## Отчёт по лабораторной работе №12

Простейший вариант

Ахмад Мд Шешир

### Содержание

1	ль работы	
2	Задания	6
3	Выполнение лабораторной работы	7
	3.1 Задача 1	7
	3.2 Задача 2	9
	3.3 Задача 3	10
	3.4 Задача 4	11
4	Выводы	13

# Список иллюстраций

3.1	Набераю текст	8
3.2	файл	11
3.3	программа	12
3.4	исполняемый файл	12
3.5	проверка	12

## Список таблиц

## 1 Цель работы

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

#### 2 Задания

- 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 ta
- 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

#### 3.1 Задача 1

1. Создаю файл backup\_script.sh где будет сам скрипт (рис. **??**).



2. Набираю программу где создаю переменные с путями для резервной копии . (рис. 3.1).

```
Открыть ▼ 

*backup_script.sh

Coxpanutь 

1 | 2 васкир_DIR="$HOME/backup" 3 4 mkdir -p "$Backup_DIR" 5 6 SCRIPT_NAME=$(basename "$0") 7 8 tar -czf "$Backup_DIR/$(SCRIPT_NAME).tar.gz" "$0" 9 10 echo "Резервная копия создана: $BACKUP_DIR/$(SCRIPT_NAME).tar.gz"  

*sh▼ Ширина табуляции: 8 ▼ Ln1, Col1
```

Рис. 3.1: Набераю текст

3. Делаю файл исполняемым, выполнив команду в терминале chmod (рис. ??).

```
(base) [sheshir.m@fedora ~]$ chmod +x backup_script.sh
(base) [sheshir.m@fedora ~]$
[
```

4. Запускаю исполняемый файл и проверяю существует ли архив (рис. ??).

```
(base) [sheshir.m@fedora ~]$ bash backup_script.sh
Резервная копия создана: /home/sheshir.m/backup/backup_script.sh.tar.gz
(base) [sheshir.m@fedora ~]$ 

(base) [sheshir.m@fedora ~]$ ls /home/sheshir.m/backup/backup_script.sh.tar.gz
/home/sheshir.m/backup/backup_script.sh.tar.gz
(base) [sheshir.m@fedora ~]$
```

#### 3.2 Задача 2

1. Открываю текстовый редактор и создайте новый файл print\_args.sh (рис. **??**).

```
[sheshir.m@fedora ~]$ touch print_args.sh
[sheshir.m@fedora ~]$ gedit print_args.sh

I
```

2. Набираю программу которая будет сначала проверять переданы ли аргументы, а если переданы будет их перебирать с помощью цикла for и выводить (рис. ??).

```
Тургіть  

*print_args.sh

1
2 if [ "$#" -eq 0 ]; then
3 echo "Нет переданных аргументов."
4 exit 1
5 fi
6
7 echo "Переданные аргументы:"
8 for arg in "$@"; do
9 echo "$arg"

10 done

*print_args.sh
*print_args.sh
```

3. Делаю файл исполняемым, выполнив команду в терминале chmod .(рис. **??**).

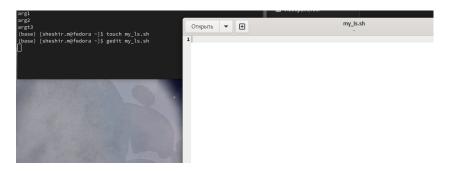
```
[sheshir.m@fedora ~]$ chmod +x print_args.sh
[sheshir.m@fedora ~]$
```

4. Запускаю исполняемы файл и проверяю работу (рис. ??).

```
(base) [sheshir.m@fedora ~]$ ./print_args.sh arg1 arg2 argt3
Переданные аргументы:
arg1
arg2
argt3
(base) [sheshir.m@fedora ~]$
```

#### 3.3 Задача 3

1. Открываю редактор gedit и создаю новый файл my\_ls.sh (рис. ??).



2. Пишу программу которая будет проверять существование переданного каталога, и вывода всего его содержимого. (рис. ??).

```
Тироверка на передачу каталога
2 if [ "$#" -ne l ]; then
3 echo "Использование: $0 <каталог>"
4 exit 1
5 fi
6 DIRECTORY=$1
7
8 # перебираем все файлы в каталоге|
9 echo "Содержимое каталога '$DIRECTORY':"
10 for FILE in "$DIRECTORY"+; do
11 if [ -e "$FILE" ]; then
12 # Получаем имя файла
13 FILENAME=$(basename "$FILE")
14 # Получаем права доступа
15 PERMISSIONS=$(ls -ld "$FILE" | awk '{print $1}')
16 # Выводим имя файла и права доступа
17 echo "$PERMISSIONS $FILENAME"
18 fi
19 done
```

3. Делаю файл исполняемым, выполнив команду в терминале chmod (рис. ??).

```
(base) [sheshir.m@fedora ~]$ chmod +x my_ls.sh
(base) [sheshir.m@fedora ~]$ █
```

4. Запускаю исполняемы файл, передаю ей в качестве аргумента путь до нужной директории. Проверяю работу (рис. ??).

```
(base) [sheshir.m@fedora ~]$ ./my_ls.sh /home/sheshir.m/lab7
Содержим каталога '/home/sheshir.m/lab7':
-rw-r--r-. abc1
-rw-r--r-. dis.cpp
-rw-r--r-. equipment.h
-rw-rw-r---. lab07.txt
-rwx----. my_os
-rw-r--r-. piu.py
-rwx----. text.txt
(base) [sheshir.m@fedora ~]$
```

#### 3.4 Задача 4

1. Открываю редактор gedit и создаю новый файл count\_files.sh (рис. 3.2).

```
(base) [sheshir.m@fedora ~]$ touch count_files.sh
(base) [sheshir.m@fedora ~]$ S
```

Рис. 3.2: файл

2. Пишу программу которая будет проверять существование переданного каталога, и вывода всего его содержимого. (рис. 3.3).

```
Открыть ▼ 

1 # Проверяем переданы ли два аргумента
2 if [ "$#" -ne 2 ]; then
3 echo "Использование: $0 < nуть_к_каталогу> <формат_файла>"
4 exit 1
5 fi
6 DIRECTORY=$1
7 EXTENSION=$2
8 # Проверяем существует каталог
9 if [: -d "$DIRECTORY"]; then
10 echo "Ошибка: '$DIRECTORY' не является каталогом или не существует."
11 exit 1
12 fi
13
14 # Считаем количество файлов с указанным форматом
15
16 FILE_COUNT=$(find "$DIRECTORY" -maxdepth 1 -type f -name "*.$EXTENSION" | wc -l)
17 # Выводим результат
18 echo "Количество файлов с форматом '$EXTENSION' в каталоге '$DIRECTORY': $FILE_COUNT"
```

Рис. 3.3: программа

3. Делаю файл исполняемым, выполнив команду в терминале chmod (рис. 3.4).

```
(base) [sheshir.m@fedora ~]$ chmod +x count_files.sh
(base) [sheshir.m@fedora ~]$ █
```

Рис. 3.4: исполняемый файл

4. Запускаю исполняемы файл, передаю ей в качестве аргумента путь до нужной директории и формал файла (png, txt doc и тд). Проверяю работу (рис. 3.5).



Рис. 3.5: проверка

#### 4 Выводы

В ходе работы я познакомился с основами программирования в оболочке ОС UNIX/Linux. Научился писать небольшие командные файлы.