Липецкий государственный технический университет

Отчет по Лабораторной работе № 5
по дисциплине «Операционная система Linux»
на тему «Программирование на SHELL. Использование командных файлов»

Студент		Елфимова Д.А.
	подпись, дата	фамилия, инициалы
Руководитель доцент, к.п.н.		Кургасов В.В.
учёная степень, учёное звание	подпись, дата	фамилия, инициалы

Липецк 2019 г.

Задание

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 переменных. 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 просмат-

ривается содержимое файла my.tar, затем командой GZIP архивный файл my.tar сжимается.

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

Оглавление

1. I часть 4
2. II Часть 5
Заключение 14

1. I часть

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

```
test@ubuntu:~$ echo Hello, my dear user!
Hello, my dear user!
test@ubuntu:~$ printf 'Best wishes, \n Your computer \n'
Best wishes,
 Your computer
test@ubuntu:~$ _
```

Рисунок 1. Задание - 1

2. Присвоить переменной A целочисленное значение. Просмотреть значение переменной A.

```
test@ubuntu:~$ A=10
test@ubuntu:~$ echo $A
10
```

Рисунок 2. Задание - 2

3. Присвоить переменной B значение переменной A. Просмотреть значение переменной B.

```
test@ubuntu:~$ B=$A
test@ubuntu:~$ echo $B
10
```

Рисунок 3. Задание - 3

4. Присвоить переменной C значение путь до своего каталога. Перейти в этот каталог с использованием переменной.

```
test@ubuntu:~$ C=$PWD
test@ubuntu:~$ echo $C
/home/test
test@ubuntu:~$ cd ..
test@ubuntu:/home$ cd $C
test@ubuntu:~$ _
```

Рисунок 4. Задание - 4

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

```
test@ubuntu:~$ D=date
test@ubuntu:~$ $D
Вт. нояб.  5 09:30:19 MSK 2019
```

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.

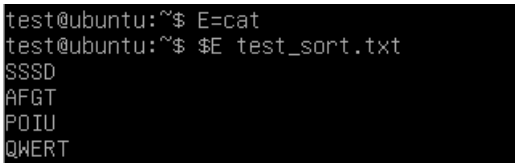

Рисунок 6. Задание - 6

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

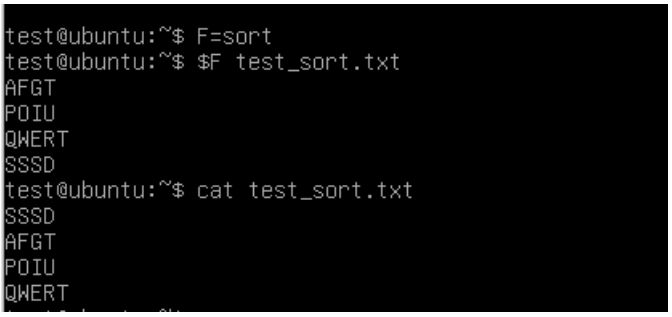

Рисунок 7. Задание - 7

2. II Часть

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. Чтобы сделать файл исполняемым в linux выполните chmod ugo+x файл_ скрипта

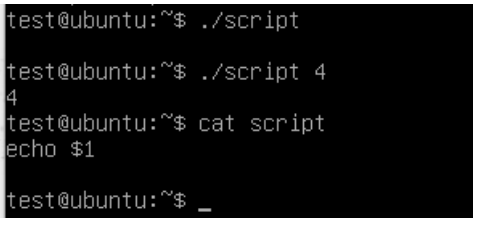

Рисунок 8. Задание - 8

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

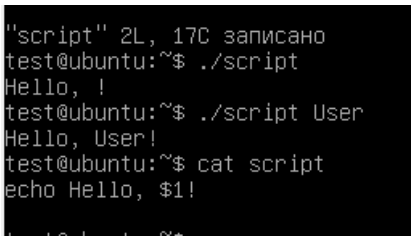

Рисунок 9. Задание - 9

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

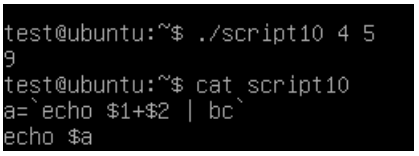

Рисунок 10. Задание - 10

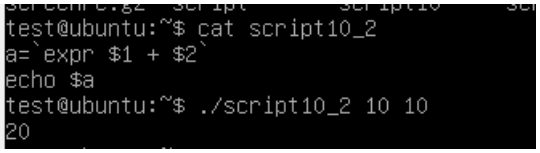

Рисунок 11. Задание - 10(2)

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

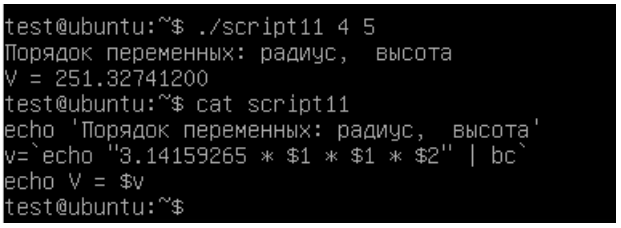

Рисунок 12. Задание - 11

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

```
test@ubuntu:~$ ./script12 1 2 3 4
Имя программы -  ./script12.
Количество аргументов - 4.
Значение каждого аргумента -  1
Значение каждого аргумента -  2
Значение каждого аргумента -  3
Значение каждого аргумента -  4
1 2 3 4
test@ubuntu:~$ cat script12
printf 'Имя программы -  %s. \nКоличество аргументов - %s. \n' $0 $#
printf 'Значение каждого аргумента -  %s \n' $@
echo $@
```

Рисунок 13. Задание - 12

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

```
test@ubuntu:~$ chmod ugo+x script13
test@ubuntu:~$ ./script13 /home/user/1.txt
```

Рисунок 14. Задание - 13

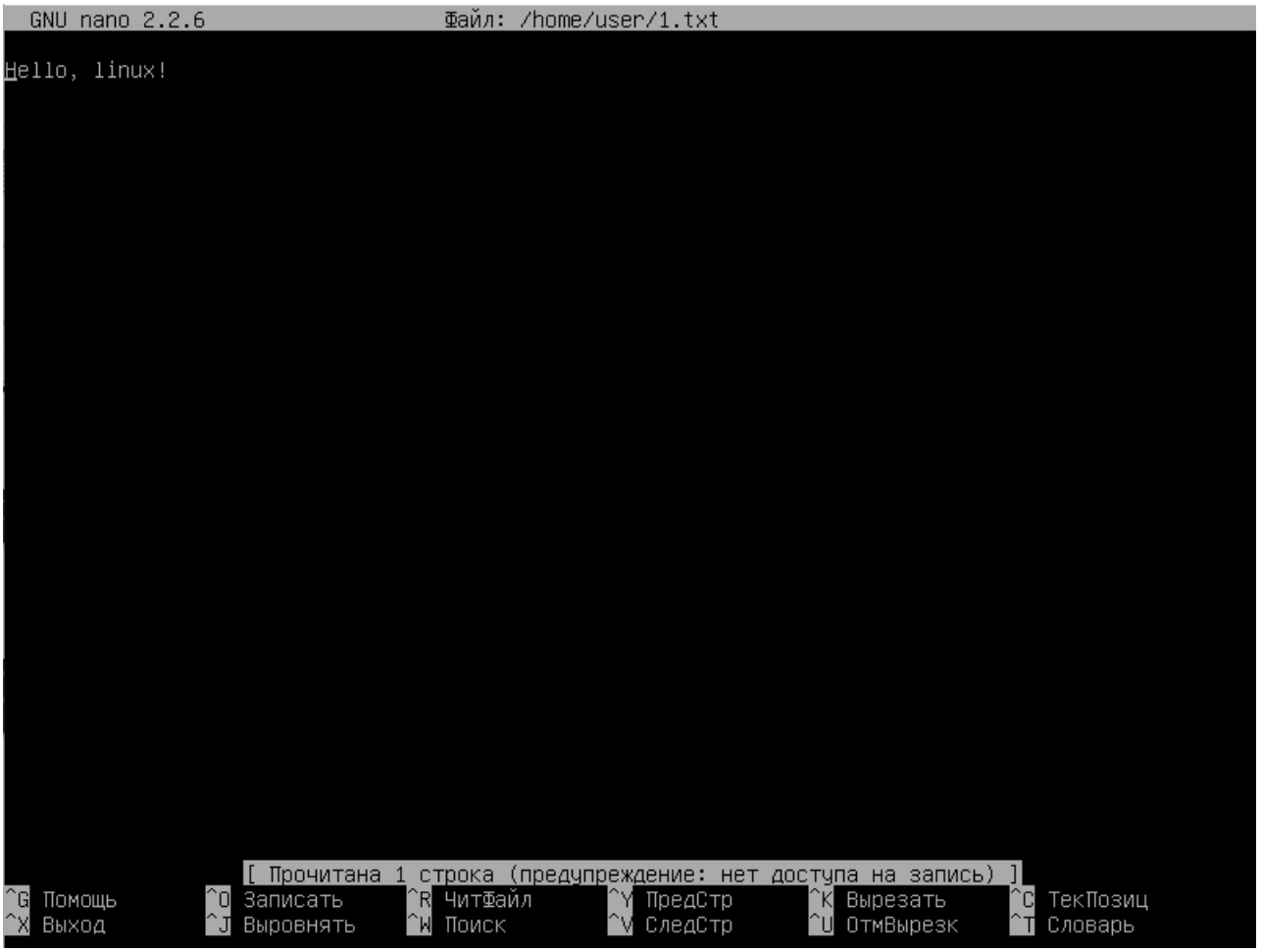

Рисунок 15. Задание - 13(2)

```
test@ubuntu:~$ cat script13
nano $1
test@ubuntu:~$ _
```

Рисунок 16. Задание - 13(3)

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

```
test@ubuntu:~/new$ chmod ugo+x script14
test@ubuntu:~/new$ ./script14
I'm here for another testing. I'll be useful, user)
This is file for testing different things.
test@ubuntu:~/new$ cat script14
for file in ./*.txt
do
cat $file
done
test@ubuntu:~/new$
```

Рисунок 17. Задание - 14

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

```
test@ubuntu:~/new$ ./script15 1
Проверка на диапазон от 1 до 10
Число 1 принадлежит множеству допустимых значений
test@ubuntu:~/new$ ./script15 12
Проверка на диапазон от 1 до 10
Число 12 не удовлетворяет условиям. Выход за границы массива
test@ubuntu:~/new$ cat script15
printf 'Проверка на диапазон от 1 до 10\n'
if (($1 < 1 || $1 > 10))
then
printf 'Число %s не удовлетворяет условиям. Выход за границы массива\n' $1
else
printf 'Число %s принадлежит множеству допустимых значений\n' $1
fi
```

Рисунок 18. Задание - 15

16. Программой запрашивается год, определяется, високосный ли он. Результат выдается на экран.

```
test@ubuntu:~/new$ ./script16
Введите год - 1600
1600 - високосный год
test@ubuntu:~/new$ ./script16
Введите год - 1601
1601 - невисокосный год
test@ubuntu:~/new$ cat script16
read -p "Введите год - " Y
if (($Y % 4 == 0 ))
then
printf '%s - високосный год\n' $Y
else
printf '%s - невисокосный год\n' $Y
fi
```

Рисунок 19. Задание - 16

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

```
test@ubuntu:~/new$ ./script17
Введите целочисленные значения двух переменных (x,y) - 1 2
Введите диапазон данных - 1 5
Значения входят в диапазон. Инкрементируем.
x = 2
y = 3
Значения входят в диапазон. Инкрементируем.
x = 3
y = 4
Значения входят в диапазон. Инкрементируем.
x = 4
y = 5
test@ubuntu:~/new$ cat script17
read -p "Введите целочисленные значения двух переменных (x,y) - " x y
read -p "Введите диапазон данных - " a b
if (($x >= $a && $y >= $a && a < b))
then
while (($x < $b && $y < $b))
do
printf 'Значения входят в диапазон. Инкрементируем.\n'
x=$(($x+1))
y=$(($y+1))
printf 'x = %s\n' $x
printf 'y = %s\n' $y
done
else
printf 'Выход за границы диапазона или неверный диапазон'
fi
```

Рисунок 20. 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.

```
test@ubuntu:~/new$ cat script18
if (($1 == "test"))
then
ls -li /etc | less
else
echo Неверный пароль
fi
test@ubuntu:~/new$ ./script18 'test'
```

Рисунок 21. Задание - 18

```
итого 860
406841 drwxr-xr-x 3 root root    4096 февр.  6  2018 acpi
393225 -rw-r--r-- 1 root root    2981 авг.   3  2016 adduser.conf
393226 drwxr-xr-x 2 root root    4096 апр.  10  2018 alternatives
406705 drwxr-xr-x 6 root root    4096 февр. 13  2018 apm
393278 drwxr-xr-x 3 root root    4096 нояб. 11  2018 apparmor
393282 drwxr-xr-x 9 root root    4096 дек.  19  2018 apparmor.d
406852 drwxr-xr-x 3 root root    4096 нояб. 11  2018 apport
393291 drwxr-xr-x 6 root root    4096 июня   9  2018 apt
406871 -rw-r----- 1 root daemon   144 окт.  21  2013 at.deny
393302 -rw-r--r-- 1 root root    2177 апр.   9  2014 bash.bashrc
405962 -rw-r--r-- 1 root root      45 марта 23  2014 bash_completion
393303 drwxr-xr-x 2 root root    4096 нояб. 11  2018 bash_completion.d
393308 -rw-r--r-- 1 root root     356 янв.   2  2012 bindresvport.blacklist
393309 -rw-r--r-- 1 root root     321 апр.  16  2014 blkid.conf
412442 lrwxrwxrwx 1 root root      15 нояб. 24  2016 blkid.tab -> /dev/.blkid.tab
  8532 drwxr-xr-x 2 root root    4096 февр.  6  2018 byobu
406499 drwxr-xr-x 3 root root    4096 февр.  6  2018 ca-certificates
393591 -rw-r--r-- 1 root root    8464 апр.  10  2018 ca-certificates.conf
393229 -rw-r--r-- 1 root root    7788 февр.  6  2018 ca-certificates.conf.dpkg-old
406304 drwxr-xr-x 2 root root    4096 февр.  6  2018 calendar
406802 drwxr-s--- 2 root dip     4096 нояб. 11  2018 chatscripts
408707 drwxr-xr-x 5 root root    4096 февр. 13  2018 ConsoleKit
393311 drwxr-xr-x 2 root root    4096 февр.  6  2018 console-setup
393344 drwxr-xr-x 2 root root    4096 февр.  6  2018 cron.d
393346 drwxr-xr-x 2 root root    4096 нояб. 11  2018 cron.daily
393353 drwxr-xr-x 2 root root    4096 февр.  6  2018 cron.hourly
393355 drwxr-xr-x 2 root root    4096 февр.  6  2018 cron.monthly
393360 -rw-r--r-- 1 root root     722 февр.  9  2013 crontab
393357 drwxr-xr-x 2 root root    4096 апр.  10  2018 cron.weekly
393361 drwxr-xr-x 4 root root    4096 апр.  10  2018 dbus-1
393365 -rw-r--r-- 1 root root    2969 февр. 23  2014 debconf.conf
393366 -rw-r--r-- 1 root root      11 февр. 20  2014 debian_version
393367 drwxr-xr-x 2 root root    4096 нояб. 11  2018 default
393378 -rw-r--r-- 1 root root     604 нояб.  7  2013 deluser.conf
393379 drwxr-xr-x 2 root root    4096 мая   17  2018 depmod.d
:
```

Рисунок 22. Задание - 18(2)

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

```
test@ubuntu:~/new$ ./script19 /etc/bebebe
Файл не существует
test@ubuntu:~/new$ ./script19 /home/test/new/test.txt
файл существует
This is file for testing different things.
test@ubuntu:~/new$ cat script19
if [ -f $1 ]
then
echo "файл существует"
cat $1
else
echo "Файл не существует"
fi
test@ubuntu:~/new$
```

Рисунок 23. 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.

```
test@ubuntu:~/new$ ./script20 /home/test/new/dirdir
Это каталог и его можно читать
test@ubuntu:~/new$
test@ubuntu:~/new$ ./script20 /home/test/new
Это каталог и его можно читать
10]   dirdir  script14  script16  script18  script20   test.txt
10]]  scr     script15  script17  script19  test2.txt
test@ubuntu:~/new$ ./script20 /home/test/new/test.txt
Это файл
This is file for testing different things.
test@ubuntu:~/new$ ./script20 /home/test/new/dir
./script20: строка 10: Ни каталога ни файла нет. Создадим: команда не найдена
test@ubuntu:~/new$ ./script20 /home/test/new/dir
Это каталог и его можно читать
test@ubuntu:~/new$ cat script20
if [ -d $1 ]
then
echo "Это каталог и его можно читать"
ls $1
elif [ -f $1 ]
then
echo "Это файл"
cat $1
else
"Ни каталога ни файла нет. Создадим"
mkdir -p $1
fi
```

Рисунок 24. 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).

```
test@ubuntu:~/new$ ./script21 test.txt test2.txt
Переписываем...
test@ubuntu:~/new$ cat script21
if test -r $1 -a -w $2
then
echo "Переписываем..."
cat $1 > $2
elif ! -r $1 -a -w $2
then
echo "Первый файл недоступен для чтения"
elif -r $1 && -a ! -w $2
then
echo "Второй файл недоступен для записи"
else
echo "Все очень плохо"
fi
```

Рисунок 25. Задание - 21

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

```
test@ubuntu:~/new$ ./script22 ./script17
Программа найдена. Запустить? 1/0 - 1
Введите целочисленные значения двух переменных (x,y) - 1 1
Введите диапазон данных - 1 2
Значения входят в диапазон. Инкрементируем.
x = 2
y = 2
test@ubuntu:~/new$
test@ubuntu:~/new$ ./script22 ./s
Не найдена
```

Рисунок 26. Задание - 22

```
if [ -x $1 ]
then
read -p "Программа найдена. Запустить? 1/0 - " C
else
echo "Не найдена"
fi
if (( $C == 1 ))
then
$1
fi
```

Рисунок 27. Задание - 22(2)

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.

```
test@ubuntu:~/new$ cat test.txt
One One Two
Abc der tru
WWW WWW QQQ

test@ubuntu:~/new$ ./script23 test.txt test2.txt

Abc der tru
One One Two
WWW WWW QQQ
test@ubuntu:~/new$ cat script23
size=$(wc -c $1 | awk '{print $1}')
if ((size>0))
then
sort -k1 $1
cat $1 > $2
else
echo "Oops, it's zero-file"
fi
```

Рисунок 28. 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.

```
test@ubuntu:~/new$ ./script24
Архивируем
test2.txt
test.txt
Просматриваем
test2.txt
test.txt
Сжимаем
Готово!
test@ubuntu:~/new$ cat script24
echo "Архивируем"
tar -cvf my.tar *.txt
echo "Просматриваем"
tar -tf my.tar
echo "Сжимаем"
gzip -c my.tar > my.gz
echo "Готово!"
test@ubuntu:~/new$ _
```

Рисунок 29. Задание - 24

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

```
test@ubuntu:~/new$ ./script25 1 2
3
test@ubuntu:~/new$ cat script25
func() {
res=`echo $1+$2 | bc`
echo $res
}
func $1 $2
```

Рисунок 30. Задание - 25

Заключение

В ходе данной лабораторной работы были изучены или повторно рассмотрены некоторые команды ОС Linux, было проведено ознакомление и анализ рекомендованной литературы, а также информации о средствах управления процессами ОС Ubuntu. Также было произведено изучение основных возможностей языка программирования Shell с целью автоматизации процесса администрирования системы за счет написания и использования командных файлов.