

# **Отчёт по лабораторной работе 13**

**Операционные системы**

Нелиа Нджову

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>14</b>
	<b>Список литературы</b>	<b>15</b>

## Список иллюстраций

3.1	программа . . . . .	8
3.2	запуск . . . . .	9
3.3	работа программы . . . . .	9
3.4	программа на си . . . . .	10
3.5	программа . . . . .	10
3.6	запуск . . . . .	11
3.7	программа . . . . .	11
3.8	запуск . . . . .	12
3.9	удаление созданных файлов . . . . .	12
3.10	программа . . . . .	12
3.11	запуск . . . . .	13
3.12	работа программы . . . . .	13

## Список таблиц

# 1 Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

## 2 Задание

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-iinputfile` — прочитать данные из указанного файла; `-ooutputfile` — вывести данные в указанный файл; `-ршаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-р`.
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `n` (например `1.tmp`, `2.tmp`, `3.tmp`, `4.tmp` и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
4. Написать командный файл, который с помощью команды `tag` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду `find`).

### 3 Выполнение лабораторной работы

Я создала файл lab11-1.sh, используя команды `getopts` `grep`, я написала командный файл, который анализирует командную строку с ключами: `-iinputfile` — прочитать данные из указанного файла; `-ooutputfile` — вывести данные в указанный файл; `-ршаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-р`.(рис.1)

```
lab11-1.sh x
#!/bin/bash
while getopts i:o:p:cn optletter
do
case $optletter in
i) iflag=1; ival=$OPTARG;;
o) oflag=1; oval=$OPTARG;;
p) pflag=1; pval=$OPTARG;;
c) cflag=1;;
n) nflag=1;;
*) echo Illegal option $optletter;;
esac
done
if ! test $cflag
then
cf=-i
fi
if test $nflag
then
nf=-n
fi
grep $cf $nf $pval $ival >> $oval
```

Рис. 3.1: программа

После этого я поменяла права доступа на файл lab11-1.sh и запустила его(рис.2) и (рис.3)



```

nelianjovu@nelianjovu-Aspire-5755G:~$ chmod u+x lab11-1.sh
nelianjovu@nelianjovu-Aspire-5755G:~$ ls -l lab11-1.sh
-rwxrwx-r-- 1 nelianjovu nelianjovu 367 Apr 28 14:06 lab11-1.sh
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-1.sh -p улит -i input.txt -o ou
tput.txt -c -n
grep: input.txt: No such file or directory
nelianjovu@nelianjovu-Aspire-5755G:~$ touch input.txt
nelianjovu@nelianjovu-Aspire-5755G:~$ touch output.txt
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-1.sh -p улит -i input.txt -o ou
tput.txt -c -n
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-1.sh -p улит -i input.txt -o ou
tput.txt -c -n
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-1.sh -p am -i input.txt -o ou
tput.txt -c -n
nelianjovu@nelianjovu-Aspire-5755G:~$

```

Рис. 3.2: запуск

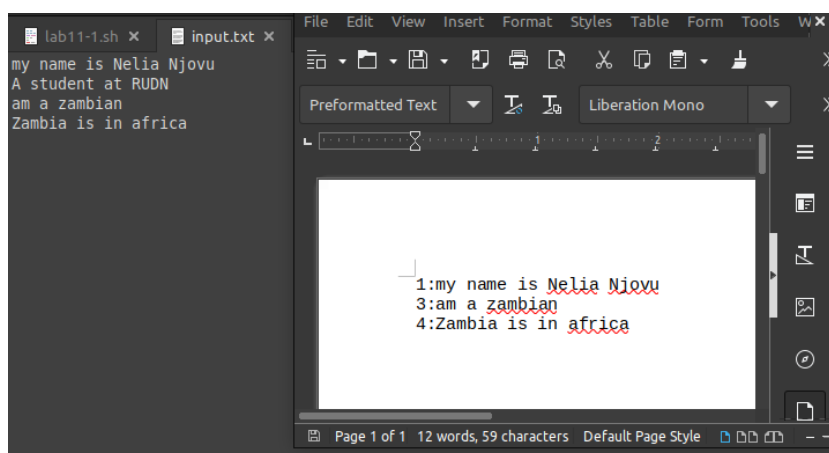
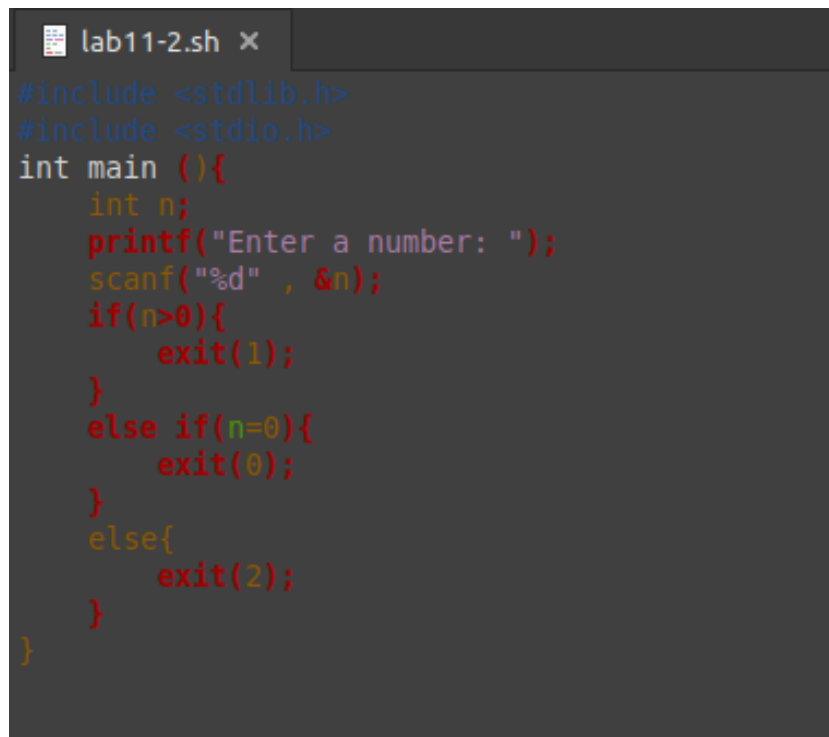


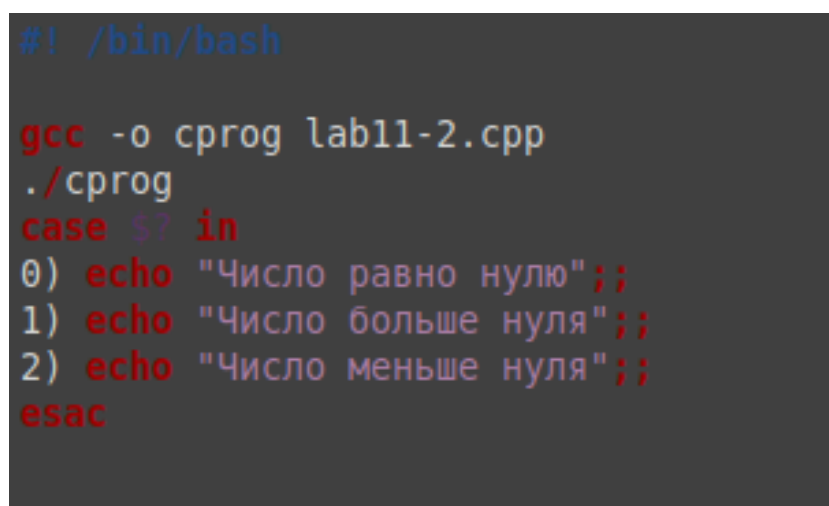
Рис. 3.3: работа программы

Я создала файл lab11-2.sh и lab11-2.cpp, написала на языке Си программу(рис.4), которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции exit(n), передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды \$?, выдать сообщение о том, какое число было введено(рис.5)



```
lab11-2.sh x
#include <stdlib.h>
#include <stdio.h>
int main (){
    int n;
    printf("Enter a number: ");
    scanf("%d" , &n);
    if(n>0){
        exit(1);
    }
    else if(n=0){
        exit(0);
    }
    else{
        exit(2);
    }
}
```

Рис. 3.4: программа на си



```
#!/bin/bash

gcc -o cprog lab11-2.cpp
./cprog
case $? in
0) echo "Число равно нулю";;
1) echo "Число больше нуля";;
2) echo "Число меньше нуля";;
esac
```

Рис. 3.5: программа

После этого я поменяла права доступа на файл lab11-2.sh и запустила его(рис.6)

```

nelianjovu@nelianjovu-Aspire-5755G:~$ chmod u+x lab11-2.sh
nelianjovu@nelianjovu-Aspire-5755G:~$ ls -l lab11-2.sh
-rwxrw-r-- 1 nelianjovu nelianjovu 233 Apr 28 14:30 lab11-2.sh
nelianjovu@nelianjovu-Aspire-5755G:~$ touch lab11-2.cpp
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-2.sh
lab11-2.sh: line 3: syntax error near unexpected token `('
lab11-2.sh: line 3: `int main (){'
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-2.sh
lab11-2.sh: line 3: syntax error near unexpected token `('
lab11-2.sh: line 3: `int main(){'
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-2.sh
lab11-2.sh: line 4: syntax error near unexpected token `('
lab11-2.sh: line 4: `int main () {'
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-2.sh
Enter a number: 3
Число больше нуля
nelianjovu@nelianjovu-Aspire-5755G:~$ █

```

Рис. 3.6: запуск

Я создала файл lab11-3.sh, написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до n (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют)(рис.7)

```

#!/bin/bash
for((i=1; i<=*$*; i++))
do
if test -f "$i".tmp
then rm "$i".tmp
else touch "$i.tmp"
fi
done

```

Рис. 3.7: программа

После этого я поменяла права доступа на файл lab11-3.sh и запустила его(рис.8) и (рис.9)

```
nelianjovu@nelianjovu-Aspire-5755G:~$ chmod u+x lab11-3.sh
nelianjovu@nelianjovu-Aspire-5755G:~$ ls -l lab11-3.sh
-rwxrw-r-- 1 nelianjovu nelianjovu 104 Apr 28 14:46 lab11-3.sh
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-3.sh 5
nelianjovu@nelianjovu-Aspire-5755G:~$ ls
1.tmp          lab11-3.sh
'2024-02-20 21-16-09.mkv' 'Lab3 - Jupyter Notebook.pdf'
2.tmp          'Lab4 - Jupyter Notebook.pdf'
3.tmp          'Lab5 (1) - Jupyter Notebook.pdf'
4.tmp          'Lab6 - Jupyter Notebook.pdf'
5.tmp          LICENSE
```

Рис. 3.8: запуск

```
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-3.sh 5
nelianjovu@nelianjovu-Aspire-5755G:~$ ls
'2024-02-20 21-16-09.mkv' 'Lab4 - Jupyter Notebook.pdf'
backup                  'Lab5 (1) - Jupyter Notebook.pdf'
bin                     'Lab6 - Jupyter Notebook.pdf'
binn                    LICENSE
cprog                   Music
Desktop                 Music.zip
Documents               my_pygame
Downloads               my-winter-project
```

Рис. 3.9: удаление созданных файлов

Я создала файл lab11-4.sh, написала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find)(рис.10)

```
#!/bin/bash
find . -mtime -7 -mtime +0 -type f > FILES.txt
tar -cf archive.tar -T FILES.txt
```

Рис. 3.10: программа

После этого я поменяла права доступа на файл lab11-4.sh и запустила его(рис.11) и (рис.12)

```
nelianjovu@nelianjovu-Aspire-5755G:~$ touch lab11-4.sh
nelianjovu@nelianjovu-Aspire-5755G:~$ chmod u+x lab11-4.sh
nelianjovu@nelianjovu-Aspire-5755G:~$ bash lab11-4.sh /home/nelianjovu
```

Рис. 3.11: запуск

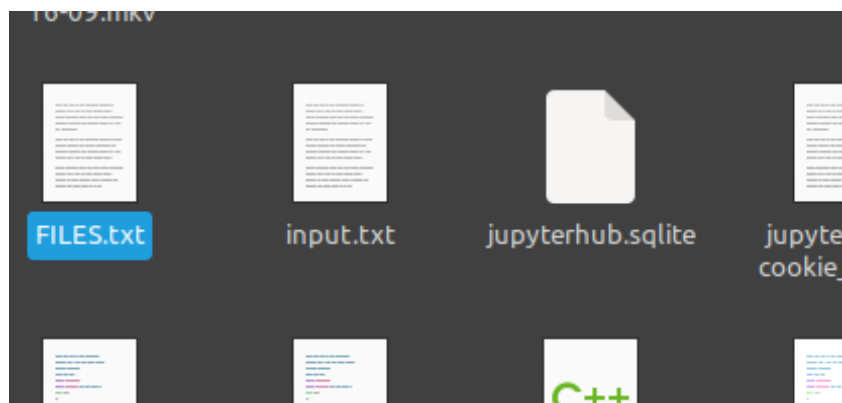


Рис. 3.12: работа программы

## **4 Выводы**

Выполняя эту лабораторную работу, я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов

# Список литературы

лабораторная работа 12