

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа № 4

по OS Linux

Программирование на SHELL. Использование командных файлов

Студент

Комаричев А. В.

Группа АИ-19

Руководитель

Кургасов В. В.

Липецк 2021г.

Оглавление

Цель работы.....	3
Ход работы.....	4
Скрипты, при запуске которых выполняются следующие действия:	5

Цель работы

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

Ход работы

1. Используя команды ECHO, PRINTF вывести информационные сообщения на экран.

```
alex@alexserver:~$ echo Hello, World!  
Hello, World!  
alex@alexserver:~$ printf "Hello, World! \n"  
Hello, World!
```

Рисунок 1 – Задание 1

2. Присвоить переменной А целочисленное значение. Просмотреть значение переменной А.

```
alex@alexserver:~$ A=10  
alex@alexserver:~$ echo $A  
10
```

Рисунок 2 – Задание 2

3. Присвоить переменной В значение переменной А. Просмотреть значение переменной В.

```
alex@alexserver:~$ B=$A  
alex@alexserver:~$ echo $B  
10
```

Рисунок 3 – Задание 3

4. Присвоить переменной С значение “путь до своего каталога”.
Перейти в этот каталог с использованием переменной.

```
alex@alexserver:~$ C='/bin/'  
alex@alexserver:~$ cd $C  
alex@alexserver:/bin$ _
```

Рисунок 4 – Задание 4

5. Присвоить переменной D значение “имя команды”, а именно, команды DATE. Выполнить эту команду, используя значение переменной.

```
alex@alexserver:~$ D='date'  
alex@alexserver:~$ $D  
Чт 25 ноя 2021 23:45:42 MSK  
alex@alexserver:~$
```

Рисунок 5 – Задание 5

6. Присвоить переменной E значение “имя команды”, а именно, команды просмотра содержимого файла, просмотреть содержимое переменной. Выполнить эту команду, используя значение переменной.

```
alex@alexserver:~$ E='cat loop'
alex@alexserver:~$ echo $E
cat loop
alex@alexserver:~$ $E
while true;
do true;
done
```

Рисунок 6 – Задание 6

7. Присвоить переменной F значение “имя команды”, а именно сортировки содержимого текстового файла. Выполнить эту команду, используя значение переменной.

```
alex@alexserver:~$ F='sort loop'
alex@alexserver:~$ $F
done
do true;
while true;
```

Рисунок 7 – Задание 7

Скрипты, при запуске которых выполняются следующие действия:

8. Программа запрашивает значение переменной, а затем выводит значение этой переменной.

```
alex@alexserver:~$ sh myScript
Hello, Alex!
Hello, Alex!
alex@alexserver:~$ cat myScript
read value; echo $value;
```

Рисунок 8 – Задание 8

9. Программа запрашивает имя пользователя, затем здоровается с ним, используя значение введенной переменной.

```
alex@alexserver:~$ sh myScript
Hello alex, how are you?
alex@alexserver:~$ cat myScript
value=$(whoami); echo "Hello $value, how are you?";
```

Рисунок 9 – Задание 9

10. Программа запрашивает значения двух переменных, вычисляет сумму (разность, произведение, деление) этих переменных. Результат выводится на экран (использовать команды а) EXPR; б) BC;

```
alex@alexserver:~$ sh myScript
first:
10
second:
2
22
alex@alexserver:~$ cat myScript
echo 'first: '; read first; echo 'second: ';
read second; expr $first \* $second + 2;
```

а)

```
alex@alexserver:~$ sh myScript
first:
1000
second:
3
3002
alex@alexserver:~$ cat myScript
echo 'first: '; read first; echo 'second: ';
read second; echo "$first * $second + 2" | bc;
```

б)

Рисунок 10 – Задание 10

11. Вычислить объем цилиндра. Исходные данные запрашиваются программой. Результат выводится на экран.

```
alex@alexserver:~$ sh myScript
radius: 10
high: 2
628.00
alex@alexserver:~$ cat myScript
printf 'radius: '; read radius; printf 'high: ';
read high; echo "3.14 * $radius * $radius * $high" | bc;
```

Рисунок 11 – Задание 11

12. Используя позиционные параметры, отобразить имя программы, количество аргументов командной строки, значение каждого аргумента командной строки.

```
alex@alexserver:~$ sh myScript 2 3 4 5 6
Programm name: myScript number of args=5, arguments: 2 3 4 5 6
alex@alexserver:~$ cat myScript
printf "Programm name: $0 number of args=$#, arguments: $*\n";
```

Рисунок 12 – Задание 12

13. Используя позиционный параметр, отобразить содержимое текстового файла, указанного в качестве аргумента командной строки. После паузы экран очищается.

```
alex@alexserver:~$ cat myScript
cat $1; sleep 4; clear;
alex@alexserver:~$ sh myScript hello
Hello, World!
```

Рисунок 13 – Задание 13

14. Используя оператор FOR, отобразить содержимое текстовых файлов текущего каталога поэкранно.

```
for file in *  
do  
echo;  
sleep 2;  
more -c $file;  
done
```

Рисунок 14 – Задание 14

15. Программой запрашивается ввод числа, значение которого затем сравнивается с допустимым значением. В результате этого сравнения на экран выдаются соответствующие сообщения.

```
alex@alexserver:~$ sh myScript  
Print a number: 1  
It is less than 10  
alex@alexserver:~$ sh myScript  
Print a number: 11  
It is more than 10  
alex@alexserver:~$ sh myScript  
Print a number: 10  
It is 10!  
alex@alexserver:~$ cat myScript  
printf "Print a number: "; read number;  
if test $number -lt 10  
then  
echo "It is less than 10"  
fi  
  
if test $number -gt 10  
then  
echo "It is more than 10"  
fi  
  
if test $number -eq 10  
then  
echo "It is 10!"  
fi
```

Рисунок 15 – Задание 15

16. Программой запрашивается год, определяется, високосный ли он.

Результат выдается на экран.

```
alex@alexserver:~$ sh myScript 1992
it is leap year
alex@alexserver:~$ sh myScript 1100
it is not
alex@alexserver:~$ sh myScript 1200
it is leap year
alex@alexserver:~$ sh myScript 5
it is not
alex@alexserver:~$ sh myScript 4
it is leap year
alex@alexserver:~$ cat myScript
a=$( expr $1 % 4 );
b=$( expr $1 % 100 );
c=$( expr $1 % 400 );
if test $a -eq 0
then if test $b -eq 0 -a $c -ne 0
    then echo "it is not"
    else echo "it is leap year"
    fi
else echo "it is not"
fi
```

Рисунок 16 – Задание 16

17. Вводятся целочисленные значения двух переменных. Вводится диапазон данных. Пока значения переменных находятся в указанном диапазоне, их значения инкрементируются.

```
alex@alexserver:~$ sh myScript
Arg a: 10
arg b: 11
Range c: 11
range d: 20
a=10, b=11 not in range
alex@alexserver:~$ sh myScript
Arg a: 10
arg b: 11
Range c: 9
range d: 20
a=11, b=12
a=12, b=13
a=13, b=14
a=14, b=15
a=15, b=16
a=16, b=17
a=17, b=18
a=18, b=19
a=19, b=20
alex@alexserver:~$ cat myScript
read -p "Arg a: " a;
read -p "arg b: " b;
read -p "Range c: " c;
read -p "range d: " d;
if test $a -gt $c -a $b -gt $c
then
while test $a -lt $d -a $b -lt $d
do
a=$((a+1))
b=$((b+1))
echo "a=$a, b=$b"
done
else echo "a=$a, b=$b not in range"
fi
```

Рисунок 17 – Задание 17

18. В качестве аргумента командной строки указывается пароль. Если пароль введен верно, постранично отображается в длинном формате с указанием скрытых файлов содержимое каталога /etc.

```
alex@alexserver:~$ cat myScript
password="password";
if test $1 = $password
then ls -la /etc |more
fi
```

Рисунок 18 – Задание 18

19. Проверить, существует ли файл. Если да, выводится на экран его содержимое, если нет - выдается соответствующее сообщение.

```
alex@alexserver:~$ sh myScript hello
Hello, World!
alex@alexserver:~$ sh myScript Hello
File not found
alex@alexserver:~$ cat myScript
if test -f $1
then cat $1
else echo "File not found"
fi
```

Рисунок 19 – Задание 19

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.

```
alex@alexserver:~$ sh myScript hello
Hello, World!
alex@alexserver:~$ sh myScript ./hello
Hello, World!
alex@alexserver:~$ sh myScript Hello
alex@alexserver:~$ dir
hello Hello loop loop2 myScript
alex@alexserver:~$ cat myScript
if test -d $1
then if test -r $1
then ls -la $1
else echo "Cannot read"
fi
else if test -f $1 -a -r $1
then more $1
else mkdir $1
fi
fi
```

Рисунок 20 – Задание 20

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).

```
alex@alexserver:~$ sh myScript
alex@alexserver:~$ cat hello
Hello, World!
alex@alexserver:~$ cat loop2
Hello, World!
alex@alexserver:~$ cat myScript
first="hello";
second="loop2";
if test -r $first
then if test -w $second
    then cat $first>$second
    else echo "cannot open file 2"
    fi
else echo "cannot open file 1"
fi
```

а) имена файлов

```
alex@alexserver:~$ sh myScript hello loop
alex@alexserver:~$ cat hello
Hello, World!
alex@alexserver:~$ cat loop
Hello, World!
alex@alexserver:~$ cat myScript
if test -r $1
then if test -w $2
    then cat $1>$2
    else echo "cannot open file 2"
    fi
else echo "cannot open file 1"
fi
```

б) позиционные параметры

Рисунок 21 – Задание 21

22. Если файл запуска программы найден, программа запускается (по выбору).

```
alex@alexserver:~$ sh myScript ./sayHello
Hello, World!
alex@alexserver:~$ cat myScript
if test -f $1
then $1
else echo "cannot execute programm"
fi
```

Рисунок 22 – Задание 22

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.

```
alex@alexserver:~$ sh myScript loop loop2
File size = 14
1
2
3
4
5
6
7
alex@alexserver:~$ cat loop
2
4
3
1
5
7
6
alex@alexserver:~$ cat myScript
size=$( stat -c %s $1);
echo "File size = $size";
if test $size -gt 0
then sort $1>$2; more $2;
else echo "Empty file";
fi
```

Рисунок 23 – Задание 23

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.

```
alex@alexserver:~$ ls -l
total 36
drwxrwxr-x 2 alex alex 4096 ноя 26 12:29 Hello
-rw-rw-r-- 1 alex alex 14 ноя 26 14:38 hello.txt
-rw-rw-r-- 1 alex alex 14 ноя 26 14:13 loop
-rw-rw-r-- 1 alex alex 14 ноя 26 14:25 loop2
-rw-rw-r-- 1 alex alex 14 ноя 26 14:39 loop2.txt
-rw-rw-r-- 1 alex alex 15 ноя 26 14:39 loop.txt
-rw-rw-r-- 1 alex alex 68 ноя 26 14:37 myScript
-rw-rw-r-- 1 alex alex 339 ноя 26 14:35 my.tar.gz
-rwxrwxr-- 1 alex alex 21 ноя 26 12:50 sayHello
alex@alexserver:~$ sh myScript
hello.txt
loop.txt
loop2.txt
gzip: my.tar.gz already exists; do you wish to overwrite (y or n)? y
alex@alexserver:~$ cat myScript
tar -cf my.tar *txt;
sleep 4;
tar -tf my.tar;
sleep 4;
gzip my.tar;
alex@alexserver:~$ _
```

Рисунок 24 – Задание 24

25. Написать скрипт с использованием функции, например, функции, суммирующей значения двух переменных.

```
alex@alexserver:~$ sh myScript 10 2
10 * 2 + 2 = 22
alex@alexserver:~$ sh myScript 1000 3
1000 * 3 + 2 = 3002
alex@alexserver:~$ cat myScript
function() {
    result=$(( $1 * $2 + 2 ));
    return $result;
}
function $1 $2;
echo "$1 * $2 + 2 = $result";
```

Рисунок 25 – Задание 25