

Лабораторная работа №10

Операционные системы

Савурская Полина

Содержание

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

Список иллюстраций

3.1	создание файла script.sh	6
3.2	пишем код	6
3.3	папка backup	7
3.4	заархивированный файл	7
3.5	создание файла script2.sh	8
3.6	пишем код	8
3.7	запуск файла script2.sh	8
3.8	создание файла script3.sh	9
3.9	пишем код	9
3.10	запуск файла script3.sh	10
3.11	создание файла script4.sh	10
3.12	пишем код	11
3.13	запуск файла script4.sh	11

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 передается в виде аргумента командной строки.

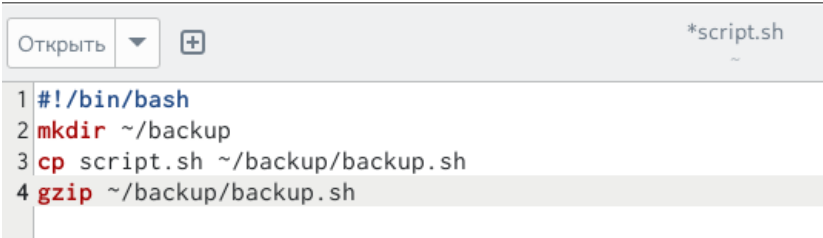
3 Выполнение лабораторной работы

1. Изучаем справку о команде tar. Создаем файл script.sh. Задаем ему необходимые разрешения. Проверяем, появился ли у нас этот файл.

```
pasavurskaya@dk8n60 ~ $ man tar
pasavurskaya@dk8n60 ~ $ touch script.sh
pasavurskaya@dk8n60 ~ $ chmod +x script.sh
pasavurskaya@dk8n60 ~ $ ls
2.py          lab07.sh      public_html
4.py          lab07.sh~    __pycache__
abc1          may          PycharmProjects
australia     mkdir        reports
bin           monthly     savurskaya.github.io
conf.txt      my_os       script.sh
-            -          -
```

Рис. 3.1: создание файла script.sh

2. Открываем этот файл и пишем там нужный код.



```
1 #!/bin/bash
2 mkdir ~/backup
3 cp script.sh ~/backup/backup.sh
4 gzip ~/backup/backup.sh
```

Рис. 3.2: пишем код

3. В домашней директории появляется папка backup. Внутри нее лежит заархивированный файл.

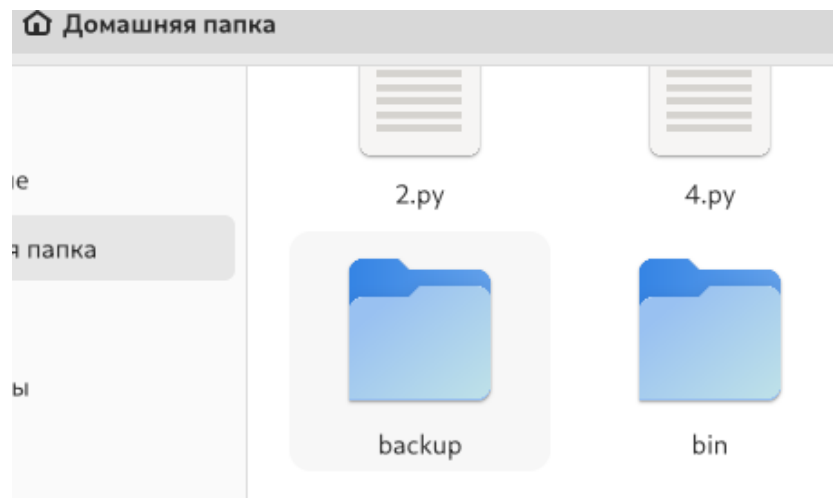


Рис. 3.3: папка backup



Рис. 3.4: заархивированный файл

4. Создаем файл script2.sh. Задаем ему необходимые разрешения. Проверяем, появился ли у нас этот файл.

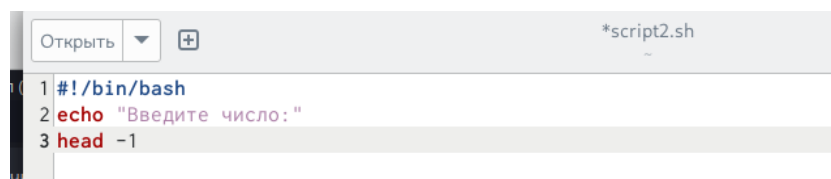
```

pasavurskaya@dk8n60 ~ $ touch script2.sh
pasavurskaya@dk8n60 ~ $ chmod +x script2.sh
pasavurskaya@dk8n60 ~ $ ls
2.py          lab07.sh~      reports
4.py          may            savurskaya.github.io
abc1         mkdir          script2.sh

```

Рис. 3.5: создание файла script2.sh

5. Открываем этот файл и пишем там нужный код.



```

1 #!/bin/bash
2 echo "Введите число:"
3 read -1

```

Рис. 3.6: пишем код

6. Запускаем файл script2.sh и вводим числа. Они выводятся в таком же порядке. Все сделано правильно.

```

pasavurskaya@dk8n60 ~ $ ./script2.sh
Введите число:
8 7 6
8 7 6
pasavurskaya@dk8n60 ~ $ ./script2.sh
Введите число:
22 34
22 34

```

Рис. 3.7: запуск файла script2.sh

7. Создаем файл script3.sh. Задаем ему необходимые разрешения. Проверяем, появился ли у нас этот файл.

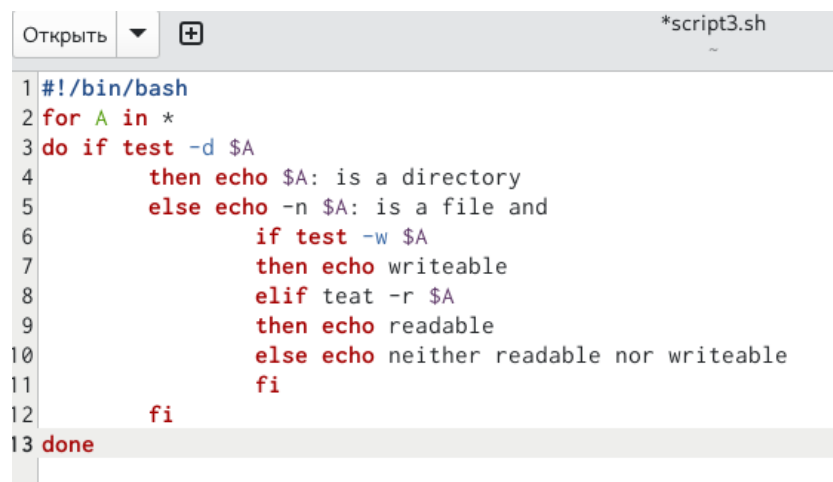

```

pasavurskaya@dk8n60 ~ $ touch script3.sh
pasavurskaya@dk8n60 ~ $ chmod +x script3.sh
pasavurskaya@dk8n60 ~ $ ls
2.py          may          script2.sh
4.py          mkdir        script3.sh
..            ..          ..

```

Рис. 3.8: создание файла script3.sh

8. Открываем этот файл и пишем там нужный код.



```

1 #!/bin/bash
2 for A in *
3 do if test -d $A
4     then echo $A: is a directory
5     else echo -n $A: is a file and
6         if test -w $A
7         then echo writeable
8         elif test -r $A
9         then echo readable
10        else echo neither readable nor writeable
11        fi
12    fi
13 done

```

Рис. 3.9: пишем код

9. Запускаем файл script3.sh. Он выдает информацию о нужном каталоге и выводит информацию о возможностях доступа к файлам этого каталога.

```
pasavurskaya@dk8n60 ~ $ ./script3.sh
2.py: is a file andwriteable
4.py: is a file andwriteable
abc1: is a file andwriteable
australia: is a directory
backup: is a directory
bin: is a directory
conf.txt: is a file andwriteable
feathers: is a file andwriteable
file.txt: is a file andwriteable
```

Рис. 3.10: запуск файла script3.sh

10. Создаем файл script4.sh. Задаем ему необходимые разрешения. Проверяем, появился ли у нас этот файл.

```
pasavurskaya@dk8n60 ~ $ touch script4.sh
pasavurskaya@dk8n60 ~ $ chmod +x script4.sh
pasavurskaya@dk8n60 ~ $ ls
2.py      may      script2.sh
4.py      mkdir    script3.sh
abc1      monthly  script4.sh
australia my_os    script.sh
. . .
```

Рис. 3.11: создание файла script4.sh

11. Открываем этот файл и пишем там нужный код.

```
script4.sh - GNU Emacs at dk8n6
File Edit Options Buffers Tools Sh-Script Outline
#!/bin/bash
direct=''
form=''
echo 'напишите формат'
read form
echo 'напишите директорию'
read direct
find "$direct" -name ".$form" -type f | wc -l
ls
```

Рис. 3.12: пишем код

12. Запускаем файл script4.sh. Он получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории.

```
pasavurskaya@dk8n60 ~ $ ./script4.sh
напишите формат
png
напишите директорию
work
0
2.py      may      script2.sh  Виде
4.py      mkdir    script3.sh  Доку
abc1      monthly  script4.sh  Загр
australia my_os     script.sh   зада
backup    pasavurskaya  ski.places  Изоб
```

Рис. 3.13: запуск файла script4.sh

4 Выводы

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