

# Программирование в командном процессоре ОС UNIX.

## Ветвления и циклы

Ыбырай Роза

### Содержание

Цель работы .....	1
Задание .....	1
Выполнение лабораторной работы.....	1
Первое задание .....	1
Второе задание .....	3
Третье задание.....	7
Четвертое задание .....	9
Выводы .....	10

### Цель работы

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

### Задание

Выполнить 4 задания

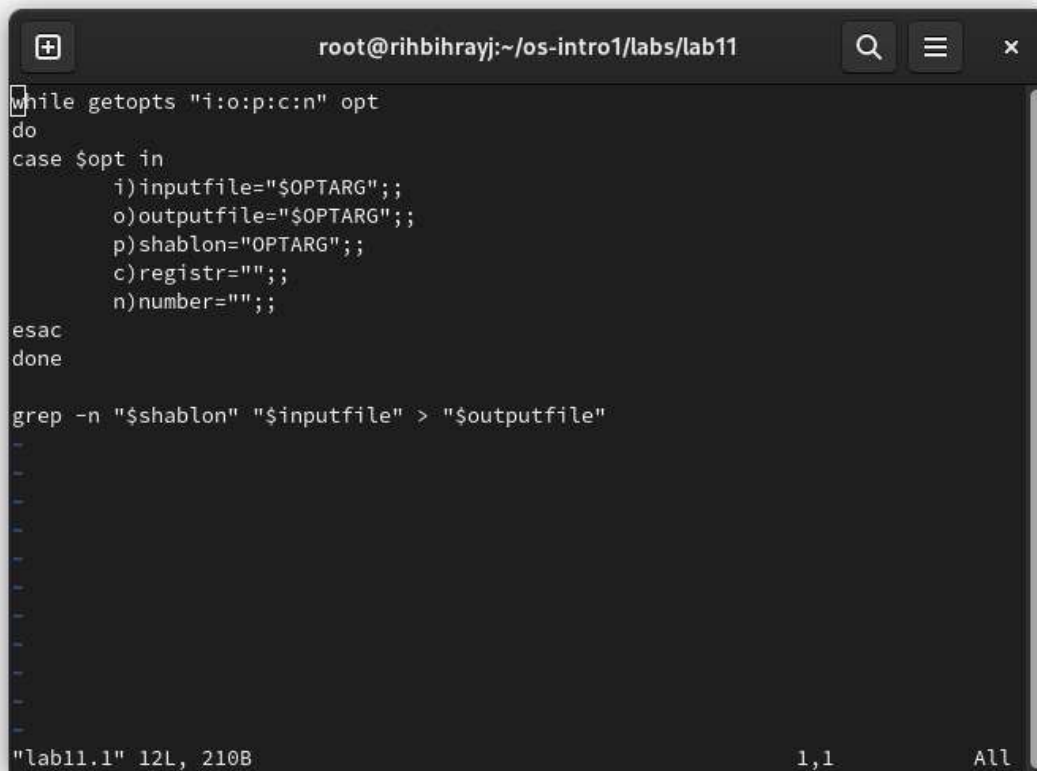
### Выполнение лабораторной работы

#### Первое 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; `-r` — шаблон — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-p`

```
root@rihbihrayj:~/os-intro1/labs/lab11
[rozaybyrai@rihbihrayj ~]$ sudo -i
[sudo] пароль для rozaybyrai:
[root@rihbihrayj ~]# cd os-intro1/labs/lab11
[root@rihbihrayj lab11]# ls
lab11.1  output  presentation  report
[root@rihbihrayj lab11]# vi lab11.1
[root@rihbihrayj lab11]# chmod +x lab11.1
[root@rihbihrayj lab11]# ./lab11.1 -i conf.txt -o output.txt -p h -c -n
grep: conf.txt: Нет такого файла или каталога
[root@rihbihrayj lab11]# mkdir conf.txt
[root@rihbihrayj lab11]# ./lab11.1 -i conf.txt -o output.txt -p h -c -n
grep: conf.txt: Это каталог
[root@rihbihrayj lab11]# ls
conf.txt  lab11.1  output  output.txt  presentation  report
[root@rihbihrayj lab11]#
```

*Рис.1:*



```
root@rihbihrayj:~/os-intro1/labs/lab11
while getopts "i:o:p:c:n" opt
do
case $opt in
    i)inputfile="$OPTARG";;
    o)outputfile="$OPTARG";;
    p)shablon="OPTARG";;
    c)registr="";;
    n)number="";;
esac
done

grep -n "$shablon" "$inputfile" > "$outputfile"
```

"lab11.1" 12L, 210B 1,1 All

Рис.2:

## Второе задание

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.

```
root@rihbihrayj:~/os-intro1/labs/lab11
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка : 1/1
Установка : libstdc++-devel-13.2.1-1.fc38.x86_64 1/2
Установка : gcc-c++-13.2.1-1.fc38.x86_64 2/2
Запуск скриптлета: gcc-c++-13.2.1-1.fc38.x86_64 2/2
Проверка : gcc-c++-13.2.1-1.fc38.x86_64 1/2
Проверка : libstdc++-devel-13.2.1-1.fc38.x86_64 2/2

Установлен:
gcc-c++-13.2.1-1.fc38.x86_64 libstdc++-devel-13.2.1-1.fc38.x86_64

Выполнено!
[root@rihbihrayj lab11]# ./lab11.2 3
Rebuilding compare ...
./lab11.2: строка 10: =g++: команда не найдена
./lab11.2: строка 13: ./compare: Нет такого файла или каталога
[root@rihbihrayj lab11]# vi lab11.2
[root@rihbihrayj lab11]# ./lab11.2 0
Rebuilding compare ...
argument =0
[root@rihbihrayj lab11]# ./lab11.2 1
argument > 0
[root@rihbihrayj lab11]#
```

Рис.3:

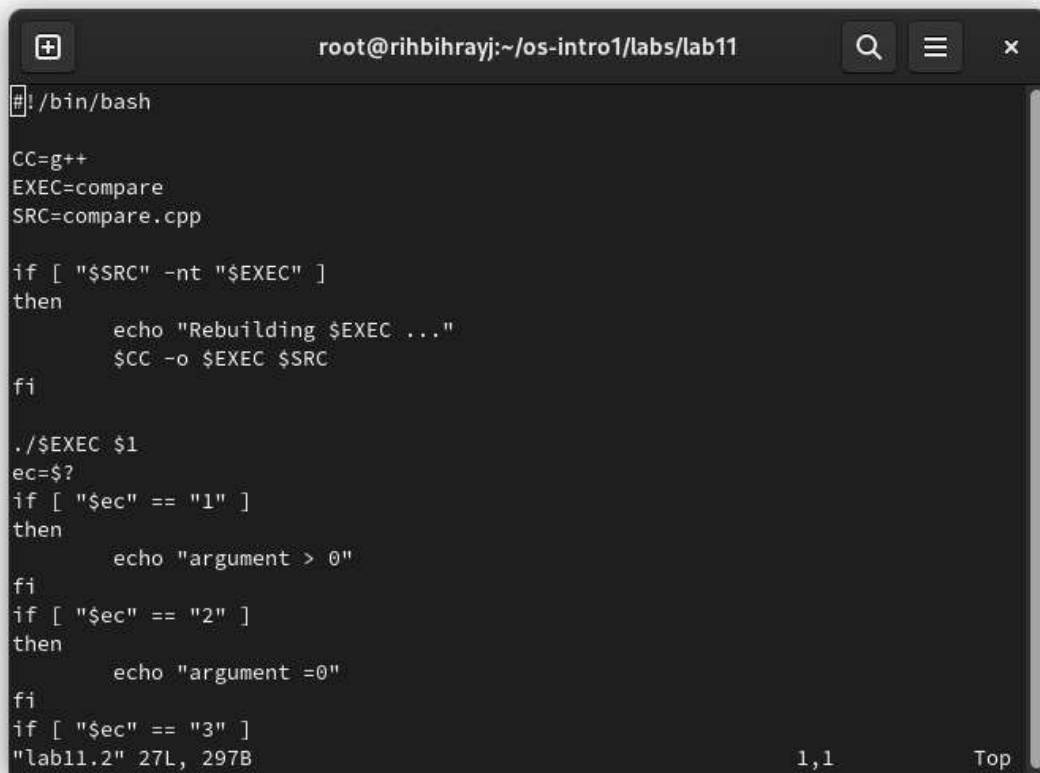
```
root@rihbihrayj:~/os-intro1/labs/lab11

#include <iostream>
using namespace std;

int main(int argument, char *arg[]){
    if (atoi(arg[1]) > 0) {
        exit(1);
    }
    else if (atoi(arg[1]) == 0){
        exit(2);
    }
    else {
        exit(3);
    }
    return 0;
}
```

14,10-17 All

*Puc.4:*



```
root@rihbihrayj:~/os-intro1/labs/lab11
# ./bin/bash

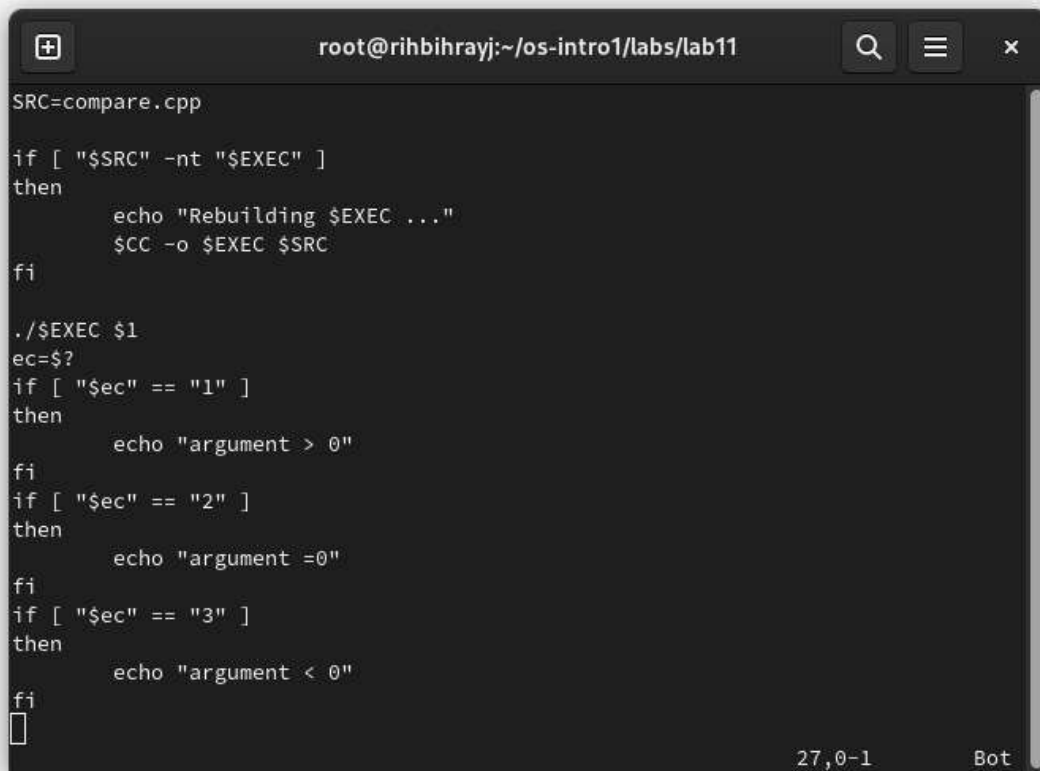
CC=g++
EXEC=compare
SRC=compare.cpp

if [ "$SRC" -nt "$EXEC" ]
then
    echo "Rebuilding $EXEC ..."
    $CC -o $EXEC $SRC
fi

./$EXEC $1
ec=$?
if [ "$ec" == "1" ]
then
    echo "argument > 0"
fi
if [ "$ec" == "2" ]
then
    echo "argument =0"
fi
if [ "$ec" == "3" ]
"lab11.2" 27L, 297B
```

1,1 Top

*Puc.5:*



```
root@rihbihrayj:~/os-intro1/labs/lab11
SRC=compare.cpp

if [ "$SRC" -nt "$EXEC" ]
then
    echo "Rebuilding $EXEC ..."
    $CC -o $EXEC $SRC
fi

./$EXEC $1
ec=$?
if [ "$ec" == "1" ]
then
    echo "argument > 0"
fi
if [ "$ec" == "2" ]
then
    echo "argument =0"
fi
if [ "$ec" == "3" ]
then
    echo "argument < 0"
fi
```

27,0-1 Bot

Рис.6:

### Третье задание

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).

```
root@rihbihrayj:~/os-intro1/labs/lab11
По команде «touch --help» можно получить дополнительную информацию.
touch: неверный ключ - «.»
По команде «touch --help» можно получить дополнительную информацию.
touch: неверный ключ - «.»
По команде «touch --help» можно получить дополнительную информацию.
[root@rihbihrayj lab11]# vi lab11.3
[root@rihbihrayj lab11]# chmod +x lab11.3
[root@rihbihrayj lab11]# ./lab11.3 -c 4
[root@rihbihrayj lab11]# ls
1.tmp  3.tmp  compare  conf.txt  lab11.2  output  presentation
2.tmp  4.tmp  compare.cpp  lab11.1  lab11.3  output.txt  report
[root@rihbihrayj lab11]# vi lab11.3
[root@rihbihrayj lab11]# vi lab11.2
[root@rihbihrayj lab11]# vi lab11.1
[root@rihbihrayj lab11]# vi lab11.3
[root@rihbihrayj lab11]# ./lab11.3 -c 4
[root@rihbihrayj lab11]# ls
1.tmp  3.tmp  compare  conf.txt  lab11.2  output  presentation
2.tmp  4.tmp  compare.cpp  lab11.1  lab11.3  output.txt  report
[root@rihbihrayj lab11]# ./lab11.3 -r
[root@rihbihrayj lab11]# ls
compare  conf.txt  lab11.2  output  presentation
compare.cpp  lab11.1  lab11.3  output.txt  report
[root@rihbihrayj lab11]#
```

Рис.7:



```
root@rihbihrayj:~/os-intro1/labs/lab11
# ./bin/bash

while getopts c:r opt
do
case $opt in
  c)n="$OPTARG"; for i in $(seq 1 $n); do touch "$i.tmp";done;;
  r)for i in $(find -name "*.tmp");do rm $i;done;;
esac
done

"lab11.3" 9L, 175B 1,1 All
```

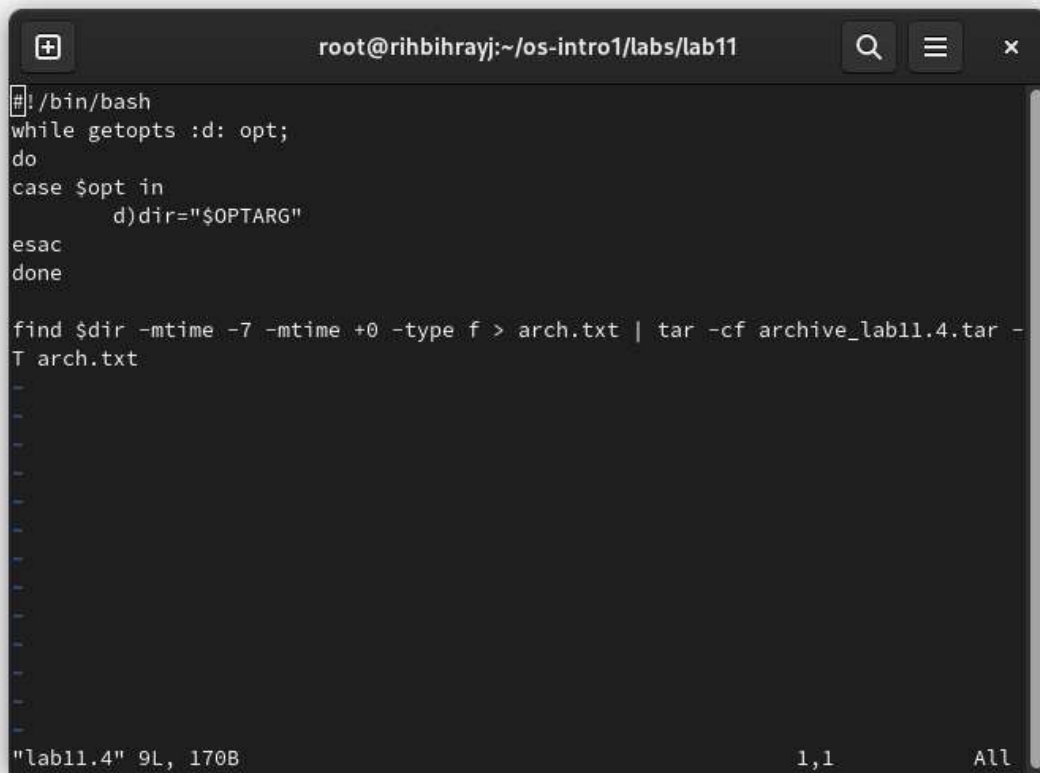
Рис.8:

#### Четвертое задание

4. Написать 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).

```
[root@rihbihrayj lab11]# vi lab11.4
[root@rihbihrayj lab11]# chmod +x lab11.4
[root@rihbihrayj lab11]# ./lab11.4 -d /home
[root@rihbihrayj lab11]# ls
archive_lab11.4.tar  compare.cpp  lab11.2  output  report
arch.txt             conf.txt    lab11.3  output.txt
compare              lab11.1    lab11.4  presentation
[root@rihbihrayj lab11]#
```

Рис.9:

A terminal window with a dark background. The title bar shows 'root@rihbihrayj:~/os-intro1/labs/lab11'. The terminal contains a shell script that uses a while loop and a case statement to process command-line options. It then runs a find command to locate files and pipes the output to a tar command to create an archive. The status bar at the bottom shows '"lab11.4" 9L, 170B', '1,1', and 'All'.

```
root@rihbihrayj:~/os-intro1/labs/lab11
# ./bin/bash
while getopts :d: opt;
do
case $opt in
    d)dir="$OPTARG"
esac
done

find $dir -mtime -7 -mtime +0 -type f > arch.txt | tar -cf archive_lab11.4.tar -
T arch.txt

"lab11.4" 9L, 170B      1,1      All
```

Рис.10:

## Выводы

В ходе данной работы изучили основы программирования в оболочке ОС UNIX и научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

...