

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра Прикладной математики

Лабораторная работа № 4

Создание и использование сценариев (скриптов) в Linux

Студент

Шишкина А.Л.

Группа ПМ-21-2

Руководитель

Доцент

Кургасов В.В.

Цели работы:

- изучить основные возможности языка программирования высокого уровня Shell;
- получить навыки написания и использования скриптов.

Ход работы:

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

```
[alina@fedora ~]$ cat > code.sh
echo "Hello world"
printf "!!!"

^C
[alina@fedora ~]$ sh code.sh
Hello world
!!![alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat code.sh
a=0
echo a=$a
[alina@fedora ~]$ ./code.sh
a=0
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
a=0
b=$a
echo a=$a b=$b
[alina@fedora ~]$ ./code.sh
a=0 b=0
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
c=/home/
echo c=$c
cd $c
[alina@fedora ~]$ ./code.sh
c=/home/
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
D=`date`
echo $D
[alina@fedora ~]$ ./code.sh
Пн 23 окт 2023 13:49:24 MSK
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
E=`cat code.sh`
echo $E
[alina@fedora ~]$ ./code.sh
E=`cat code.sh` echo $E
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
F=`sort loop`
echo $F
[alina@fedora ~]$
[alina@fedora ~]$ ./code.sh
done do true; while true;
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
echo "Enter a > "
read a
echo a=$a
[alina@fedora ~]$ ./code.sh
Enter a >
12
a=12
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
echo "Enter your name > "
read a
echo Hello $a.
[alina@fedora ~]$ ./code.sh
Enter your name >
Alina
Hello Alina.
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
echo "Enter a > "
read a
echo "Enter b > "
read b
echo "a+b = `expr $a + $b`"
echo "a-b = `expr $a - $b`"
echo "a*b = `expr $a '*' $b`"
echo "a/b = `expr $a / $b`"
[alina@fedora ~]$ ./code.sh
Enter a >
3
Enter b >
2
a+b = 5
a-b = 1
a*b = 6
a/b = 1
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
echo "Enter r > "
read r
echo "Enter h > "
read h
result=$(echo "3.14*$r*$r*$h" | bc -l)
echo V=$result
[alina@fedora ~]$ ./code.sh
Enter r >
4
Enter h >
5
V=251.20
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
echo $0
echo $1
echo $2
echo $3
[alina@fedora ~]$ ./code.sh
./code.sh

[alina@fedora ~]$ ./code.sh 1 2 3
./code.sh
1
2
3
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code.sh
cat $1
sleep 4
clear
```

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

```
^C[alina@fedora ~]$ cat > code.sh
echo 'Enter number > '
read A
if test $A -lt 5
then
echo 'A<5'
else echo 'A>=5'
fi
[alina@fedora ~]$ ./code.sh
Enter number >
7
A>=5
[alina@fedora ~]$
```

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

```
[alina@fedora ~]$ cat > code1.sh
echo 'Enter year > '
read A
if test `expr $A % 4` -eq 0 -a `expr $A % 100` -ne 0 -o `expr $A % 400` -eq 0
then
echo YES
else echo NO
fi
[alina@fedora ~]$ ./code1.sh
Enter year >
400
YES
[alina@fedora ~]$
```

Результат выдается на экран.

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

```
[alina@fedora ~]$ cat > code1.sh
if test $1 -eq '123'
then
ls -a '/etc'
else
echo Password no valid
fi
[alina@fedora ~]$ ./code1.sh 123
.                  idmapd.conf                .pwd.lock
..                 init.d                      qemu
abrt                inittab                    qemu-ga
adjtime             inputrc                    rc0.d
aliases             iproute2                  rc1.d
alsa                iscsi                      rc2.d
alternatives        issue                      rc3.d
anaconda            issue.d                   rc4.d
anthy-unicode.conf  issue.net                  rc5.d
appstream.conf      java                      rc6.d
asound.conf         kdump                     rc.d
audit               kdump.conf                 reader.conf.d
authselect          kernel                    redhat-release
avahi               krb5.conf                 request-key.conf
bash_completion.d   krb5.conf.d               request-key.d
bashrc              ld.so.cache               resolv.conf
bindresvport.blacklist ld.so.conf                rhashrc
binfmt.d            ld.so.conf.d              rnc
```

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

```

[alina@fedora ~]$ cat > code1.sh
if test -e $1
then
cat $1
else
echo No file
fi
[alina@fedora ~]$ ./code1.sh loop
while true;
do true;
done

```

18. 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.

```

[alina@fedora ~]$ cat > code1.sh
if test -d $1
then
    ls $1
else
    if test -e $1
    then
cat $1
    else
mkdir $1
    fi
fi
[alina@fedora ~]$ ./code1.sh
build-EducationalPractice-Desktop-Debug  oracle_vbox.asc
code1.sh                                  PycharmProjects
code.sh                                   PyLab
EducationalPractice                      saharok
graph.js                                  snap
IdeaProjects                             style.css
index.html                               Task
'<invalid path>'                         'VirtualBox VMs'
'<invalid path>.layout'                  Видео
'Lab_2(JavaScript)'                     Документы
lab2.py                                  Загрузки
Lab2_py                                  Изображения
lab2var2                                'Лабораторная работа СРРПО №1 .pdf'

```

19. 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).

```
[alina@DESKTOP-I9JKIAN ~]$  
[alina@DESKTOP-I9JKIAN ~]$ cat > code1.sh  
if test -r $1  
then  
    if test -w $2  
    then  
        cat $1 >> $2  
    else  
        echo File 2 no write  
    fi  
else  
    echo File 1 no read  
fi  
[alina@DESKTOP-I9JKIAN ~]$ ./code1.sh loop test  
[alina@DESKTOP-I9JKIAN ~]$ cat loop  
while true;  
do true;  
done  
[alina@DESKTOP-I9JKIAN ~]$ cat test  
while true; do true; done  
while true;  
do true;  
done  
[alina@DESKTOP-I9JKIAN ~]$
```

20. 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.

```
[alina@DESKTOP-I9JKIAN ~]$ cat > code1.sh  
if test -s $1  
then  
    cat $1 | sort -k 1 -o $2  
    cat $2  
else  
    echo File size = 0  
fi  
[alina@DESKTOP-I9JKIAN ~]$ ./code1.sh loop test  
done  
do true;  
while true;  
[alina@DESKTOP-I9JKIAN ~]$
```