

Липецкий государственный технический университет

Кафедра прикладной математики

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

Студент

Заев В.В.
фамилия, инициалы

подпись, дата

Группа

Руководитель

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

Кургасов В.В.
фамилия, инициалы

подпись, дата

Липецк 2021 г.

Содержание

Цель работы	3
Задание кафедры	4
Ход работы	7
Выводы	21

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 В значение переменной A. Просмотреть значение переменной В.
4. Присвоить переменной С 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 Е 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

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

```
user@test:~$ echo 'text_1'  
text_1  
user@test:~$ printf 'text_2\n'  
text_2  
user@test:~$ _
```

Рисунок 1 – Задание 1.

- Присвоить переменной А целочисленное значение. Просмотреть значение переменной А.

```
user@test:~$ A=1106  
user@test:~$ echo $A  
1106
```

Рисунок 2 – Задание 2.

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

```
user@test:~$ B=$A  
user@test:~$ echo $B  
1106
```

Рисунок 3 – Задание 3.

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

```
user@test:~$ cd folder
user@test:~/folder$ C=$PWD
user@test:~/folder$ echo $C
/home/user/folder
user@test:~/folder$ cd
user@test:~$ cd $C
user@test:~/folder$
```

Рисунок 4 – Задание 4.

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

```
user@test:~/folder$ D=date
user@test:~/folder$ $D
Thu 20 Jan 2022 08:55:50 PM UTC
```

Рисунок 5 – Задание 5.

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

```
user@test:~$ E=cat
user@test:~$ ls
folder
user@test:~$ $E folder
cat: folder: Is a directory
```

Рисунок 6 – Задание 6.

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

```
user@test:~/folder$ F=sort
user@test:~/folder$ $F text.txt
text_1
text_2
text_3
text_4
user@test:~/folder$ cat text.txt
text_1
text_2
text_4
text_3
user@test:~/folder$ _
```

Рисунок 7 – Задание 7.

Написать скрипты, при запуске которых выполняются следующие действия:

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

```
user@test:~/folder$ bash src1
Insert number:1232
Your number is 1232
user@test:~/folder$ _
```

Рисунок 8 – Задание 8.

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

```
user@test:~/folders$ bash src2
What is your name ? Vlad
Hi, Vlad
user@test:~/folders$ _
```

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).

```
user@test:~/folder$ bash src3
Write two numbers: 12 12
24
0
144
1
user@test:~/folder$
```

Рисунок 10 – Задание 10.

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

```
user@test:~/folder$ bash src4
Radius: 10
Height: 20
628.00
user@test:~/folder$ _
```

Рисунок 11 – Задание 11.

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

```
user@test:~/folder$ bash src5 1 2 3 a b
Name: src5
qt: 5
Argument value: 1
Argument value: 2
Argument value: 3
Argument value: a
Argument value: b
user@test:~/folder$ _
```

Рисунок 12 – Задание 12.

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

```
user@test:~/folder$ bash src6 text.txt
text_1
text_2
text_4
text_3
```

Рисунок 13 – Задание 13.

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

```
user@test:~/folder$ ls
script1  src1  src2  src3  src4  src5  src6  src7  src8  text.txt
user@test:~/folder$ bash src8 text.txt
text_1
text_2
text_4
text_3
user@test:~/folder$
```

Рисунок 14 – Задание 14.

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

```
user@test:~/folder$ bash src9
Number:12
Your number is less 20
user@test:~/folder$ bash src9
Number:21
Your number is over 20
user@test:~/folder$ _
```

Рисунок 15 – Задание 15

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

```
user@test:~/folder$ bash src10
year:2020
2020 is leap year
user@test:~/folder$
```

Рисунок 16 – Задание 16.

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

```
user@test:~/folder$ bash src11
X=12
Y=13
Error: no such file ":Border:"
10 16
x= 13
y= 14

x= 14
y= 15

x= 15
y= 16

x= 16
y= 17

x= 17
y= 18
```

Рисунок 17 – 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.

```
user@test:~/folder$ bash src12
Password
12345_
```

Рисунок 18 – Задание 18.

```
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
796566 -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 19 – Задание 18

19. Проверить, существует ли файл. Выдается соответствующее сообщение.

```
user@test:~/folder$ bash src13
File name:text
File Found
user@test:~/folder$ _
```

Рисунок 20 – 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.

```
user@test:~/folder$ bash src14
File name: folder2
This is a catalog
text.txt
user@test:~/folder$ _
```

Рисунок 21 – 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).

```
user@test:~/folder$ bash src15
Files names: text1 text2
file text2 added
user@test:~/folder$ _
```

Рисунок 22 – Задание 21.

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

```
user@test:~/folder$ bash src16
File name:src1
Insert number:1200
Your number is 1200
user@test:~/folder$
```

Рисунок 23 – 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.

```
user@test:~/folder$ bash src17
File name:text3.txt
aaaaaa
cccccc
ddd
user@test:~/folder$
```

Рисунок 24 – 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.

```
user@test:~/folder$ bash src18
archiving
text.txt
text2.txt
text3.txt
viewing
text.txt
text2.txt
text3.txt
compression
user@test:~/folder$
```

Рисунок 25 – Задание 24.

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

```
user@test:~/folder$ bash src19
numbers:
20 17
20 + 17 = 37
user@test:~/folder$
```

Рисунок 26 – Задание 25.

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.