

Липецкий государственный технический университет

Кафедра прикладной математики

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

Студент

подпись, дата

Пустовалова И.П.
фамилия, инициалы

Группа

Руководитель

Доцент, к. пед. наук
ученая степень, ученое звание

подпись, дата

Кургасов В.В.
фамилия, инициалы

Липецк 2021 г.

Содержание

Цель работы	3
Задание кафедры	4
Ход работы	7
Выводы	24

Цель работы

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

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.

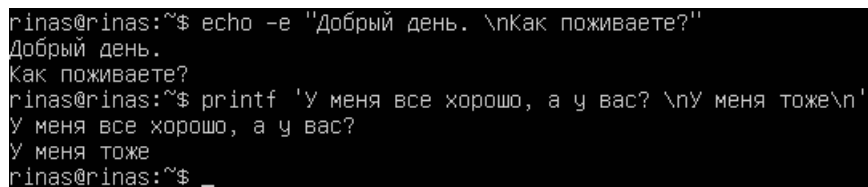
12. Используя позиционные параметры, отобразить имя программы, количество аргументов командной строки, значение каждого аргумента командной строки.
13. Используя позиционный параметр, отобразить содержимое текстового файла, указанного в качестве аргумента командной строки. После паузы экран очищается.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

Ход работы

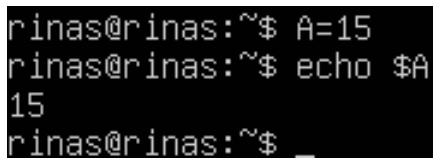
1. Используя команды ECHO, PRINTF вывести информационные сообщения на экран.



```
rinas@rinas:~$ echo -e "Добрый день. \nКак поживаете?"
Добрый день.
Как поживаете?
rinas@rinas:~$ printf 'У меня все хорошо, а у вас? \nУ меня тоже\n'
У меня все хорошо, а у вас?
У меня тоже
rinas@rinas:~$ _
```

Рисунок 1 – Задание 1.

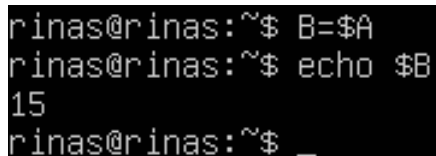
2. Присвоить переменной А целочисленное значение. Просмотреть значение переменной А.



```
rinas@rinas:~$ A=15
rinas@rinas:~$ echo $A
15
rinas@rinas:~$ _
```

Рисунок 2 – Задание 2.

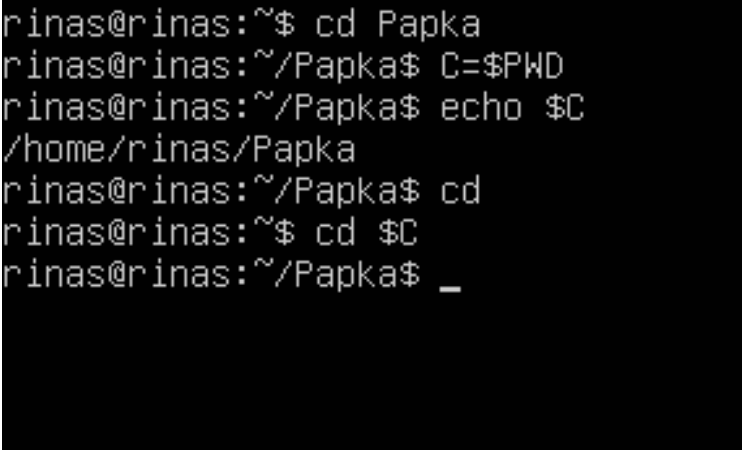
3. Присвоить переменной В значение переменной А. Просмотреть значение переменной В.



```
rinas@rinas:~$ B=$A
rinas@rinas:~$ echo $B
15
rinas@rinas:~$ _
```

Рисунок 3 – Задание 3.

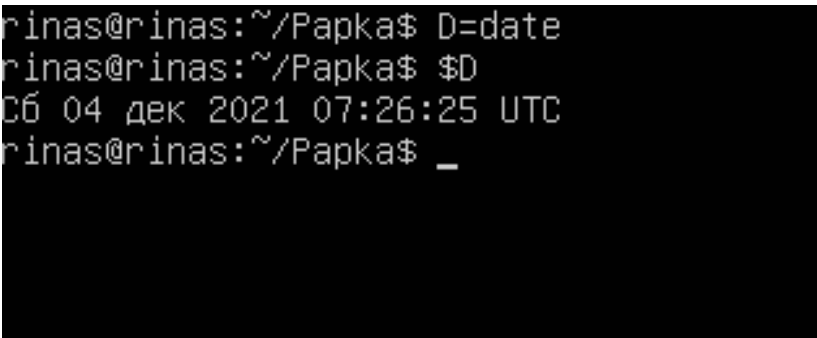
4. Присвоить переменной C значение “путь до своего каталога”. Перейти в этот каталог с использованием переменной.

A terminal window with a black background and white text. The text shows a series of commands and their outputs. The prompt is 'rinas@rinas:~\$'. The first command is 'cd Papka', which changes the directory to '~/. Papka'. The second command is 'C=\$PWD', which assigns the current directory path to variable C. The third command is 'echo \$C', which outputs '/home/rinas/Papka'. The fourth command is 'cd', which changes the directory to '~/. Papka'. The fifth command is '\$ cd \$C', which changes the directory back to '~/. Papka'. The prompt is now 'rinas@rinas:~/Papka\$' followed by a cursor.

```
rinas@rinas:~$ cd Papka
rinas@rinas:~/Papka$ C=$PWD
rinas@rinas:~/Papka$ echo $C
/home/rinas/Papka
rinas@rinas:~/Papka$ cd
rinas@rinas:~$ cd $C
rinas@rinas:~/Papka$ _
```

Рисунок 4 – Задание 4.

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

A terminal window with a black background and white text. The text shows a series of commands and their outputs. The prompt is 'rinas@rinas:~/Papka\$'. The first command is 'D=date', which assigns the command 'date' to variable D. The second command is '\$D', which outputs the date and time. The third command is '\$ _', which outputs the prompt. The prompt is now 'rinas@rinas:~/Papka\$' followed by a cursor.

```
rinas@rinas:~/Papka$ D=date
rinas@rinas:~/Papka$ $D
06 04 дек 2021 07:26:25 UTC
rinas@rinas:~/Papka$ _
```

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.

```
rinas@rinas:~$ E=cat
rinas@rinas:~$ ls
1.txt  echo  Loop  Папка
rinas@rinas:~$ $E Loop
while true; do true;done

rinas@rinas:~$ _
```

Рисунок 6 – Задание 6.

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

```
rinas@rinas:~$ F=sort
rinas@rinas:~$ $F 2.txt
А у тебя?
Все хорошо
Как дела?
Привет
У меня тоже
rinas@rinas:~$ cat 2.txt
Привет
Как дела?
Все хорошо
А у тебя?
У меня тоже
rinas@rinas:~$ _
```

Рисунок 7 – Задание 7.

Написать скрипты, при запуске которых выполняются следующие действия:

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

```
rinas@rinas:~$ ls -l
total 24
-rw-rw-r-- 1 rinas rinas 3966 ноя 27 07:18 1.txt
-rw-rw-r-- 1 rinas rinas  87 дек  4 07:43 2.txt
-rw-rw-r-- 1 rinas rinas   0 дек  4 07:00 echo
-rw-rw-r-- 1 rinas rinas  26 ноя 27 05:53 Loop
drwxrwxr-x 2 rinas rinas 4096 дек  4 07:12 Папка
-rw-rw-r-- 1 rinas rinas  22 дек  4 07:54 script
-rw-rw-r-- 1 rinas rinas  38 дек 11 02:46 skr.sh
rinas@rinas:~$ sudo chmod a+x skr.sh
[sudo] password for rinas:
rinas@rinas:~$ ls -l
total 24
-rw-rw-r-- 1 rinas rinas 3966 ноя 27 07:18 1.txt
-rw-rw-r-- 1 rinas rinas  87 дек  4 07:43 2.txt
-rw-rw-r-- 1 rinas rinas   0 дек  4 07:00 echo
-rw-rw-r-- 1 rinas rinas  26 ноя 27 05:53 Loop
drwxrwxr-x 2 rinas rinas 4096 дек  4 07:12 Папка
-rw-rw-r-- 1 rinas rinas  22 дек  4 07:54 script
-rwxrwxr-x 1 rinas rinas  38 дек 11 02:46 skr.sh
rinas@rinas:~$ ./skr.sh
Мне лет
rinas@rinas:~$ ./skr.sh 20
Мне 20 лет
rinas@rinas:~$ cat skr.sh
#!/bin/bash

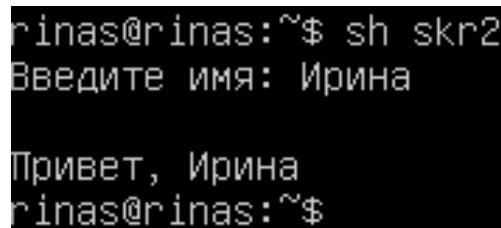
echo "Мне $1 лет"
rinas@rinas:~$ _
```

Рисунок 8 – Задание 8.

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

```
printf 'Введите имя: '
read name
printf '\nПривет, '
echo $name
_
```

Рисунок 9 – Задание 9 скрипт.

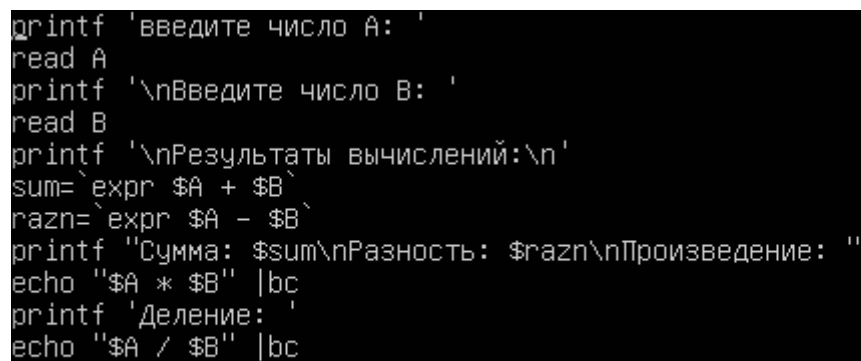


```
rinas@rinas:~$ sh skr2
Введите имя: Ирина

Привет, Ирина
rinas@rinas:~$
```

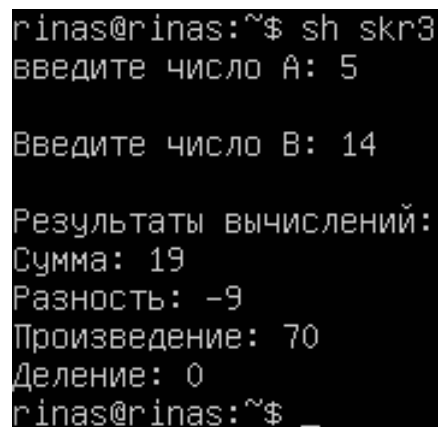
Рисунок 10 – Задание 9 выполнение.

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



```
printf 'введите число A: '
read A
printf '\nВведите число B: '
read B
printf '\nРезультаты вычислений:\n'
sum=`expr $A + $B`
razn=`expr $A - $B`
printf "Сумма: $sum\nРазность: $razn\nПроизведение: "
echo "$A * $B" |bc
printf 'Деление: '
echo "$A / $B" |bc
```

Рисунок 11 – Задание 10 скрипт.



```
rinas@rinas:~$ sh skr3
Введите число A: 5

Введите число B: 14

Результаты вычислений:
Сумма: 19
Разность: -9
Произведение: 70
Деление: 0
rinas@rinas:~$ _
```

Рисунок 12 – Задание 10 выполнение.

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

```
printf 'Введите радиус: '  
read r  
printf '\nВведите высоту: '  
read h  
printf '\nV = '  
echo "$r * $r * $h * 3.1416" |bc  
printf '\n'
```

Рисунок 13 – Задание 11 скрипт.

```
rinas@rinas:~$ sh skr5  
Введите радиус: 2  
  
Введите высоту: 2  
  
V = 25.1328  
  
rinas@rinas:~$ _
```

Рисунок 14 – Задание 11 выполнение.

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

```
#!/bin/bash  
echo "Имя программы: $0"  
echo "Количество аргументов: $#"  
for argument in $@  
do echo "Значение аргумента: $argument"  
done
```

Рисунок 15 – Задание 12 скрипт.

```
rinas@rinas:~$ sh skr6 1 2 x w
Имя программы: skr6
Количество аргументов: 4
Значение аргумента: 1
Значение аргумента: 2
Значение аргумента: x
Значение аргумента: w
rinas@rinas:~$
```

Рисунок 16 – Задание 12 выполнение.

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

```
#!/bin/bash
cat $1
sleep 20
clear
exit
```

Рисунок 17 – Задание 13 скрипт.

```
rinas@rinas:~$ sh skr7 2.txt
Привет
Как дела?
Все хорошо
А у тебя?
У меня тоже
_
```

Рисунок 18 – Задание 13 выполнение.

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

```
#!/bin/bash
for file in ./*.txt
do
cat $file
done
```

Рисунок 19 – Задание 14 скрипт.

```
rinas@rinas:~$ ./skr8
Привет
Как дела?
Все хорошо
А у тебя?
У меня тоже
rinas@rinas:~$ _
```

Рисунок 20 – Задание 14 выполнение.

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

```
#!/bin/bash
printf "Введите число от 1 до 20\n"
read A
if((A > 1 && A < 20))
then
printf 'Введено верное значение\n'
else
printf 'Ошибка\n'
fi
```

Рисунок 21 – Задание 15 скрипт.

```

rinas@rinas:~$ ./skr8
Введите число от 1 до 20
A = g
Ошибка
rinas@rinas:~$ ./skr8
Введите число от 1 до 20
A = -5
Ошибка
rinas@rinas:~$ ./skr8
Введите число от 1 до 20
A = 15
Введено верное значение
rinas@rinas:~$ ./skr8
Введите число от 1 до 20
A = 70
Ошибка
rinas@rinas:~$ _

```

Рисунок 22 – Задание 15 выполнение.

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

```

rinas@rinas:~$ ./skr8
Введите год
A = 2020
Введен високосный год
rinas@rinas:~$ ./skr8
Введите год
A = 2001
Введен не високосный год
rinas@rinas:~$ cat skr8
#!/bin/bash
printf "Введите год\nA = "
read A
if((A % 4 == 0))
then
printf 'Введен високосный год\n'
else
printf 'Введен не високосный год\n'
fi
rinas@rinas:~$

```

Рисунок 23 – Задание 16.

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

```
#!/bin/bash
printf "Введите значение x\nx = "
read x
printf "Введите значение y\ny = "
read y
printf "Введите границы\n"
read lg rg

while((x >=lg && x <=rg || y>= lg && y <= rg))
do
((x++))
((y++))
echo 'x =' $x
echo "y ="$y
echo ""
done
```

Рисунок 24 – Задание 17 скрипт.

```
rinas@rinas:~$ ./skr8
Введите значение x
x = 3
Введите значение y
y = 7
Введите границы
2 9
x = 4
y = 8

x = 5
y = 9

x = 6
y = 10

x = 7
y = 11

x = 8
y = 12

x = 9
y = 13

x = 10
y = 14
rinas@rinas:~$ _
```

Рисунок 25 – Задание 17 выполнение.

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

```
#!/bin/bash
printf "Введите пароль\n"
read x
if((x=="Qq123"))
then
ls -li /etc |more
else
printf "Пароль неверный"
fi
```

Рисунок 26 – Задание 18 скрипт.

```
rinas@rinas:~$ ./skr9
Введите пароль
Qq123
```

Рисунок 27 – Задание 18 выполнение.

```
total 812
786965 -rw-r--r-- 1 root root 3028 авг 24 08:42 adduser.conf
786445 drwxr-xr-x 2 root root 4096 авг 24 08:47 alternatives
786447 drwxr-xr-x 3 root root 4096 авг 24 08:47 apparmor
786446 drwxr-xr-x 7 root root 4096 авг 24 08:47 apparmor.d
786448 drwxr-xr-x 3 root root 4096 ноя 26 23:25 apport
786449 drwxr-xr-x 7 root root 4096 окт 9 15:47 apt
786966 -rw-r----- 1 root daemon 144 ноя 12 2018 at.deny
786967 -rw-r--r-- 1 root root 2319 фев 25 2020 bash.bashrc
786968 -rw-r--r-- 1 root root 45 янв 26 2020 bash_completion
786450 drwxr-xr-x 2 root root 4096 ноя 26 23:25 bash_completion.d
786969 -rw-r--r-- 1 root root 367 апр 14 2020 bindresvport.blacklist
786451 drwxr-xr-x 2 root root 4096 апр 22 2020 binfmt.d
786452 drwxr-xr-x 2 root root 4096 авг 24 08:47 byobu
786453 drwxr-xr-x 3 root root 4096 авг 24 08:42 ca-certificates
786566 -rw-r--r-- 1 root root 6570 окт 9 15:57 ca-certificates.conf
786970 -rw-r--r-- 1 root root 6569 авг 24 08:45 ca-certificates.conf.dpkg-old
786454 drwxr-xr-x 2 root root 4096 авг 24 08:47 calendar
786455 drwxr-xr-x 4 root root 4096 окт 9 15:54 cloud
786456 drwxr-xr-x 2 root root 4096 окт 9 15:59 console-setup
786457 drwxr-xr-x 2 root root 4096 авг 24 08:47 cron.d
786458 drwxr-xr-x 2 root root 4096 ноя 26 23:25 cron.daily
786459 drwxr-xr-x 2 root root 4096 авг 24 08:43 cron.hourly
786460 drwxr-xr-x 2 root root 4096 авг 24 08:43 cron.monthly
786972 -rw-r--r-- 1 root root 1042 фев 13 2020 crontab
786461 drwxr-xr-x 2 root root 4096 авг 24 08:47 cron.weekly
786462 drwxr-xr-x 2 root root 4096 авг 24 08:47 cryptsetup-initramfs
786973 -rw-r--r-- 1 root root 54 авг 24 08:46 crypttab
786463 drwxr-xr-x 4 root root 4096 авг 24 08:42 dbus-1
786464 drwxr-xr-x 3 root root 4096 авг 24 08:46 dconf
786974 -rw-r--r-- 1 root root 2969 авг 3 2019 debconf.conf
786975 -rw-r--r-- 1 root root 13 дек 5 2019 debian_version
786465 drwxr-xr-x 3 root root 4096 ноя 26 23:25 default
786976 -rw-r--r-- 1 root root 604 сен 15 2018 deluser.conf
786466 drwxr-xr-x 2 root root 4096 авг 24 08:43 depmod.d
786467 drwxr-xr-x 4 root root 4096 авг 24 08:45 dhcp
--More--
```

Рисунок 28 – Задание 18 выполнение.

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

```
#!/bin/bash
printf "Введите название файла\n"
read x
if [ -f $x ]
then
printf "Файл найден\n"
cat $x
else
printf "Файл не найден\n"
fi
```

Рисунок 29 – Задание 19 скрипт.

```
rinas@rinas:~$ ./skr9
Введите название файла
loop
Файл найден
while true; do true;done
rinas@rinas:~$
```

Рисунок 30 – Задание 19 выполнение.

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.

```
#!/bin/bash
printf "Введите название файла\n"
read fn
if [ -e $fn ]
then
echo "Файл существует"
if [ -d $fn ]
then
echo "Это каталог"
if [ -r $fn ]
then
echo "Содержимое каталога:"
ls $fn
else echo "Каталог недоступен для чтения"
fi
else
if [ -r $fn ]
then
echo "Содержимое файла:"
cat $fn
else echo "Файл не доступен для чтения"
fi
fi
else
echo "Каталог создан"
mkdir $fn
fi
```

Рисунок 31 – Задание 20 скрипт.

```

rinas@rinas:~$ ls
2.txt  echo  Loop  script  skr2  skr4  skr6  skr8  skr.sh
4.txt  '=lg'  Papka  skr10  skr3  skr5  skr7  skr9  top
rinas@rinas:~$ ./skr10
Введите название файла
Papka
Файл существует
Это каталог
Содержимое каталога:
1.txt 3.txt
rinas@rinas:~$ ./skr10
Введите название файла
Loop
Файл существует
Содержимое файла:
while true; do true;done
rinas@rinas:~$ ./skr10
Введите название файла
7
Каталог создан
rinas@rinas:~$ ls
2.txt  7  '=lg'  Papka  skr10  skr3  skr5  skr7  skr9  top
4.txt  echo  Loop  script  skr2  skr4  skr6  skr8  skr.sh

```

Рисунок 32 – Задание 20 выполнение.

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).

```

#!/bin/bash
printf "Имя первого файла: "
read f1
printf "\nИмя второго файла: "
read f2
if [ -e $f1 ]
then
    if [ -r $f1 ]
    then
        if [ -e $f2 ]
        then
            if [ -w $f2 ]
            then
                cat $f1 > $f2
                echo "Содержимое файла перенаправлено"
            else
                echo "Файл $f2 не записывается"
            fi
        else
            echo "Файл $f2 не существует"
        fi
    else
        echo "Файл $f1 не читается"
    fi
else
    echo "Файл $f1 не существует"
fi

```

Рисунок 33 – Задание 21 скрипт.

```
rinas@rinas:~$ ./skr9
Имя первого файла: echo

Имя второго файла: 2.txt
Содержимое файла перенаправлено
rinas@rinas:~$ _
```

Рисунок 34 – Задание 21 выполнение.

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

```
#!/bin/bash
printf "Введите название программы:"
read exe
if [ -e $exe ]
then
    if [ -x $exe ]
    then
        sh $exe
    else
        echo "Файл $exe не читается"
    fi
else
    echo "Файл $exe не существует"
fi
```

Рисунок 35 – Задание 22 скрипт.

```
rinas@rinas:~$ ./skr9
Введите название программы:skr8
Здравствуй мир
rinas@rinas:~$ ./skr9
Введите название программы:1
Файл 1 не существует
rinas@rinas:~$ ./skr9
Введите название программы:2.txt
Файл 2.txt не читается
rinas@rinas:~$ _
```

Рисунок 36 – Задание 22 выполнение.

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.

```
size=$(wc -c $1 | awk '{print $1}')
if ((size>0))
then
    sort -k1 $1
    cat $1 > new_file.txt
else
    echo "Размер файла равен нулю"
fi
```

Рисунок 37 – Задание 23 скрипт.

```
rinas@rinas:~$ ./skr9 txt.txt
aaaaaaaaaaaaaaaaaaaaa
ddddddddddddddddddd
iiiiiiiiiiiiiiiiiii
qqqqqqqqqqqqqqqqqqq
ttttttttttttttttttt
wwwwwwwwwwwwwwwwwww
zzzzzzzzzzzzzzzzzzz
rinas@rinas:~$ cat txt.txt
qqqqqqqqqqqqqqqqqqq
wwwwwwwwwwwwwwwwwww
ttttttttttttttttttt
iiiiiiiiiiiiiiiiiii
aaaaaaaaaaaaaaaaaaaaa
ddddddddddddddddddd
zzzzzzzzzzzzzzzzzzz
rinas@rinas:~$
```

Рисунок 38 – Задание 23 выполнение.

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.

```
#!/bin/bash
echo "Архивация..."
tar -cvf my.tar *.txt
echo "Просмотр архива"
tar -tf my.tar
echo "Сжатие архива..."
gzip my.tar > my.gz
echo "Сжатие завершено"
```

Рисунок 39 – Задание 24 скрипт.

```
rinas@rinas:~$ ./skr9
Архивация...
2.txt
new_file.txt
txt.txt
Просмотр архива
2.txt
new_file.txt
txt.txt
Сжатие архива...
Сжатие завершено
rinas@rinas:~$
```

Рисунок 40 – Задание 24 выполнение.

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

```
#!/bin/bash
printf "Введите число A: "
read A
printf "Введите число B: "
read B
sum () {
    sum=$(expr $A + $B)
    echo $sum
}
SUM=$(sum)
printf "Сложение: "
echo $SUM
diviz () {
    div=$(expr $A / $B)
    echo $div
}
printf "Деление: "
DIV=$(diviz)
echo $DIV
```

Рисунок 41 – Задание 25 скрипт.

```
rinas@rinas:~$ ./skr9
Введите число A: 15
Введите число B: 5
Сложение: 20
Деление: 3
rinas@rinas:~$ _
```

Рисунок 42 – Задание 25 выполнение.

Вывод

В ходе выполнения данной лабораторной работы мной были получены знания о основных возможностях языка программирования Shell с целью автоматизации процесса администрирования системы за счет написания и использования командных файлов.