

Отчет по лабораторной работе №12

**Программирование в командном процессоре ОС UNIX. Командные
файлы**

Разанацуа Сара Естэлл

Содержание

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

Список иллюстраций

3.1	Выполнение программы 1	7
3.2	Написанная программа 1	7
3.3	Выполнение программы 2	8
3.4	Написанная программа 2	8
3.5	Выполнение программы 3	9
3.6	Написанная программа программа 3	10
3.7	Создание файла с программой 4	10
3.8	Написанная программа 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 Выполнение лабораторной работы

Создаю файл prog1.sh в котором буду пистать программу с расширением sh (shell) с помощью утилиты touch, далее делаю его исполняемым с помощью chmod +x, открываю файл в любом текстовом редакторе, пишу в нем код, и после того как я написала программу в файле, я могу его запустить bash (рис. fig. 3.1).

```
serazanacua@serazanacua:~$ mkdir backup
serazanacua@serazanacua:~$ bash prog1.sh
prog1.sh
serazanacua@serazanacua:~$
```

Рис. 3.1: Выполнение программы 1

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 (рис. fig. 3.2).



```
Открыть ▾ + prog1.sh
~/
#!/bin/bash
tar -cvf ~/backup/backup.tar prog1.sh
```

Рис. 3.2: Написанная программа 1

#!/bin/bash tar -cvf ~/backup/backup.tar prog1.sh Создаю файл prog2.sh в котором буду пистать программу с расширением sh (shell) с помощью утилиты touch, далее делаю его исполняемым с помощью chmod +x, открываю файл в любом текстовом

редакторе, пишу в нем код, и после того как я написала программу в файле, я могу его запустить bash (рис. fig. 3.3).

```
serazanacua@serazanacua:~$ chmod +x prog2.sh
serazanacua@serazanacua:~$ bash prog2.sh 1 2 3 44 56 77
1
2
3
44
56
77
serazanacua@serazanacua:~$ bash prog2.sh 30 io 229
30
io
229
serazanacua@serazanacua:~$
```

Рис. 3.3: Выполнение программы 2

Пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов (рис. fig. 3.4).



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

Рис. 3.4: Написання программа 2

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

Создаю файл prog3.sh в котором буду пистаь программу с расширением sh (shell) с помощью утилиты touch, далее делаю его исполняемым с помощью chmod +x, открываю файл в любом текстовом редакторе, пишу в нем код, и после того как я написала программу в файле, я могу его запустить bash (рис. fig. 3.5).


```

serazanacua@serazanacua:~$ touch prog3.sh
serazanacua@serazanacua:~$ chmod +x prog3.sh
serazanacua@serazanacua:~$ bash prog3.sh iloveos/
abc1: is a file and writeable
readable
backup^ is a directory
bin^ is a directory
conf.txt: is a file and writeable
readable
Desktop^ is a directory
dotfiles^ is a directory
Downloads^ is a directory
file.txt: is a file and writeable
readable
git-extended^ is a directory
git@github.com:Sarahestelle^ is a directory
https:^ is a directory
#lab07.sh#: is a file and writeable
readable
lab07.sh: is a file and writeable
readable
lab07.sh~: is a file and writeable
readable
LICENSE: is a file and writeable
readable
may: is a file and writeable
readable
monthly^ is a directory
newdir^ is a directory
pandoc-3.1.12.2^ is a directory
pandoc-crossref: is a file and writeable
readable
pandoc-crossref.1: is a file and writeable
readable
prog1.sh: is a file and writeable
readable
prog2.sh: is a file and writeable
readable
prog3.sh: is a file and writeable
readable
reports^ is a directory
ski.places: is a file and writeable
readable

```

Рис. 3.5: Выполнение программы 3

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. (рис. fig. 3.6).

```

#!/bin/bash
for A in *
do
    if test -d "$A"
    then
        echo "$A^ is a directory"
    else
        echo -n "$A: is a file and "
        if test -w $A
        then
            echo writeable
            if test -r $A
            then
                echo "readable"
            else
                echo "neither readable or writeable"
            fi
        fi
    fi
done

```

Рис. 3.6: Написанная программа программа 3

`#!/bin/bash for A in * do if test -d "$A" then echo "$A^ is a directory" else echo -n "$A: is a file and " if test -w $A then echo writeable if test -r $A then echo "readable" else echo "neither readable or writeable" fi fi fi done`

Создаю файл `prog4.sh` в котором буду писать программу с расширением `sh` (shell) с помощью утилиты `touch`, далее делаю его исполняемым с помощью `chmod +x`, открываю файл в любом текстовом редакторе, пишу в нем код (рис. fig. 3.7).

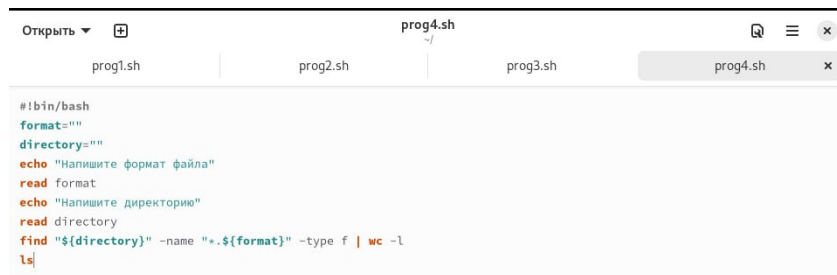
```

serazanacua@serazanacua:~$ bash prog4.sh
Напишите формат файла
text.txt
Напишите директорию
/home/serazanacua
find: '/home/serazanacua/ski.places/test/play/games': Отказано в доступе
0

```

Рис. 3.7: Создание файла с программой 4

- 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 (рис. fig. 3.8).



```
#!/bin/bash
format=""
directory=""
echo "Напишите формат файла"
read format
echo "Напишите директорию"
read directory
find "${directory}" -name ".*${format}" -type f | wc -l
ls
```

Рис. 3.8: Написанная программа 4

4 Выводы

- При выполнении данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX/Linux, научилась писать небольшие командные файлы.