

Липецкий государственный технический университет

Факультет автоматизации и информатики

Автоматизированные системы управления

ЛАБОРАТОРНАЯ РАБОТА №4

по курсу ДПО Интаро - Linux

«Программирование на Shell. Использование командных файлов»

Студент

Митина М. В.

Группа ПИ-20-1

Руководитель
доц.

Кургасов В.В.

Липецк 2022 г.

Цель работы

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

Задание кафедры

Все скрипты выполнить, содержимое отобразить в отчете по лабораторной работе.

Оглавление

1. Ход работы.....	4
Задание 1	4
Задание 2	4
Задание 3	4
Задание 4	4
Задание 5	5
Задание 6	5
Задание 7	6
Задание 8	6
Задание 9	7
Задание 10	7
Задание 11	8
Задание 12	8
Задание 13	9
Задание 14	9
Задание 15	10
Задание 16	11
Задание 17	11
Задание 18	12
Задание 19	13
Задание 20	13
Задание 21	14
Задание 22	16
Задание 23	16
Задание 24	17
Задание 25	18
Вывод.....	19

1. Ход работы

Задание 1

Формулировка задания: Используя команды ECHO, PRINTF вывести информационные сообщения на экран.

```
mitina_mv@debian:~$ echo 'привет'
привет
```

Рисунок 1. результат выполнения.

Задание 2

Формулировка задания: Присвоить переменной A целочисленное значение. Просмотреть значение переменной A.

```
mitina_mv@debian:~$ A=3
mitina_mv@debian:~$ echo $A
3
```

Рисунок 2. результат выполнения.

Задание 3

Формулировка задания: Присвоить переменной B значение переменной A. Просмотреть значение переменной B.

```
mitina_mv@debian:~$ B=A
mitina_mv@debian:~$ echo $B
A
mitina_mv@debian:~$ B=$A
mitina_mv@debian:~$ echo $B
3
```

Рисунок 3. результат выполнения.

Задание 4

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

```
mitina_mv@debian:~$ C='/home/mitina_mv/'
mitina_mv@debian:~$ cd $C
mitina_mv@debian:~$ pwd
/home/mitina_mv
mitina_mv@debian:~$ cd /
mitina_mv@debian:/$ pwd
/
mitina_mv@debian:/$ cd $C
mitina_mv@debian:~$ pwd
/home/mitina_mv
```

Рисунок 4. результат выполнения.

Задание 5

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

```
mitina_mv@debian:~$ D='DATE'
mitina_mv@debian:~$ $D
-bash: DATE: команда не найдена
mitina_mv@debian:~$ D=date
mitina_mv@debian:~$ date
Чт 01 дек 2022 15:35:16 MSK
mitina_mv@debian:~$ D='date'
mitina_mv@debian:~$ $D
Чт 01 дек 2022 15:35:25 MSK
```

Рисунок 5. результат выполнения.

Задание 6

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

```
mitina_mv@debian:~$ cat loop
while true;
do true;
done;
mitina_mv@debian:~$ E='cat loop'
mitina_mv@debian:~$ echo $E
cat loop
mitina_mv@debian:~$ $E
while true;
do true;
done;
```

Рисунок 6. результат выполнения.

Задание 7

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



```
mitina_mv@debian:~$ sort sorting.txt
1
1
2
2
23
3
3
3
4
4
5
6
6
68
7
86
mitina_mv@debian:~$ F='sort sorting.txt'
mitina_mv@debian:~$ $F
1
1
2
2
23
3
3
3
4
4
5
6
6
68
7
86
```

Рисунок 7. результат выполнения.

Задание 8

Формулировка задания: Программа запрашивает значение переменной, а затем выводит значение этой переменной.



```
GNU nano 5.4 task8
#!/bin/bash
echo 'Enter number'
read a
echo $a
```

Рисунок 8. скрипт задания

```
mitina_mv@debian:~$ sh task8
Enter number
75
75
```

Рисунок 9. результат выполнения.

Задание 9

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

```
GNU nano 5.4 task9
#!/bin/bash
echo 'Введите свое имя'
read name
echo "Hello, " $name "!"
```

Рисунок 10.скрипт задания

```
mitina_mv@debian:~$ sh task9
Введите свое имя
Мария
Hello, Мария !
```

Рисунок 11.результат выполнения.

Задание 10

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

```
GNU nano 5.4 task10
#!/bin/bash
echo 'введите первое число:'
read a
echo 'введите второе число:'
read b
printf "сумма = "
expr $a + $b
```

Рисунок 12.скрипт задания – вариант 1

```
mitina_mv@debian:~$ sh task10
введите первое число:
4
введите второе число:
5
сумма = 9
```

Рисунок 13.результат выполнения.

```

GNU nano 5.4 task10.2
#!/bin/bash
echo "введите первое число:"
read a
echo "введите второе число:"
read b
echo "сумма = "
echo $a + $b | bc

```

Рисунок 14.скрипт задания – вариант 2

Задание 11

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

```

GNU nano 5.4 task11
#!/bin/bash
echo 'введите радиус:'
read R; export R
echo "введите высоту"
read h; export h
pi=3.121528; export pi
printf "V = "
echo "scale=3; $pi*$R*$h*$R" | bc

```

Рисунок 15.скрипт задания

```

mitina_mv@debian:~$ sh task11
введите радиус:
6
введите высоту
8
V = 899.000064

```

Рисунок 16.результат выполнения.

Задание 12

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


```

GNU nano 5.4 task12
#!/bin/bash
echo "Запущена программа: $0"
echo "Количество параметров: $#"
```

Переданы параметры: "

```

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

Рисунок 17.скрипт задания

```

mitina_mv@debian:~$ sh task12 75 3 привет
Запущена программа: task12
Количество параметров: 3
Переданы параметры:
75
3
привет
```

Рисунок 18.результат выполнения.

Задание 13

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

```

GNU nano 5.4 task13
#!/bin/bash
echo "Открываем файл $1..."
cat $1
sleep 5s
clear
```

Рисунок 19.скрипт задания

```

mitina_mv@debian:~$ sh task13 loop
Открываем файл loop...
while true;
do true;
done;
```

Рисунок 20.результат выполнения.

Задание 14

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

```

GNU nano 5.4 task14
#!/bin/bash
files=`ls | grep txt`
for file in $files
do
echo "Просмотр файла $file"
more $file
done

```

Рисунок 21.скрипт задания

```

mitina_mv@debian:~$ sh task14
Просмотр файла f1.txt
Просмотр файла lsR1.txt
f1.txt

Просмотр файла lsR.txt
.:
f1.txt
loop
loop2
lsR.txt
Просмотр файла sorting.txt
1
2
4

```

Рисунок 22.результат выполнения.

Задание 15

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

```

GNU nano 5.4 task15 *
#!/bin/bash
printf "Введите любое число и попробуйте угадать:"
read number
if [ $number -ge 43 -a $number -le 78 ]
then echo "OK! вы угадали, выше число входит в интервал от 43 до 78"
else echo "Эээх! вы не угадали :("
fi

```

Рисунок 23.скрипт задания

```

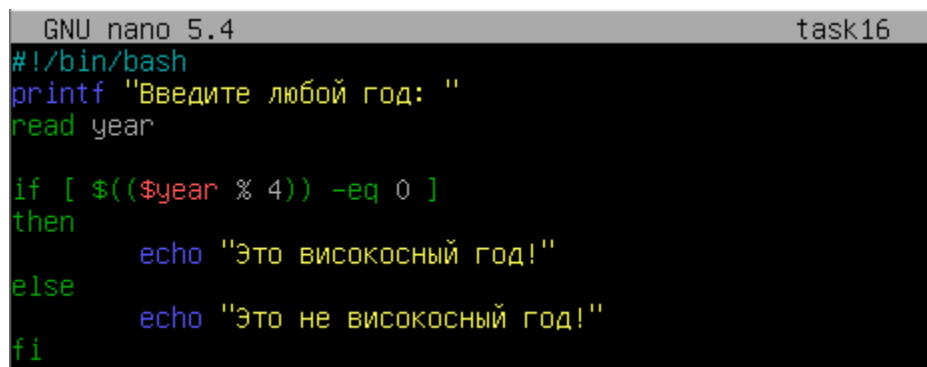
mitina_mv@debian:~$ sh task15
Введите любое число и попробуйте угадать:12
Эээх! вы не угадали :(
mitina_mv@debian:~$ sh task15
Введите любое число и попробуйте угадать:73
OK! вы угадали, выше число входит в интервал от 43 до 78
mitina_mv@debian:~$ sh task15
Введите любое число и попробуйте угадать:43
OK! вы угадали, выше число входит в интервал от 43 до 78

```

Рисунок 24.результат выполнения.

Задание 16

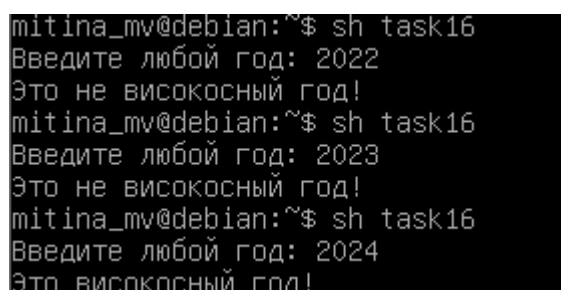
Формулировка задания: Программой запрашивается год, определяется, високосный ли он. Результат выдается на экран.



```
GNU nano 5.4 task16
#!/bin/bash
printf "Введите любой год: "
read year

if [ $((year % 4)) -eq 0 ]
then
    echo "Это високосный год!"
else
    echo "Это не високосный год!"
fi
```

Рисунок 25.скрипт задания



```
mitina_mv@debian:~$ sh task16
Введите любой год: 2022
Это не високосный год!
mitina_mv@debian:~$ sh task16
Введите любой год: 2023
Это не високосный год!
mitina_mv@debian:~$ sh task16
Введите любой год: 2024
Это високосный год!
```

Рисунок 26.результат выполнения.

Задание 17

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

```

GNU nano 5.4 task17
#!/bin/bash
printf "Первое число: "
read num1

printf "Второе число: "
read num2

printf "Начало интервала: "
read beg

printf "Конец интервала: "
read end

while [ $num1 -gt $beg -a $num2 -gt $beg -a $num1 -lt $end -a $num2 -lt $end ]
do
    num1=$((num1 + 1))
    num2=$((num2 + 1))
    echo "Текущие значения переменных: "
    echo "num1 = $num1 | num2 = $num2"
done

```

Рисунок 27.скрипт задания

```

mitina_mv@debian:~$ sh task17
Первое число: 2
Второе число: 3
Начало интервала: 0
Конец интервала: 5
Текущие значения переменных:
num1 = 3 | num2 = 4
Текущие значения переменных:
num1 = 4 | num2 = 5

```

Рисунок 28.результат выполнения.

Задание 18

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

```

GNU nano 5.4 task18
#!/bin/bash
pass='123'

if [ $pass = "$1" ]
then
    ls -al /etc | more
else
    echo "Не угадал, пароль не подходит"
fi

```

Рисунок 29.скрипт задания

```

...skipping 1 line
-rw-r--r-- 1 root root      14 ноя  3 16:51 timezone
drwxr-xr-x 2 root root    4096 авг  7 16:25 tmpfiles.d
-rw-r--r-- 1 root root    1260 июн 16 2020 ucf.conf
drwxr-xr-x 4 root root    4096 ноя  3 16:39 udev
drwxr-xr-x 3 root root    4096 ноя  3 16:59 ufw
drwxr-xr-x 2 root root    4096 ноя  3 16:39 update-motd.d
drwxr-xr-x 2 root root    4096 ноя  3 16:39 vim
-rw-r--r-- 1 root root    4942 ноя 23 2021 wgetrc
drwxr-xr-x 2 root root    4096 ноя  3 17:00 wpa_supplicant
drwxr-xr-x 3 root root    4096 ноя  3 16:40 X11
-rw-r--r-- 1 root root     642 дек 24 2020 xattr.conf
drwxr-xr-x 4 root root    4096 ноя  3 16:59 xdg
mitina_mv@debian:~$ sh task18 pass
Не угадал, пароль не подходит

```

Рисунок 30.результат выполнения.

Задание 19

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

```

GNU nano 5.4 task19
#!/bin/bash
if [ -s "$1" ]
then
    cat "$1"
else
    echo "а файла нет, простите"
fi

```

Рисунок 31.скрипт задания

```

mitina_mv@debian:~$ sh task19 loop
while true;
do true;
done;
mitina_mv@debian:~$ sh task19 lfjs
а файла нет, простите

```

Рисунок 32.результат выполнения.

Задание 20

Формулировка задания: 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.

```

GNU nano 5.4 task20
#!/bin/bash
echo "сейчас будем искать штуку с именем $1"

if [ -d "$1" -a -r "$1" ]
then
    echo "ух ты, мы нашли такую директорию, вот что в ней: "
    ls "$1"
elif [ -s "$1" -a -f "$1" ]
then
    echo "вау, да это же файл, смотрите, что в нем:"
    cat "$1"
else
    echo "ничего подобного нет, вот это вы выдумщик! делаем такую директорию..."
    mkdir "$1"
fi

```

Рисунок 33.скрипт задания

```

mitina_mv@debian:~$ sh task20 loop
сейчас будем искать штуку с именем loop
вау, да это же файл, смотрите, что в нем:
while true;
do true;
done;
mitina_mv@debian:~$ sh task20 dir
сейчас будем искать штуку с именем dir
ничего подобного нет, вот это вы выдумщик! делаем такую директорию...
mitina_mv@debian:~$ ls
ath.tar  kanal  loop  lsR.txt  sorting.txt  task10.2  task13  task16  task19  task9
dir      kanal2  loop2  mashKanal  tar          task11    task14  task17  task20
fl.txt   kanal3  lsR1.txt  outer.gz  task10       task12    task15  task18  task8
mitina_mv@debian:~$ sh task20 dir
сейчас будем искать штуку с именем dir
ух ты, мы нашли такую директорию, вот что в ней:
mitina_mv@debian:~$ _

```

Рисунок 34.результат выполнения.

Задание 21

Формулировка задания: 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)

```
GNU nano 5.4 task21
#!/bin/bash
file1=$1
file2=$2

if [ -s $file1 -a -r $file1 ]
then
    if [ -e $file2 -a -w $file2 ]
    then
        echo "Все хорошо, хозяин! Уже делаю, хозяин!"
        cat $file1 > $file2
    else
        echo "Хозяин, не бей, в файл $file2 нельзя писать!"
    fi
else
    echo "Хозяин, не бей, файл $file1 нельзя читать!"
fi
```

Рисунок 35.скрипт задания

```
mitina_mv@debian:~$ sh task21 loop f1.txt
Все хорошо, хозяин! Уже делаю, хозяин!
mitina_mv@debian:~$ cat f1.txt
while true;
do true;
done;
mitina_mv@debian:~$ sh task21 loop f.txt
Хозяин, не бей, в файл f.txt нельзя писать!
mitina_mv@debian:~$ sh task21 f.txt f1.txt
Хозяин, не бей, файл f.txt нельзя читать!
```

Рисунок 36.результат выполнения.

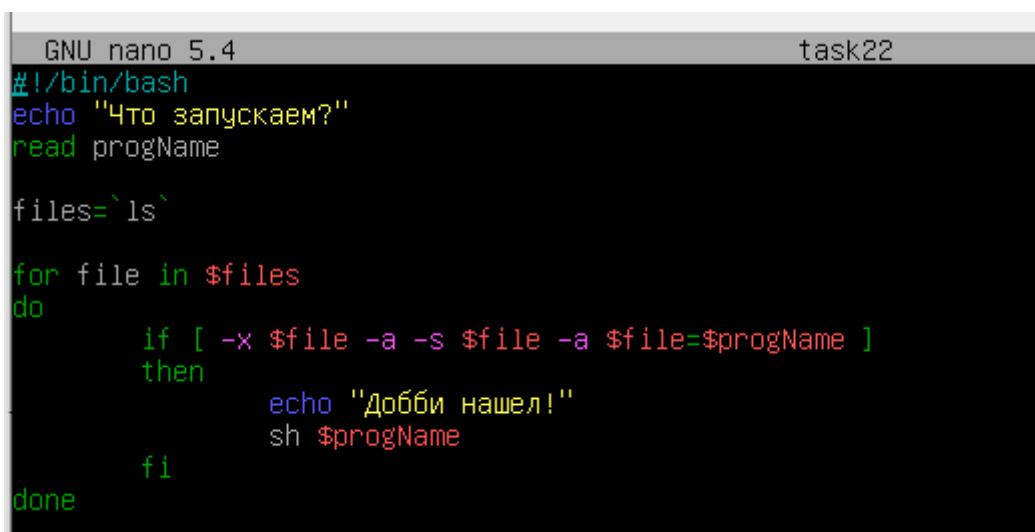
```
GNU nano 5.4 task21
#!/bin/bash
echo "введите названия файлов, хозяин:"
read file1
read file2_

if [ -s $file1 -a -r $file1 ]
then
    if [ -e $file2 -a -w $file2 ]
    then
        echo "Все хорошо, хозяин! Уже делаю, хозяин!"
        cat $file1 > $file2
    else
        echo "Хозяин, не бей, в файл $file2 нельзя писать!"
    fi
else
    echo "Хозяин, не бей, файл $file1 нельзя читать!"
fi
```

Рисунок 37. скрипт задания, вариант 2.

Задание 22

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

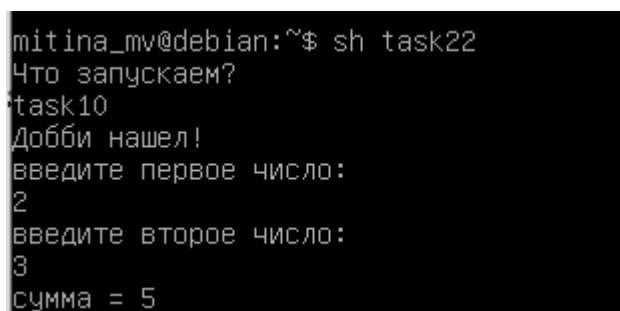


```
GNU nano 5.4 task22
#!/bin/bash
echo "Что запускаем?"
read progName

files=`ls`

for file in $files
do
    if [ -x $file -a -s $file -a $file=$progName ]
    then
        echo "Добби нашел!"
        sh $progName
    fi
done
```

Рисунок 38.скрипт задания



```
mitina_mv@debian:~$ sh task22
Что запускаем?
task10
Добби нашел!
введите первое число:
2
введите второе число:
3
сумма = 5
```

Рисунок 39.результат выполнения.

Задание 23

Формулировка задания: . 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.


```

GNU nano 5.4                                     task23
#!/bin/bash
file=$1

if [ -e $file ]
then
    size=`du $file | cut -c 1`
    if [ $size -gt 0 ]
    then
        cat $file | sort -k 1 | cat > f1.txt | cat f1.txt
    fi
else
    echo "а файла нетю"
fi

```

Рисунок 40.скрипт задания

```

mitina_mv@debian:~$ sh task23 sorting.txt
1
1
2
2
23
3
3
3
3
4
4
5
6
6
68
7
86

```

Рисунок 41.результат выполнения.

Задание 24

Формулировка задания: 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

```

GNU nano 5.4 task24
#!/bin/bash
echo "собираю архивчик..."
tar -cvf my.tar *.txt
echo "архив создан!"

step=5
while [ $step -ge 0 ]
do
    echo "подождите, осталось $step сек."
    step=$((step-1))
done

echo "смотрим содержимое архива: "
tar -tf my.tar

echo "сжимааааааю!"
gzip my.tar

```

Рисунок 42.скрипт задания

```

mitina_mv@debian:~$ sh task24
собираю архивчик...
f1.txt
f2.txt
lsR.txt
lsR1.txt
sorting.txt
архив создан!
подождите, осталось 5 сек.
подождите, осталось 4 сек.
подождите, осталось 3 сек.
подождите, осталось 2 сек.
подождите, осталось 1 сек.
подождите, осталось 0 сек.
смотрим содержимое архива:
f1.txt
f2.txt
lsR.txt
lsR1.txt
sorting.txt
сжимааааааю!
gzip: my.tar.gz already exists; do you wish to overwrite (y or n)? y

```

Рисунок 43.результат выполнения.

Задание 25

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

```
GNU nano 5.4 task25
#!/bin/bash

func () {
    echo "sum $1 + $2 = $((($1 + $2)))"
}

echo "вводи 1 число, живо!"
read a
echo "вводи второе число, быыыстро!"
read b

func $a $b
```

Рисунок 44.скрипт задания

```
mitina_mv@debian:~$ sh task25
вводи 1 число, живо!
2
вводи второе число, быыыстро!
5
sum 2 + 5 = 7
```

Рисунок 45.результат выполнения.

Вывод

В ходе работы я научилась создавать скрипты на shell.