

Липецкий государственный технический университет

Факультет автоматизации и информатики
Кафедра прикладной математики

Отчет по лабораторной работе № 4
по дисциплине «Операционная система Linux»
Тема «Shell»

Студент

подпись, дата

Егорова М.Р.
фамилия, инициалы

Группа ПМ-20-2

Руководитель
учёная степень, учёное звание

подпись, дата

Кургасов В.В.
фамилия, инициалы

Липецк 2022 г.

Содержание

1. Цель работы	3
2. Задание кафедры	4
3. Задания	7

1. Цель работы

изучение основных возможностей языка программирования Shell с целью автоматизации процесса администрирования системы за счет написания и использования командных файлов.

2. Задание кафедры

1. Используя команды ECHO, PRINTF вывести информационные сообщения на экран.
2. Присвоить переменной A целочисленное значение. Просмотреть значение переменной A.
3. Присвоить переменной B значение переменной A. Просмотреть значение переменной B.
4. Присвоить переменной C значение “путь до своего каталога”. Перейти в этот каталог с использованием переменной.
5. Присвоить переменной D значение “имя команды”, а именно, команды DATE. Выполнить эту команду, используя значение переменной.
6. Присвоить переменной E значение “имя команды”, а именно, команды просмотра содержимого файла, просмотреть содержимое переменной. Выполнить эту команду, используя значение переменной.
7. Присвоить переменной F значение “имя команды”, а именно сортировки содержимого текстового файла. Выполнить эту команду, используя значение переменной. Написать скрипты, при запуске которых выполняются следующие действия:
8. Программа запрашивает значение переменной, а затем выводит значение этой переменной.
9. Программа запрашивает имя пользователя, затем здоровается с ним, используя значение введенной переменной.
10. Программа запрашивает значения двух переменных, вычисляет сумму (разность, произведение, деление) этих перемен-

ных. Результат выводится на экран (использовать команды а) EXPR; б) BC).,

11. Вычислить объем цилиндра. Исходные данные запрашиваются программой. Результат выводится на экран.
12. Используя позиционные параметры, отобразить имя программы, количество аргументов командной строки, значение каждого аргумента командной строки.
13. Используя позиционный параметр, отобразить содержимое текстового файла, указанного в качестве аргумента командной строки. После паузы экран очищается.
14. Используя оператор FOR, отобразить содержимое текстовых файлов текущего каталога поэкранно.
15. Программой запрашивается ввод числа, значение которого затем сравнивается с допустимым значением. В результате этого сравнения на экран выдаются соответствующие сообщения.
16. Программой запрашивается год, определяется, високосный ли он. Результат выдается на экран.
17. Вводятся целочисленные значения двух переменных. Вводится диапазон данных. Пока значения переменных находятся в указанном диапазоне, их значения инкрементируются.
18. В качестве аргумента командной строки указывается пароль. Если пароль введен верно, постранично отображается в длинном формате с указанием скрытых файлов содержимое каталога /etc.
19. Проверить, существует ли файл. Если да, выводится на экран его содержимое, если нет - выдается соответствующее сообщение.

20. Если файл есть каталог и этот каталог можно читать, просматривается содержимое этого каталога. Если каталог отсутствует, он создается. Если файл не есть каталог, просматривается содержимое файла.
21. Анализируются атрибуты файла. Если первый файл существует и используется для чтения, а второй файл существует и используется для записи, то содержимое первого файла перенаправляется во второй файл. В случае несовпадений указанных атрибутов или отсутствия файлов на экран выдаются соответствующие сообщения (использовать а) имена файлов; б) позиционные параметры).
22. Если файл запуска программы найден, программа запускается (по выбору).
23. В качестве позиционного параметра задается файл, анализируется его размер. Если размер файла больше нуля, содержимое файла сортируется по первому столбцу по возрастанию, отсортированная информация помещается в другой файл, содержимое которого затем отображается на экране.
24. Командой TAR осуществляется сборка всех текстовых файлов текущего каталога в один архивный файл `my.tar`, после паузы просматривается содержимое файла `my.tar`, затем командой GZIP архивный файл `my.tar` сжимается.
25. Написать скрипт с использованием функции, например, функции, суммирующей значения двух переменных. Все скрипты выполнить, содержимое письменно отобразить в отчете по лабораторной работе.

3. Задания

```
masha@debian:~$ a='Hello'  
masha@debian:~$ echo $a  
Hello  
masha@debian:~$ printf $a  
Hellomasha@debian:~$
```

Рис. 1 – 1 пункт

```
masha@debian:~$ A=3  
masha@debian:~$ echo $A  
3
```

Рис. 2 – 2 пункт

```
masha@debian:~$ B=$A  
masha@debian:~$ echo $B  
3
```

Рис. 3 – 3 пункт


```
masha@debian:/home$ C=/home/masha
masha@debian:/home$ pwd
/home
masha@debian:/home$ cd $C
masha@debian:~$ pwd
/home/masha
```

Рис. 4 – 4 пункт

```
masha@debian:~$ D=`date`
masha@debian:~$ echo $D
Чт 01 дек 2022 16:31:00 MSK
```

Рис. 5 – 5 пункт

```
masha@debian:~$ E=`cat text/text1.txt`
masha@debian:~$ echo $E
Привет Апельсин Мандаринч
```

Рис. 6 – 6 пункт

```
masha@debian:~$ F=`sort text/text1.txt`
masha@debian:~$ echo $F
Апельсин Мандаринч Привет
```

Рис. 7 – 7 пункт

```
GNU nano 5.4
#!/bin/bash
read -p "A=" A
echo $A
```

Рис. 8 – 8 пункт код скрипта

```
masha@debian:~$ chmod ugo+x script.txt
masha@debian:~$ ./script.txt
A=3
3
masha@debian:~$
```

Рис. 9 – 8 пункт результат выполнения

```
#!/bin/bash
read -p "Введите имя пользователя: " A
echo "Здравствуйтесь, $A!"
```

Рис. 10 – 9 пункт код скрипта

```
masha@debian:~$ ./script.txt
Введите имя пользователя: Мария
Здравствуйтесь, Мария!
masha@debian:~$
```

Рис. 11 – 9 пункт результат выполнения

```
#!/bin/bash
read -p "Введите x: " x
read -p "Введите y: " y
a=`expr $x - $y`;
echo "x-y = "$a
b=`expr $x '*' $y`;
echo "x*y = "$b
c=`expr $x / $y`;
echo "x/y = "$c
```

Рис. 12 – 10 пункт код скрипта с использованием expr

```
masha@debian:~$ ./script.txt
Введите x: 8
Введите y: 4
x-y = 4
x*y = 32
x/y = 2
```

Рис. 13 – 10 пункт результат выполнения с использованием expr

```
#!/bin/bash
read -p "Введите x: " x
read -p "Введите y: " y
echo "$x-$y" | bc
echo "$x * $y" | bc
echo "$x/$y" | bc
```

Рис. 14 – 10 пункт код скрипта с использованием bc

```
masha@debian:~$ ./script.txt
Введите x: 9
Введите y: 3
6
27
3
masha@debian:~$
```

Рис. 15 – 10 пункт результат выполнения с использованием bc

```
#!/bin/bash
read -p "Введите R: " R
read -p "Введите h: " h
pi=3.14
echo "$pi*$R*$R*$h" | bc
```

Рис. 16 – 11 пункт код скрипта

```
masha@debian:~$ ./script.txt
Введите R: 10
Введите h: 5
1570.00
```

Рис. 17 – 11 пункт результат выполнения

```
#!/bin/bash
echo $0
echo $1
echo $2
echo $3
```

Рис. 18 – 12 пункт код скрипта

```
masha@debian:~$ ./script.txt 9 10 7
./script.txt
9
10
7
```

Рис. 19 – 12 пункт результат выполнения

```
#!/bin/bash
more $1
sleep 5s
clear
```

Рис. 20 – 13 пункт код скрипта

```
masha@debian:~$ ./script.txt text/text1.txt
Привет
Апельсин
Мандаринч
_
```

Рис. 21 – 13 пункт результат выполнения

```
#!/bin/bash
for i in *.txt
do
echo $i\n
more $i
done
```

Рис. 22 – 14 пункт код скрипта

```
masha@debian:~$ ./script.txt
li.txtn
Hello
script.txtn
#!/bin/bash
for i in *.txt
do
echo $i\n
more $i
done
masha@debian:~$
```

Рис. 23 – 14 пункт результат выполнения

```
#!/bin/bash
read -p "Введите число от 1 до 10: " x
if [ $x -lt 1 ] || [ $x -gt 10 ]
then
echo "Число введено неверно"
else
echo "Число введено правильно"
fi
```

Рис. 24 – 15 пункт код скрипта

```
masha@debian:~$ ./script.txt
Введите число от 1 до 10: 5
Число введено правильно
masha@debian:~$
```

Рис. 25 – 15 пункт результат выполнения

```
masha@debian:~$ ./script.txt
Введите число от 1 до 10: 77
Число введено неверно
```

Рис. 26 – 15 пункт результат выполнения

```
#!/bin/bash
read -p "Введите год: " x
if [  $$(($x \% 4))$  -eq 0 ]
then
echo "Год високосный"
else
echo "Год не високосный"
fi
```

Рис. 27 – 16 пункт код скрипта

```
masha@debian:~$ sh script.txt
Введите год: 2012
Год високосный
masha@debian:~$ sh script.txt
Введите год: 2014
Год не високосный
```

Рис. 28 – 16 пункт результат выполнения


```
#!/bin/bash
read -p "Введите x : " x
read -p "Введите y : " y
read -p "Введите начало диапазона: " a
read -p "Введите конец диапазона: " b
if [ $x -ge $a ] && [ $y -ge $a ]
then
while [ $x -lt $b ] && [ $y -lt $b ]
do
x=$((x+1))
y=$((y+1))
echo "$x, $y"
done
else
echo "Числа не входят в диапазон"
fi
```

Рис. 29 – 17 пункт код скрипта

```
masha@debian:~$ ./script.txt
Введите x : 2
Введите y : 10
Введите начало диапазона: 2
Введите конец диапазона: 18
3, 11
4, 12
5, 13
6, 14
7, 15
8, 16
9, 17
10, 18
```

Рис. 30 – 17 пункт результат выполнения

```
#!/bin/bash
password="29122002"
if [ $1=$password ]
then
ls -la po | more
else
echo "Неправильный пароль"
fi
```

Рис. 31 – 18 пункт код скрипта

```

masha@debian:~$ ./script.txt
итого 12
drwxr-xr-x 3 masha masha 4096 ноя 30 22:43 .
drwxr-xr-x 6 masha masha 4096 дек 15 13:17 ..
drwxr-xr-x 3 masha masha 4096 ноя 30 22:44 ta

```

Рис. 32 – 18 пункт результат выполнения

```

#!/bin/bash
if [ -f $1 ]
then
more $1
else
echo "Нет такого файла"
fi

```

Рис. 33 – 19 пункт код скрипта

```

masha@debian:~$ ./script.txt text/text1.txt
Привет
Апельсин
Мандаринч

```

Рис. 34 – 19 пункт результат выполнения

```
#!/bin/bash
if [ -d $1 ]
then
if [ -r $1 ]
then
ls -la $1
else
echo "Каталог не доступен для чтения"
fi
else
if [ -f $1 ] && [ -r $1 ]
then
more $1
else
mkdir -p $1
fi
fi
```

Рис. 35 – 20 пункт код скрипта

```
masha@debian:~$ ./script.txt text
итого 12
drwxr-xr-x 2 masha masha 4096 дек 14 18:08 .
drwxr-xr-x 6 masha masha 4096 дек 15 13:26 ..
-rw-r--r-- 1 masha masha  49 дек 14 13:15 text1.txt
masha@debian:~$ ./script.txt text/text1.txt
Привет
Апельсин
Мандаринч
masha@debian:~$ ./script.txt 2
masha@debian:~$ ls
1 2 channel li.txt loop loop1 loop2 loop3 out.gz
```

Рис. 36 – 20 пункт результат выполнения

```
#!/bin/bash
if [ -r $1 ]
then
if [ -w $2 ]
then
cat $1 >> $2
else
echo "Файл 2 не удовлетворяет условиям"
fi
else
echo "Файл 1 не удовлетворяет условиям"
fi
```

Рис. 37 – 21 пункт код скрипта

```
masha@debian:~$ chmod ugo+r li.txt
masha@debian:~$ touch write.txt
masha@debian:~$ chmod ugo+w write.txt
masha@debian:~$ ./script.txt li.txt write.txt
masha@debian:~$ cat write.txt
Hello
masha@debian:~$ cat li.txt
Hello
```

Рис. 38 – 21 пункт результат выполнения

```
#!/bin/bash
read -p "Введите название файла: " x
file=`ls`
for fop in $file
do
if [ -x $fop -a -s $fop -a $fop=$x ]
then
sh $x
fi
done
```

Рис. 39 – 22 пункт код скрипта

```
masha@debian:~$ sh script1
Введите название файла: script.txt
Введите год: 2012
Год високосный
```

Рис. 40 – 22 пункт результат выполнения

```
#!/bin/bash
if [ -f $1 ]
then
size=`stat -c %s $1`
if [ $size -gt 0 ]
then
sort $1 >> $2
more $2
else
echo "Файл пуст"
fi
fi
```

Рис. 41 – 23 пункт код скрипта

```
masha@debian:~$ ./script.txt li.txt write.txt
0
2
3
5
5
6
7
8
9
masha@debian:~$ cat li.txt
8
5
0
3
5
7
2
9
6
```

Рис. 42 – 23 пункт результат выполнения

```
#!/bin/bash
tar -cf my.tar *.txt
sleep 5s
tar -tf my.tar
sleep 5s
gzip my.tar
```

Рис. 43 – 24 пункт код скрипта


```

masha@debian:~$ ./script.txt
li.txt
script.txt
write.txt
masha@debian:~$ ls
1 channel loop loop2 my.tar.gz out.tar script.txt write.txt
2 li.txt loop1 loop3 out.gz po text

```

Рис. 44 – 24 пункт результат выполнения

```

#!/bin/bash
abs(){
if [ $1 -lt 0 ]
then
echo "$(($1 * (-1)))"
else
echo $1
fi
}
abs $1

```

Рис. 45 – 25 пункт код скрипта

```

masha@debian:~$ sh script1 5
5
masha@debian:~$ sh script1 -9
9

```

Рис. 46 – 25 пункт результат выполнения