

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

Суханов Д.И.
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
1. Ход работы	7
Выводы	22

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.

1. Ход работы

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

```
danila@danilaserver:/$ echo Hello World!  
Hello World!  
danila@danilaserver:/$ printf 'How are you? \n I am fine! \n'  
How are you?  
I am fine!  
danila@danilaserver:/$
```

Рисунок 1 – Задание 1.

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

```
danila@danilaserver:/$ A=10  
danila@danilaserver:/$ echo $A  
10  
danila@danilaserver:/$
```

Рисунок 2 – Задание 2.

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

```
danila@danilaserver:/$ B=$A  
danila@danilaserver:/$ echo $B  
10  
danila@danilaserver:/$
```

Рисунок 3 – Задание 3.

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

```
danila@danilaserver:/bin$ C=$PWD
danila@danilaserver:/bin$ echo $C
/bin
danila@danilaserver:/bin$ cd ..
danila@danilaserver:/$ cd $C
danila@danilaserver:/bin$
```

Рисунок 4 – Задание 4.

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

```
danila@danilaserver:/bin$ D=date
danila@danilaserver:/bin$ $D
Thu 25 Nov 2021 04:16:06 AM UTC
danila@danilaserver:/bin$
```

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.

```
danila@danilaserver:/$ E=cat
danila@danilaserver:/$ ls
bin    dev    lib    libx32  lost+found  opt      outNEW.tar  root  snap    sys  var
boot   etc    lib32  Loop    media      out.gz    pipe        run   srv     tmp
cdrom  home   lib64  Loop2   mnt        outNEW.gz  proc       sbin  swap.img  usr
danila@danilaserver:/$ $E Loop
while true; do true; done
danila@danilaserver:/$ _
```

Рисунок 6 – Задание 6.

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

```
root@danilaserver:/# F=sort
root@danilaserver:/# $F text.txt
After breakfast
Good morning!
I sad
root@danilaserver:/# cat text.txt
Good morning!
After breakfast
I sad
root@danilaserver:/# _
```

Рисунок 7 – Задание 7.

Написать скрипты, при запуске которых выполняются следующие действия:

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

```
root@danilaserver:/# ./script
root@danilaserver:/# ./script 7
7
root@danilaserver:/# cat script
echo $1
root@danilaserver:/# _
```

Рисунок 8 – Задание 8.

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

```
root@danilaserver:/# ./script
Hello, !
root@danilaserver:/# ./script Danila
Hello, Danila!
root@danilaserver:/# cat script
echo Hello, $1!
root@danilaserver:/#
```

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).,

```
root@danilaser:~# ./script 5 5
10
root@danilaser:~# cat script
a=`expr $1 + $2`
echo $a
root@danilaser:~# _
```

Рисунок 10 – Задание 10.

```
root@danilaser:~# ./script 5 5
10
root@danilaser:~# cat script
a=`echo $1+$2 | bc`
echo $a
root@danilaser:~#
```

Рисунок 11 – Задание 10.

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

```
root@danilaserver:/# ./script 5 5
Value: r, h
392.7000
root@danilaserver:/# cat script
echo 'Value: r, h'
v=`echo "3.1416 * $1 * $1 * $2" | bc`
echo $v
root@danilaserver:/#
```

Рисунок 12 – Задание 11.

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

```
root@danilaserver:/# ./script 1 2 3 4
Program name -- ./script
Arguments -- 4
\nArgument value -- 1
\nArgument value -- 2
\nArgument value -- 3
\nArgument value -- 4
root@danilaserver:/# cat script
echo "Program name -- $0"
echo "Arguments -- $#"
```

```
for argument in $@
do
echo "\nArgument value -- $argument"
done
root@danilaserver:/# _
```

Рисунок 13 – Задание 12.

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

```
root@danilaser:~# chmod ugo+x script
root@danilaser:~# ./script 1.txt
```

Рисунок 14 – Задание 13.



Рисунок 15 – Задание 13.

```
root@danilaser:~# cat script
nano $1
root@danilaser:~#
```

Рисунок 16 – Задание 13.

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

```
root@danilaserver:/# ./script
Hi!
Good morning!
After breakfast
I sad
root@danilaserver:/# cat s
sbin/      script  snap/      srv/      swap.img  sys/
root@danilaserver:/# cat script
for file in ./*.txt
do
cat $file
done
root@danilaserver:/# _
```

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.

```
root@danilaserver:/# ./script 7
Checking range 1 to 10
Value is right
root@danilaserver:/# ./script 90
Checking range 1 to 10
Error value:
root@danilaserver:/# cat script
printf 'Checking range 1 to 10\n'
if (($1 < 1 || $1 > 10))
then
printf 'Error value:\n' $1
else
printf 'Value is right\n' $1
fi
root@danilaserver:/#
```

Рисунок 18 – Задание 15.

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

```
root@danilaserver:/# ./script
Enter Year: 2021
2021 - not leap year
root@danilaserver:/# ./script
Enter Year: 2019
2019 - not leap year
root@danilaserver:/# ./script
Enter Year: 2000
2000 - leap year
root@danilaserver:/# cat script
read -p "Enter Year: " Y
if(($Y % 4 == 0))
then
printf '%s - leap year\n' $Y
else
printf '%s - not leap year\n' $Y
fi
root@danilaserver:/# _
```

Рисунок 19 – Задание 16.

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

```
Enter values range - 1 10
Value are in the range, we increment
x = 2
y = 5
Value are in the range, we increment
x = 3
y = 6
Value are in the range, we increment
x = 4
y = 7
Value are in the range, we increment
x = 5
y = 8
Value are in the range, we increment
x = 6
y = 9
Value are in the range, we increment
x = 7
y = 10
root@danilaserver:/# cat script
read -p "Enter integer value of two variabes (x,y) - " x y
read -p "Enter values range - " a b
if (($x >= $a && $y >= $a && a < $b))
then
while(($x < $b && $y < $b))
do
printf 'Value are in the range, we increment\n'
x=$((x+1))
y=$((y+1))
printf 'x = %s\n' $x
printf 'y = %s\n' $y
done
else
printf 'Error: out of range or wrong range'
fi
root@danilaserver:/# _
```

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.

```

root@danilaserver:/# cat script
if (($1 == "12345"))
then
ls -li /etc | less
else
echo Wrong password
fi
root@danilaserver:/# ./script '12345'_

```

Рисунок 21 – Задание 18.

```

396109 -rw-r----- 1 root shadow 1027 Oct 10 16:34 shadow-
393958 -rw-r--r-- 1 root root 146 Aug 24 08:47 shells
393298 drwxr-xr-x 2 root root 4096 Aug 24 08:45 skel
393299 drwxr-xr-x 6 root root 4096 Aug 24 08:47 sos
393300 drwxr-xr-x 3 root root 4096 Oct 10 16:34 ssh
393301 drwxr-xr-x 4 root root 4096 Oct 10 16:25 ssl
396219 -rw-r--r-- 1 root root 38 Oct 15 16:57 subgid
393960 -rw-r--r-- 1 root root 20 Oct 10 16:34 subgid-
396218 -rw-r--r-- 1 root root 38 Oct 15 16:57 subuid
395971 -rw-r--r-- 1 root root 20 Oct 10 16:34 subuid-
393961 -r--r----- 1 root root 755 Feb 3 2020 sudoers
393302 drwxr-xr-x 2 root root 4096 Aug 24 08:45 sudoers.d
393962 -rw-r--r-- 1 root root 2351 Feb 13 2020 sysctl.conf
393303 drwxr-xr-x 2 root root 4096 Aug 24 08:45 sysctl.d
393304 drwxr-xr-x 5 root root 4096 Aug 24 08:41 systemd
393305 drwxr-xr-x 2 root root 4096 Aug 24 08:42 terminfo
418069 drwxr-xr-x 2 root root 4096 Oct 10 16:21 thermald
393963 -rw-r--r-- 1 root root 8 Oct 10 16:24 timezone
393306 drwxr-xr-x 2 root root 4096 Aug 24 08:47 tmpfiles.d
393307 drwxr-xr-x 2 root root 4096 Aug 24 08:45 ubuntu-advantage
393964 -rw-r--r-- 1 root root 1260 Dec 14 2018 ucf.conf
393308 drwxr-xr-x 4 root root 4096 Aug 24 08:45 udev
393309 drwxr-xr-x 2 root root 4096 Aug 24 08:47 udisks2
393310 drwxr-xr-x 3 root root 4096 Aug 24 08:47 ufw
393311 drwxr-xr-x 3 root root 4096 Aug 24 08:47 update-manager
393312 drwxr-xr-x 2 root root 4096 Aug 24 08:47 update-motd.d
393313 drwxr-xr-x 2 root root 4096 Aug 5 14:01 update-notifier
418070 drwxr-xr-x 2 root root 4096 Oct 10 16:21 UPower
393314 drwxr-xr-x 2 root root 4096 Oct 10 16:23 vim
393315 drwxr-xr-x 4 root root 4096 Aug 24 08:47 vmware-tools
393226 lrwxrwxrwx 1 root root 23 Aug 24 08:43 vtrgb -> /etc/alternatives/vtrgb
393965 -rw-r--r-- 1 root root 4942 Jul 24 2019 wgetrc
393229 drwxr-xr-x 4 root root 4096 Aug 24 08:46 X11
393966 -rw-r--r-- 1 root root 642 Sep 24 2019 xattr.conf
393316 drwxr-xr-x 4 root root 4096 Aug 24 08:43 xdg
393967 -rw-r--r-- 1 root root 477 Oct 7 2019 zsh_command_not_found
(END)

```

Рисунок 22 – Задание 18.

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

```
root@danilaserver:/# ./script Loop
File exist
while true; do true; done
root@danilaserver:/# ./script Loopidooop
File not exist
root@danilaserver:/# cat script
if [ -f $1 ]
then
echo "File exist"
cat $1
else
echo "File not exist"
fi
root@danilaserver:/#
```

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.

```
root@danilaserver:/# ls
@      cdrom  lib      Loop      mnt      outNEW.gz  root      snap      text.txt
1.txt  dev      lib32    Loop2     new_create_dir  outNEW.tar  run      srv      tmp
bin    etc      lib64    lost+found  opt      pipe      sbin     swap.img  usr
boot   home     libx32   media     out.gz    proc      script   sys      var
root@danilaserver:/# ./script Loop2
It is file
while true; do true; echo 'Hello'; done
root@danilaserver:/# ./script home/danila/mydir
It is catalog
myfile myfile1
root@danilaserver:/# ./script myDer
It is not catalog and file. Create catalog.
```

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).

```

root@danilaserver:/# ls
@      cdrom  lib      Loop      mnt      out.gz    proc  script  sys      var
1.txt  dev      lib32    Loop2     myDer     outNEW.gz root  snap    text.txt
bin    etc      lib64    lost+found new_create_dir outNEW.tar run   srv      tmp
boot   home    libx32   media     opt       pipe      sbin  swap.img usr
root@danilaserver:/# ./script 1.txt text.txt
Lets rewrite
root@danilaserver:/# cat 1.txt
Hi!
root@danilaserver:/# cat text.txt
Hi!
root@danilaserver:/# cat script
if test -r $1 -a -w $2
then
echo "Lets rewrite"
cat $1 > $2
elif ! -r $1 -a -w $2
then
echo "First file not readable"
elif -r $1 && -a ! -w $2
then
echo "Second file unavailable for recording"
else
echo "Error: no access to file"
fi
root@danilaserver:/# _

```

Рисунок 25 – Задание 21.

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

```
root@danilaser:~# ./script2 ./superProg
Program is not exist
./script2: line 7: ((: == 1 : syntax error: operand expected (error token is "== 1 ")
root@danilaser:~# ./script2 ./script
Program is exist. Start the program7 1/0 - 1
Enter Year: 2000
2000 - leap year
root@danilaser:~# cat script2
if [ -x $1 ]
then
read -p "Program is exist. Start the program7 1/0 - " C
else
echo "Program is not exist"
fi
if (( $C == 1 ))
then
$1
fi
root@danilaser:~#
```

Рисунок 26 – 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.

```
root@danilaserver:/# cat text.txt
YYYYYYYYYYYYYYYY
NNNNNNNNNNNNNNNN
ZZZZZZZZZZZZZZZZ
AAAAAAAAAAAAAAAA
GGGGGGGGGGGGGGGG
0000000000000000
root@danilaserver:/# ./script2 text.txt text2.txt
AAAAAAAAAAAAAAAA
GGGGGGGGGGGGGGGG
NNNNNNNNNNNNNNNN
0000000000000000
YYYYYYYYYYYYYYYY
ZZZZZZZZZZZZZZZZ
root@danilaserver:/# cat script2
size=$(wc -c $1 | awk '{print $1}')
if ((size>0))
then
sort -k1 $1
cat $1 > $2
else
echo "Error: file is zero-file"
fi
root@danilaserver:/# _
```

Рисунок 27 – 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.

```

root@danilaserver:/home/danila# ls
7 mydir mydir1 my.gz my.tar new2.txt new.txt script
root@danilaserver:/home/danila# ./script
Archiving...
new2.txt
new.txt
Loking through...
new2.txt
new.txt
Compress...
Success!
root@danilaserver:/home/danila# cat script
echo "Archiving..."
tar -cvf my.tar *.txt
echo "Loking through..."
tar -tf my.tar
echo "Compress..."
gzip -c my.tar > my.gz
echo "Success!"
root@danilaserver:/home/danila# _

```

Рисунок 28 – Задание 24..

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

```

root@danilaserver:/home/danila# ./script 77 33
110
root@danilaserver:/home/danila# cat script
func() {
res=`echo $1+$2 | bc`
echo $res
}
func $1 $2
root@danilaserver:/home/danila#

```

Рисунок 29 – Задание 25.

Выводы

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.