

Липецкий государственный технический университет

Кафедра прикладной математики

Отчет по лабораторной работе № 4
«Программирование на SHELL. Использование командных
файлов»
по курсу «ОС Linux»

Студент

подпись, дата

Гришагин Е.Е.

фамилия, инициалы

Группа

ПМ-19-2

Руководитель

ученая степень, ученое звание

подпись, дата

Кургасов В.В.

фамилия, инициалы

Липецк 2021 г.

Содержание

Задание кафедры	3
1. Ход работы	6
1.1. Задание 1	6
1.2. Задание 2	6
1.3. Задание 3	6
1.4. Задание 4	6
1.5. Задание 5	7
1.6. Задание 6	7
1.7. Задание 7	8
1.8. Задание 8	8
1.9. Задание 9	8
1.10. Задание 10	9
1.11. Задание 11	10
1.12. Задание 12	11
1.13. Задание 13	11
1.14. Задание 14	12
1.15. Задание 15	12
1.16. Задание 16	13
1.17. Задание 17	13
1.18. Задание 18	14
1.19. Задание 19	15
1.20. Задание 20	15
1.21. Задание 21	16
1.22. Задание 22	17
1.23. Задание 23	18
1.24. Задание 24	19
1.25. Задание 25	19

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 переменных. Результат выводится на экран (использовать команды а) EXPR; б) BC).,
11. . Вычислить объем цилиндра. Исходные данные запрашиваются программой. Результат выводится на экран.

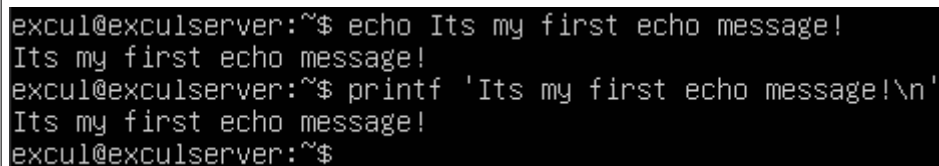
12. Используя позиционные параметры, отобразить имя программы, количество аргументов командной строки, значение каждого аргумента командной строки.
13. Используя позиционный параметр, отобразить содержимое текстового файла, указанного в качестве аргумента командной строки. После паузы экран очищается.
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 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.

1. Ход работы

1.1. Задание 1

Используя команды ECHO, PRINTF вывести информационные сообщения на экран.

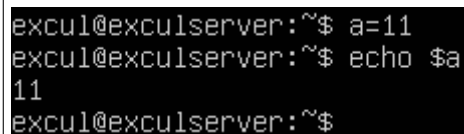


```
excul@exculserver:~$ echo Its my first echo message!  
Its my first echo message!  
excul@exculserver:~$ printf 'Its my first echo message!\n'  
Its my first echo message!  
excul@exculserver:~$
```

Рисунок 1 - Вывод сообщений

1.2. Задание 2

Присвоить переменной А целочисленное значение. Просмотреть значение переменной А.

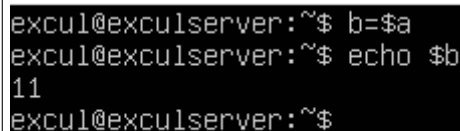


```
excul@exculserver:~$ a=11  
excul@exculserver:~$ echo $a  
11  
excul@exculserver:~$
```

Рисунок 2 - Присвоение и вывод

1.3. Задание 3

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



```
excul@exculserver:~$ b=$a  
excul@exculserver:~$ echo $b  
11  
excul@exculserver:~$
```

Рисунок 3 - Присвоение и вывод

1.4. Задание 4

Присвоить переменной С значение «путь до своего каталога». Перейти в этот каталог с использованием переменной.

```
excul@exculserver:~$ c=$pwd
excul@exculserver:~$ cd ..
excul@exculserver:/home$ cd ..
excul@exculserver:/$ cd $c
excul@exculserver:~$ _
```

Рисунок 4 - Присвоение и переход

1.5. Задание 5

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

```
excul@exculserver:~$ d=date
excul@exculserver:~$ $d
Пт 26 ноя 2021 08:57:29 UTC
excul@exculserver:~$
```

Рисунок 5 - Присвоение даты и вывод

1.6. Задание 6

Присвоить переменной E значение «имя команды», а именно, команды просмотра содержимого файла, просмотреть содержимое переменной. Выполнить эту команду, используя значение переменной.

```
excul@exculserver:~$ e=zcat
excul@exculserver:~$ $e arch.gz
/home:
excul
user

/home/excul:
123
arch.gz
pipe

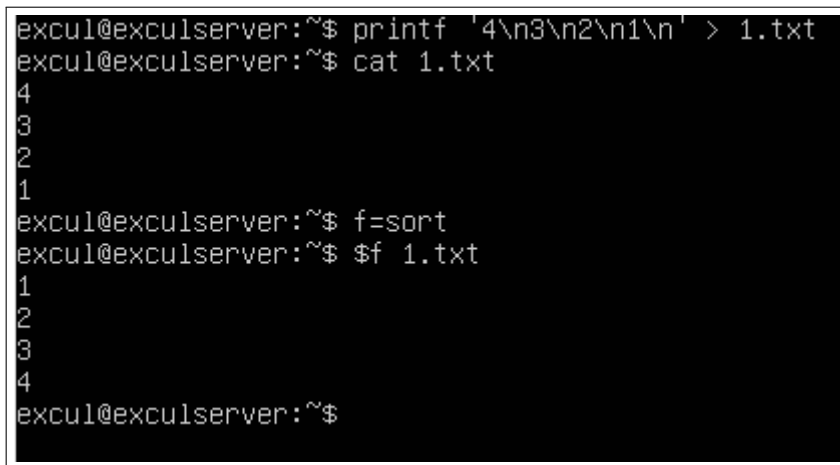
/home/excul/123:
test2.txt
test.txt

/home/user:
1.txt
3.txt
h1
s1
excul@exculserver:~$ _
```

Рисунок 6 - Присвоение переменной команды и вывод содержимого

1.7. Задание 7

Присвоить переменной F значение «имя команды», а именно сортировки содержимого текстового файла. Выполнить эту команду, используя значение переменной

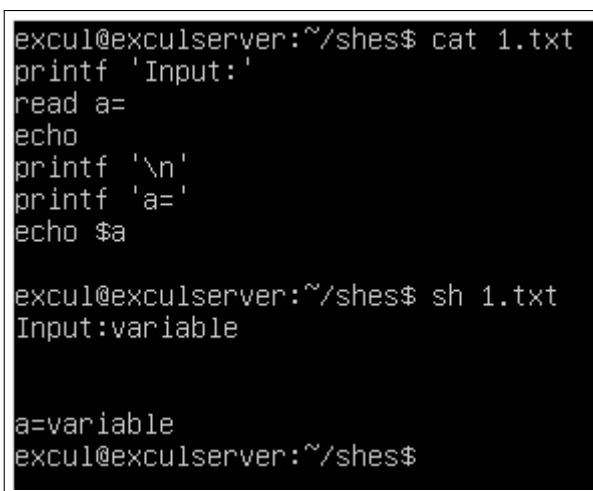


```
excul@exculserver:~$ printf '4\n3\n2\n1\n' > 1.txt
excul@exculserver:~$ cat 1.txt
4
3
2
1
excul@exculserver:~$ f=sort
excul@exculserver:~$ $f 1.txt
1
2
3
4
excul@exculserver:~$
```

Рисунок 7 - Ввод в файл, присвоение переменной команды и использование этой команды

1.8. Задание 8

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



```
excul@exculserver:~/shes$ cat 1.txt
printf 'Input:'
read a=
echo
printf '\n'
printf 'a='
echo $a

excul@exculserver:~/shes$ sh 1.txt
Input:variable

a=variable
excul@exculserver:~/shes$
```

Рисунок 8 - Код программы по заданию 8 и результат работы

1.9. Задание 9

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


```

excul@exculserver:~/shes$ cat 2.txt
printf 'Input your name:'
read a=
echo
printf '\n'
printf 'Greetings, '
echo $a

excul@exculserver:~/shes$ sh 2.txt
Input your name:Eugene

Greetings, Eugene
excul@exculserver:~/shes$

```

Рисунок 9 - Код программы по заданию 9 и результат работы

1.10. Задание 10

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

```

excul@exculserver:~/shes$ cat 3.txt
printf 'Input first number:'
read x=
printf 'Input second number:'
read y
printf 'Sum:'
echo $(expr $x + $y)
printf 'Diff:'
echo $(expr $x - $y)
printf 'Multi:'
echo $(expr $x \* $y)
printf 'Div:'
echo $(expr $x / $y)

excul@exculserver:~/shes$ sh 3.txt
Input first number:7
Input second number:3
Sum:10
Diff:4
Multi:21
Div:2
excul@exculserver:~/shes$

```

Рисунок 10.1 - Код программы по заданию 10 и результат работы (через expr)

```

excul@exculserver:~/shes$ cat 3.1.txt
printf 'Input first number:'
read x=
printf 'Input second number:'
read y
printf 'Sum: '
echo "scale=3; $x + $y" | bc
printf 'Diff: '
echo "scale=3; $x - $y" | bc
printf 'Multi: '
echo "scale=3; $x * $y" | bc
printf 'Div: '
echo "scale=3; $x / $y" | bc

excul@exculserver:~/shes$ sh 3.1.txt
Input first number:7
Input second number:4
Sum: 11
Diff: 3
Multi: 28
Div: 1.750
excul@exculserver:~/shes$

```

Рисунок 10.2 - Код программы по заданию 10 и результат работы (через bc)

1.11. Задание 11

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

```

excul@exculserver:~/shes$ cat 4.txt
printf 'Cylinder volume with precision up to 5\ncharacters after the decimal point\n'
printf 'Input radius: '
read r
printf 'Input heighth: '
read h
printf 'V = '
echo "scale=5; 4*a(1)*$r*$r*$h" | bc -l
excul@exculserver:~/shes$ sh 4.txt
Cylinder volume with precision up to 5
characters after the decimal point
Input radius: 3
Input heighth: 2
V = 56.54808
excul@exculserver:~/shes$ _

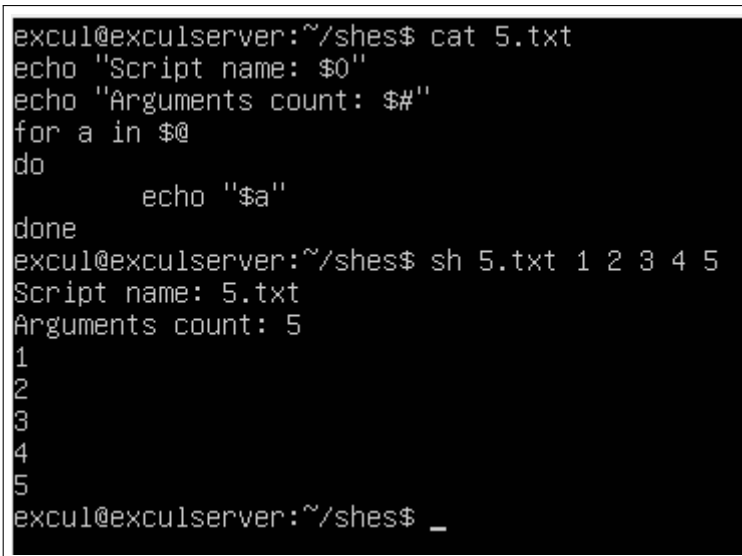
```

Рисунок 11 - Код программы по заданию 11 и результат работы

1.12. Задание 12

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

```
excul@exculserver:~/shes$ cat 5.txt
echo "Script name: $0"
echo "Arguments count: $#"
```



```
for a in $@
do
    echo "$a"
done
excul@exculserver:~/shes$ sh 5.txt 1 2 3 4 5
Script name: 5.txt
Arguments count: 5
1
2
3
4
5
excul@exculserver:~/shes$ _
```

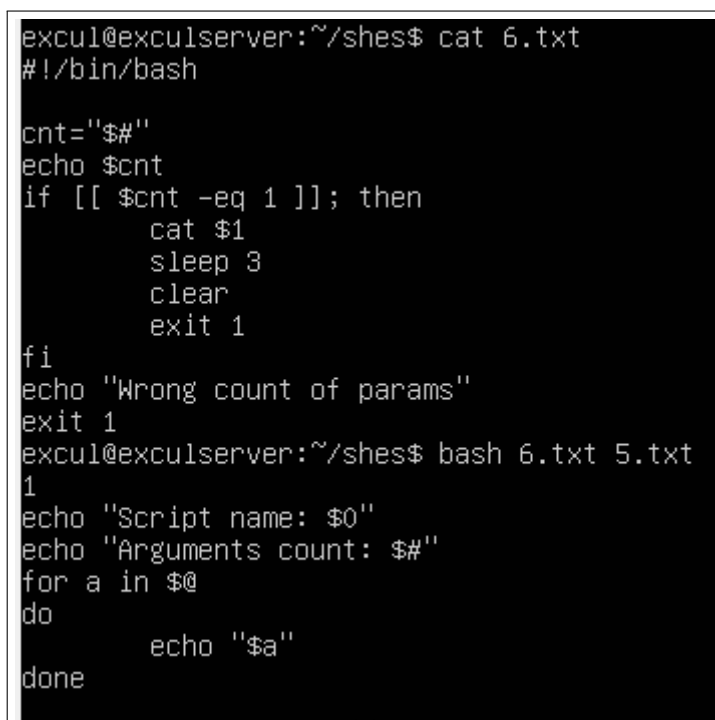
Рисунок 12 - Код программы по заданию 12 и результат работы

1.13. Задание 13

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

```
excul@exculserver:~/shes$ cat 6.txt
#!/bin/bash

cnt="$#"
echo $cnt
if [[ $cnt -eq 1 ]]; then
    cat $1
    sleep 3
    clear
    exit 1
fi
echo "Wrong count of params"
exit 1
excul@exculserver:~/shes$ bash 6.txt 5.txt
1
echo "Script name: $0"
echo "Arguments count: $#"
```



```
for a in $@
do
    echo "$a"
done
```

Рисунок 13 - Код программы по заданию 13 и результат работы

1.14. Задание 14

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

```
excul@exculserver:~/shes$ cat 7.txt
#!/bin/bash

for file in ./*
do
    if [ -f $file ];
    then
        cat $file | less -N
        echo ""
    fi
done
excul@exculserver:~/shes$ sh 7.txt
(END)
```

Рисунок 14 - Код программы по заданию 14 и результат работы

1.15. Задание 15

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

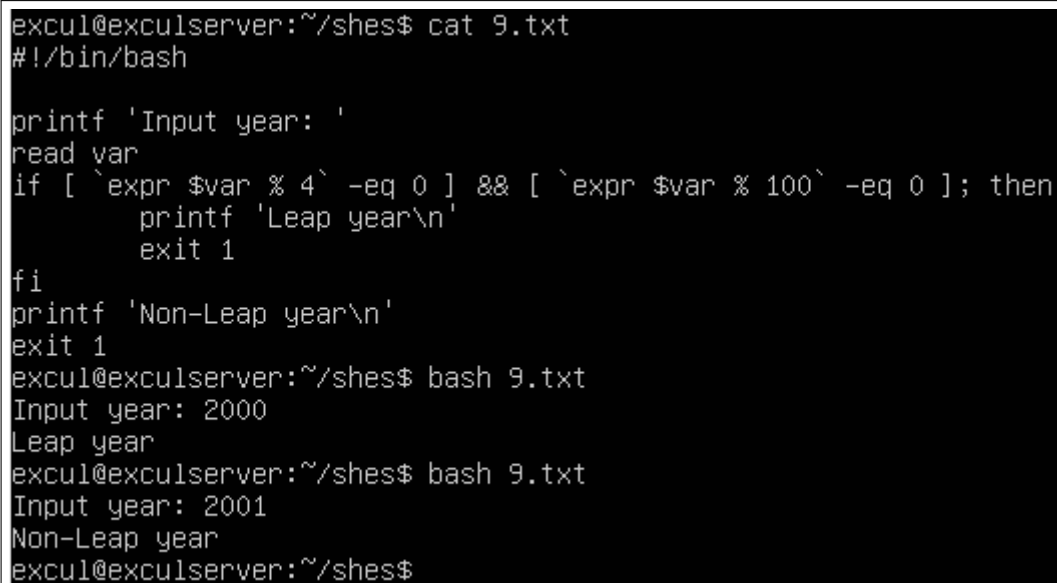
```
excul@exculserver:~/shes$ cat 8.txt
#!/bin/bash

printf 'Input var: '
read var
if [[ $var -eq 11 ]]; then
    printf 'var equal 11\n'
    exit 1
fi
printf 'var not equal 11\n'
exit 1
excul@exculserver:~/shes$ bash 8.txt
Input var: 12
var not equal 11
excul@exculserver:~/shes$ bash 8.txt
Input var: 11
var equal 11
excul@exculserver:~/shes$
```

Рисунок 15 - Код программы по заданию 15 и результат работы

1.16. Задание 16

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



```
excul@exculserver:~/shes$ cat 9.txt
#!/bin/bash

printf 'Input year: '
read var
if [ `expr $var % 4` -eq 0 ] && [ `expr $var % 100` -eq 0 ]; then
    printf 'Leap year\n'
    exit 1
fi
printf 'Non-Leap year\n'
exit 1
excul@exculserver:~/shes$ bash 9.txt
Input year: 2000
Leap year
excul@exculserver:~/shes$ bash 9.txt
Input year: 2001
Non-Leap year
excul@exculserver:~/shes$
```

Рисунок 16 - Код программы по заданию 16 и результат работы

1.17. Задание 17

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

```

excul@exculserver:~/shes$ cat 10.txt
#!/bin/bash
printf 'First var: '
read x
printf 'Second var: '
read y
printf 'Left border of num range: '
read a
printf 'Right border of num range '
read b
while [ $x -le $b ] && [ $x -gt $a ]
do
    (( --x ))
done
while [ $y -le $b ] && [ $y -gt $a ]
do
    (( ++y ))
done
printf 'First var: '
echo $x
printf 'Second var: '
echo $y
excul@exculserver:~/shes$ bash 10.txt
First var: 5
Second var: 6
Left border of num range: 2
Right border of num range 10
First var: 2
Second var: 11
excul@exculserver:~/shes$ _

```

Рисунок 17 - Код программы по заданию 17 и результат работы

1.18. Задание 18

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

```

excul@exculserver:~/shes$ cat 11.txt
#!/bin/bash
if [ $1 = $USER ]; then
    ls -a -l /etc | less
else
    printf 'Wrong username\n'
fi
excul@exculserver:~/shes$ bash 11.txt wrong
Wrong username
excul@exculserver:~/shes$

```

Рисунок 18.1. - Код программы по заданию 18 и результат работы (при вводе неправильного имени)

```
total 812
drwxr-xr-x 97 root root      4096 ноя 19 21:12 .
drwxr-xr-x 20 root root      4096 окт 13 16:30 ..
-rw-r--r--  1 root root     3028 авг 24 08:42 adduser.conf
drwxr-xr-x  2 root root      4096 авг 24 08:47 alternatives
drwxr-xr-x  3 root root      4096 авг 24 08:47 apparmor
drwxr-xr-x  7 root root      4096 авг 24 08:47 apparmor.d
drwxr-xr-x  3 root root      4096 ноя 19 21:12 appport
```

Рисунок 18.2. - Код программы по заданию 18 и результат работы (при вводе правильного имени)

1.19. Задание 19

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

```
excul@exculserver:~/shes$ cat 12.txt
#!/bin/bash
printf 'Input path: '
read path
if [ -s $path ]; then
    cat $path
else
    printf 'File doesnt existing\n'
fi
excul@exculserver:~/shes$ bash 12.txt
Input path: 1.txt
printf 'Input:'
read a=
echo
printf '\n'
printf 'a='
echo $a

excul@exculserver:~/shes$ bash 12.txt
Input path: nonexst.txt
File doesnt existing
excul@exculserver:~/shes$ _
```

Рисунок 19 - Код программы по заданию 19 и результат работы

1.20. Задание 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.

```

excul@exculserver:~/shes$ cat 13.txt
#!/bin/bash
if [ -e $1 ]; then
    echo "Object $1 exists"
    if [ -d $1 ]; then
        echo "$1 is directory"
        if [ -r $1 ]; then
            ls $1
        fi
    fi

    if [ -f $1 ]; then
        echo "$1 is file"
        if [ -r $1 ]; then
            cat $1
        fi
    fi
fi
else
    echo "Object $1 doesnt existing"
    echo "I will create folder with name $1"
    mkdir $1
fi
excul@exculserver:~/shes$ bash 13.txt 1.txt
Object 1.txt exists
1.txt is file
printf 'Input:'
read a=
echo
printf '\n'
printf 'a='
echo $a

excul@exculserver:~/shes$ bash 13.txt 1231231
Object 1231231 doesnt existing
I will create folder with name 1231231
excul@exculserver:~/shes$

```

Рисунок 20.1. - Код программы по заданию 20 и результат работы (при несуществующем файле/папке)

```

excul@exculserver:~/shes$ bash 13.txt 1231231
Object 1231231 doesnt existing
I will create folder with name 1231231
excul@exculserver:~/shes$ ls
1      11.txt  12.txt  1.txt   3.1.txt  4.txt  6.txt  8.txt  dsa
10.txt 1231231 13.txt  2.txt   3.txt   5.txt  7.txt  9.txt  src
excul@exculserver:~/shes$ _

```

Рисунок 20.1. - Проверка

1.21. Задание 21

Анализируются атрибуты файла. Если первый файл существует и используется для чтения, а второй файл существует и используется для за-

писи, 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).

```
excul@exculserver:~/shes$ cat 14.txt
#!/bin/bash
printf 'Input file for read: '
read r
printf 'Input file for write: '
read w
if [ -e $r ]; then
    if [ -r $r ]; then
        if [ -e $w ]; then
            if [ -w $w ]; then
                cat $r > $w
            else
                echo "$w doesnt writeble"
            fi
        else
            echo "$w doesnt exists"
        fi
    else
        echo "$r doesnt readble"
    fi
else
    echo "$r doesnt exists"
fi
excul@exculserver:~/shes$ cat test.txt
123123321
excul@exculserver:~/shes$ cat test2.txt
excul@exculserver:~/shes$ bash 14.txt
Input file for read: test.txt
Input file for write: test2.txt
excul@exculserver:~/shes$ cat test2.txt
123123321
excul@exculserver:~/shes$ _
```

Рисунок 21 - Код программы по заданию 21 и результат работы

1.22. Задание 22

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

```

excul@exculserver:~/shes$ cat 15.xt
#!/bin/bash
printf 'Input programm name: '
read prog
if [ -e $prog ]; then
    printf 'Open programm?\nInput - 0(if you wanna to open $prog)\nOr\nAny other symbol(if no)'
    printf '\nPrint option: '
    read opt
    if [ $opt -eq 0 ]; then
        bash $prog
    else
        exit
    fi
else
    echo "$prog doesnt existing"
fi
excul@exculserver:~/shes$ bash 15.xt
Input programm name: 1.txt
Open programm?
Input - 0(if you wanna to open $prog)
Or
Any other symbol(if no)
Print option: 0
Input:1.txt: line 2: read: `a=': not a valid identifier

a=
excul@exculserver:~/shes$ _

```

Рисунок 22 - Код программы по заданию 22 и результат работы

1.23. Задание 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.

```

excul@exculserver:~/shes$ cat 16.xt
#!/bin/bash
if [ -s $1 ]; then
    sort -k 1 $1 > file_for_write.txt
    cat file_for_write.txt
else
    echo "$1 is empty"
fi
excul@exculserver:~/shes$ bash 16.xt test.txt
test.txt is empty
excul@exculserver:~/shes$ echo '321das' > test.txt
excul@exculserver:~/shes$ bash 16.xt test.txt
321das
excul@exculserver:~/shes$ _

```

Рисунок 23 - Код программы по заданию 23 и результат работы

1.24. Задание 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.

```
excul@exculserver:~/shes$ cat 17.txt
#!/bin/bash
all_txts=$(find . -type f -name "*.txt")
tar -cf txtarch.tar $all_txts
tar -tf txtarch.tar
gzip txtarch.tar
excul@exculserver:~/shes$ bash 17.txt
./2.txt
./3.1.txt
./5.txt
./test2.txt
./9.txt
./17.txt
./12.txt
./14.txt
./13.txt
./1.txt
./8.txt
./6.txt
./7.txt
./4.txt
./file_for_write.txt
./test.txt
./10.txt
./3.txt
./11.txt
excul@exculserver:~/shes$ _
```

Рисунок 24 - Код программы по заданию 24 и результат работы

1.25. Задание 25

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

```
excul@exculserver:~/shes$ cat 25.xt
#!/bin/bash
sum (){
    local var1=$1
    local var2=$2
    local res=`expr $var1 + $var2`
    echo ${res}
}
printf 'Input first num: '
read fn
printf 'Input second num: '
read sn
res=$(sum "$fn" "$sn")
echo $res
exit

excul@exculserver:~/shes$ bash 25.xt
Input first num: 2
Input second num: 4
6
excul@exculserver:~/shes$
```

Рисунок 25 - Код программы по заданию 25 и результат работы