.
7. Написать программу вывода списка имен всех файлов из указанного каталога в алфавитном порядке (по расширению), внутри каждой группы файлов с одинаковым расширением — в порядке возрастания размера файла с указанием размера файла. Формат вызова программы: `Xlist <путь>`. Программа должна выводить список файлов в два столбца: имя файла, размер файла.
8. Написать программу форматирования файла, содержащего дробные числа. Входные данные — файл с дробными числами в произвольном формате, выходные данные — файл с дробными числами в формате `XXX.XX` (например, `001.00`) в один столбец. Формат вызова программы: `floatformat -s <исходный файл> -d <выходной файл>`. Если входной или выходной файл не указаны, то данные вводятся с клавиатуры или выводятся на экран.
9. Написать программу автоматического создания и удаления каталогов по заданному шаблону. Шаблон имеет следующий вид: «каталог %d-й», где символ %d должен быть заменен на число. Формат вызова программы: `autodir <-c или -d> <шаблон> <начальный номер> <конечный номер>`, для создания каталогов с именами `cat1, cat2, cat3`: `autodir -c cat%d 1 3`.
10. Написать программу автоматического создания и удаления файлов по заданному шаблону. В каждый файл должно быть записано его имя. Шаблон имеет следующий вид: «file%d.txt», где символ %d должен быть заменен на число. Формат вызова программы: `autofile <-c (для создания) или -d (для удаления)> <шаблон> <начальный номер> <конечный номер>`, для создания каталогов с именами `file1.txt, file2.txt, file3.txt`: `autodir -c file%d.txt 1 3`.
11. Написать программу поиска всех файлов в указанном каталоге, содержащих заданное число. Формат вызова программы: `numbersearch <путь> <искомое число>`. Программа должна выводить имена всех файлов в один столбец.
12. Написать программу вывода списка имен всех файлов из указанного каталога и всех вложенных в алфавитном порядке (сначала по расширению, затем — по имени) с указанием размера файла. Формат вызова программы: `fileextlist <путь>`. Программа должна выводить список файлов в три столбца: имя файла, расширение файла, размер файла.
13. Написать программу измерения размера всех подкаталогов в заданном каталоге. Размер подкаталога определяется рекурсивно как сумма размеров всех его файлов, включая файлы

- во вложенных подкаталогах. Формат вызова программы: `dirsize <путь>`. Программа должна выводить список каталогов в два столбца: имя каталога, размер каталога.
14. Написать программу переноса всех текстовых файлов из одного каталога в другой с заменой всех букв внутри файла на прописные. Формат вызова программы: `mvlowcase <откуда> <куда>`. Пример: `mvlowcase ~/A1 ~/A2`.
 15. Написать программу подсчета количества слов для всех текстовых файлов *.txt заданного каталога. Формат вызова программы: `wordcount <путь>`. Программа должна выводить список файлов в два столбца: имя файла, количество слов в файле.
 16. Написать программу вывода иерархического списка всех вложенных каталогов и файлов, начиная с указанного. Формат вызова программы: `catfilelist <путь>`. Программа должна выводить иерархический список каталогов по строкам: в первой строке — корневой каталог, во второй — его подкаталоги и файлы, далее — их подкаталоги и файлы и т.д.
 17. Написать программу автоматического создания символических ссылок на все файлы из указанного каталога и всех его вложенных подкаталогов. Все ссылки должны быть сохранены в одном каталоге. Формат вызова программы: `autosymlink <каталог с файлами> <каталог с ссылками>`.
 18. Написать программу поиска всех файлов в указанном каталоге и во всех вложенных подкаталогах, содержащих в своем имени указанную строку. Формат вызова программы: `filenamesearch <путь> <искомая строка>`. Программа должна выводить имена всех файлов в один столбец.
 19. Написать программу поиска слова наибольшей длины для всех текстовых файлов *.txt заданного каталога. Формат вызова программы: `wordlen <путь>`. Программа должна выводить список файлов в три столбца: имя файла, найденное слово и его длина.
 20. Написать программу автоматического создания символических ссылок на все файлы из указанного каталога, содержащие в названии заданную строку. Все ссылки должны быть сохранены в одном каталоге. Формат вызова программы: `filtersymlink <каталог с файлами> <строка> <каталог с ссылками>`.
 21. Написать программу копирования всех текстовых файлов из одного каталога в другой с заменой всех букв внутри файла на строчные. Формат вызова программы: `srlowcase <откуда> <куда>`. Пример: `srlowcase ~/A1 ~/A2`.
 22. Написать программу поиска файла по имени в указанном каталоге и во всех вложенных. Формат вызова программы: `filesearch <путь> <имя файла>`. Программа должна вывести путь к искомому файлу.
 23. Написать программу вывода списка имен всех файлов из указанного каталога и всех вложенных в алфавитном порядке с указанием размера файла. Формат вызова программы: `filelist <путь>`. Программа должна выводить список файлов в два столбца: имя файла, размер файла.
 24. Написать программу вывода списка имен всех файлов из указанного каталога и всех вложенных в порядке возрастания или убывания размера файла. Формат вызова программы: `sizelist <путь> <-а по возрастанию или -d по убыванию>`. Программа должна выводить список файлов в два столбца: имя файла, размер файла.

25. Написать программу вывода иерархического списка всех вложенных каталогов, начиная с указанного. Формат вызова программы: `catlist <путь>`. Программа должна выводить иерархический список каталогов по строкам: в первой строке — корневой каталог, во второй — его подкаталоги, далее — их подкаталоги и т.д.
26. Написать программу вывода списка имен всех файлов из указанного каталога в алфавитном порядке с группировкой по расширению (список сортируется по возрастанию расширения, внутри каждой группы с одинаковым расширением — в порядке возрастания имени файла) с указанием размера файла. Формат вызова программы: `alphalist <путь>`. Программа должна выводить список файлов в два столбца: имя файла, размер файла.
27. Написать программу вывода списка имен всех файлов из указанного каталога в алфавитном порядке (по расширению), внутри каждой группы файлов с одинаковым расширением — в порядке возрастания размера файла с указанием размера файла. Формат вызова программы: `Xlist <путь>`. Программа должна выводить список файлов в два столбца: имя файла, размер файла.
28. Написать программу форматирования файла, содержащего дробные числа. Входные данные — файл с дробными числами в произвольном формате, выходные данные — файл с дробными числами в формате XXX.XX (например, 001.00) в один столбец. Формат вызова программы: `floatformat -s <исходный файл> -d <выходной файл>`. Если входной или выходной файл не указаны, то данные вводятся с клавиатуры или выводятся на экран.
29. Написать программу автоматического создания и удаления каталогов по заданному шаблону. Шаблон имеет следующий вид: «каталог %d-й», где символ %d должен быть заменен на число. Формат вызова программы: `autodir <-с или -d> <шаблон> <начальный номер> <конечный номер>`, для создания каталогов с именами cat1, cat2, cat3: `autodir -с cat%d 1 3`.
30. Написать программу автоматического создания символических ссылок на все файлы из указанного каталога, содержащие в названии заданную строку. Все ссылки должны быть сохранены в одном каталоге. Формат вызова программы: `filtersymlink <каталог с файлами> <строка> <каталог с ссылками>`.
31. Написать программу автоматического создания символических ссылок на все файлы из указанного каталога, содержащие в названии заданную строку. Все ссылки должны быть сохранены в одном каталоге. Формат вызова программы: `filtersymlink <каталог с файлами> <строка> <каталог с ссылками>`.
32. Написать программу автоматического создания и удаления файлов по заданному шаблону. В каждый файл должно быть записано его имя. Шаблон имеет следующий вид: «file%d.txt», где символ %d должен быть заменен на число. Формат вызова программы: `autofile <-с (для создания) или -d (для удаления)> <шаблон> <начальный номер> <конечный номер>`, для создания каталогов с именами file1.txt, file2.txt, file3.txt: `autodir -с file%d.txt 1 3`.
33. Написать программу поиска всех файлов в указанном каталоге, содержащих заданное число. Формат вызова программы: `numbersearch <путь> <искомое число>`. Программа должна выводить имена всех файлов в один столбец.
34. Написать программу вывода списка имен всех файлов из указанного каталога и всех вложенных в порядке возрастания или убывания размера файла. Формат вызова программы: `sizelist <путь> <-а по возрастанию или`

–d по убыванию>. Программа должна выводить список файлов в два столбца: имя файла, размер файла.

35. Написать программу поиска всех файлов в указанном каталоге, содержащих заданное слово.

Формат вызова программы: wordsearch

<путь> <искомое слово>. Программа должна выводить имена всех файлов в один столбец.