

Лабораторная работа №10

Операционные системы

Александрова Ульяна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	12

Список иллюстраций

4.1	Программа 1	8
4.2	Запуск программы 1	8
4.3	Архив	8
4.4	Создание программы 2	9
4.5	Проверка программы 2	9
4.6	Написание программы 3	9
4.7	Запуск программы 3	10
4.8	Создание программы 4	10
4.9	Проверка программы 4	11

Список таблиц

1 Цель работы

Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

2 Задание

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку.
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.
3. Написать командный файл — аналог команды ls (без использования самой этой команды и команды dir). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.
4. Написать командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки

3 Теоретическое введение

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек:

- оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций; - C-оболочка (или csh) — надстройка на оболочкой Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд; - оболочка Корна (или ksh) — напоминает оболочку C, но операторы управления программой совместимы с операторами оболочки Борна; - BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек C и Корна (разработка компании Free Software Foundation).

POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ.

Стандарты POSIX разработаны комитетом IEEE (Institute of Electrical and Electronics Engineers) для обеспечения совместимости различных UNIX/Linux-подобных операционных систем и переносимости прикладных программ на уровне исходного кода.

POSIX-совместимые оболочки разработаны на базе оболочки Корна.

4 Выполнение лабораторной работы

Я создаю файл программы и прописываю в ней необходимый код (рис. fig. 4.1).

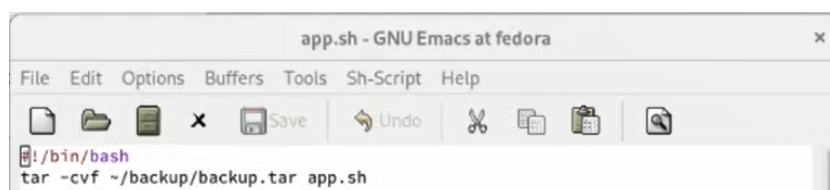


Рис. 4.1: Программа 1

Я запускаю программу и проверяю выполнение. Программа работает (рис. fig. 4.2), (рис. fig. 4.3).

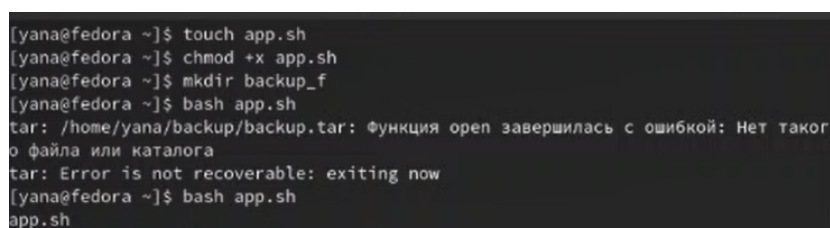


Рис. 4.2: Запуск программы 1

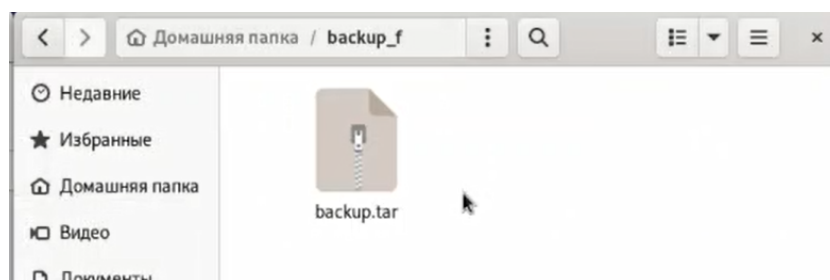
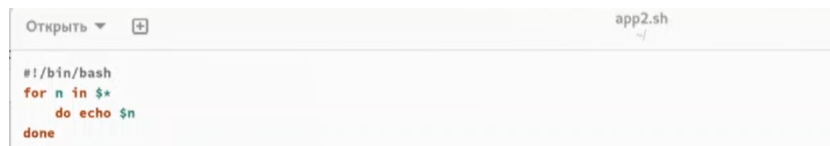


Рис. 4.3: Архив

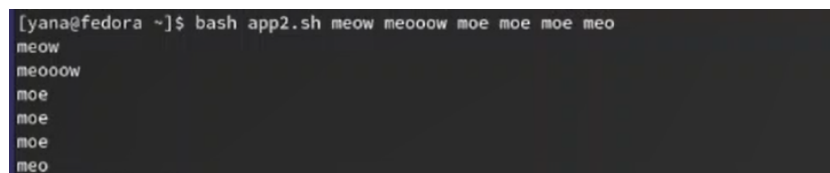
Создаю файл для второй программы и также ввожу код (рис. fig. 4.4).

A screenshot of a code editor window titled 'app2.sh'. The editor shows a shell script with the following content:

```
#!/bin/bash
for n in $*
do echo $n
done
```

Рис. 4.4: Создание программы 2

Проверяю работу командного файла. Все работает (рис. fig. 4.5).

A screenshot of a terminal window showing the execution of the script 'app2.sh'. The prompt is '[yana@fedora ~]\$'. The command entered is 'bash app2.sh meow meoooow мое мое мое meo'. The output of the script is:

```
meow
meoooow
мое
мое
мое
мое
meo
```

Рис. 4.5: Проверка программы 2

Создаю третий командный файл, редактирую (рис. fig. 4.6).

A screenshot of a code editor window titled 'app3.sh'. The editor shows a shell script with the following content:

```
#!/bin/bash
for A in *
do
    if test -d "$A"
    then
        echo "$A" - a directory"
    else
        echo -n "$A: - a file and "
        if test -w $A
        then
            echo "writeable"
        if test -r $A
        then
            echo "readable"
        else
            echo "not readable nor writeable"
        fi
    fi
done
```

Рис. 4.6: Написание программы 3

Проверяю работу. Все корректно (рис. fig. 4.7).

```
[yana@fedora ~]$ bash app3.sh ~/backup_f/
AleksandrovaUV.github.io^ - a directory
app2.sh: - a file and writeable
readable
app3.sh: - a file and writeable
readable
app.sh: - a file and writeable
readable
app.sh~: - a file and writeable
readable
backup_f^ - a directory
blog_UV^ - a directory
lab07.sh: - a file and writeable
readable
lab07.sh~: - a file and writeable
readable
new1.lab07: - a file and writeable
readable
reports^ - a directory
work^ - a directory
Видео^ - a directory
Документы^ - a directory
Загрузки^ - a directory
Изображения^ - a directory
Музыка^ - a directory
Общедоступные^ - a directory
Рабочий стол^ - a directory
Шаблоны^ - a directory
```

Рис. 4.7: Запуск программы 3

Создаю четвертый файл, заполняю (рис. fig. 4.8).

```
Открыть ▾ [icon] • app4.sh
~/
#!/bin/bash
format = ""
direc = ""
echo "Введите формат: "
read format
echo "Введите директорию: "
read direc
find "${direc}" -name ".*${format}" -type f | wc -l
ls
```

Рис. 4.8: Создание программы 4

Файл работает корректно (рис. fig. 4.9).

```
[yana@fedora ~]$ bash app4.sh
Введите формат:
sh
Введите директорию:
/home/yana
6
AleksandrovaUV.github.io  app.sh~  new1.lab07  Загрузки  Шаблоны
app2.sh                   backup_f  reports     Изображения
app3.sh                   blog_UV   work        Музыка
app4.sh                   lab07.sh  Видео       Общедоступные
app.sh                    lab07.sh~  Документы   'Рабочий стол'
```

Рис. 4.9: Проверка программы 4

5 Выводы

Я изучила основы программирования в оболочке ОС UNIX/Linux, научилась писать небольшие командные файлы.

...