

# Отчёт по лабораторной работе №11

## Дисциплина: операционные системы

Лебедева Алёна Алексеевна

### Содержание

Цель работы .....	1
Задание .....	1
Выполнение лабораторной работы.....	2
Контрольные вопросы .....	8
Выводы .....	9

### Цель работы

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

### Задание

1. Используя команды `grep`, написать командный файл, который анализирует командную строку с ключами:
  - `-i` inputfile — прочитать данные из указанного файла; `-o` outfile — 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 `-r`.
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 код 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 `tar` запаковывает архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду `find`).

## Выполнение лабораторной работы

1. Используя команды `getopts` `grep`, написала командный файл, который анализирует командную строку с ключами: `-i` — прочитать данные из указанного файла; `-o` — вывести данные в указанный файл; `-r` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк.

(рис. [-@fig:001])

Открыть + \*lab11.sh

```
1 #!/bin/bash
2 iflag=0; oflag=0; pflag=0; cflag=0; nflag=0;
3 while getopts i:op:C:n optletter
4 do case $optletter in
5 i) iflag=1; ival=$OPTARG;;
6 o) oflag=1; oval=$OPTARG;;
7 p) pflag=1; pval=$OPTARG;;
8 C) cflag=1;;
9 n) nflag=1;;
10 *) echo illegal option $optletter
11 esac
12 done
13 if (($pflag==0))
14 then echo "Шаблон не найден"
15 else
16 if (($iflag==0))
17 then echo "Файл не найден"
18 else
19 if (($oflag==0))
20 then if (($cflag==0))
21 then if (($nflag==0))
22 then grep $pval $ival
23 else grep -n $pval $ival
24 fi
25 else if (($nflag==0))
26 then grep -i $pval $ival
27 else grep -i -n $pval $ival
28 fi
29 fi
30 else if (($cflag==0))
31 then if (($nflag==0))
32 then grep $pval $ival > $oval
33 else grep -n $pval $ival > $oval
34 fi
35 else if (($nflag==0))
36 then grep -i $pval $ival > $oval
37 else grep -i -n $pval $ival > $oval
38 fi
39 fi
40 fi
```

*script1*

(рис. [-@fig:002])



текст

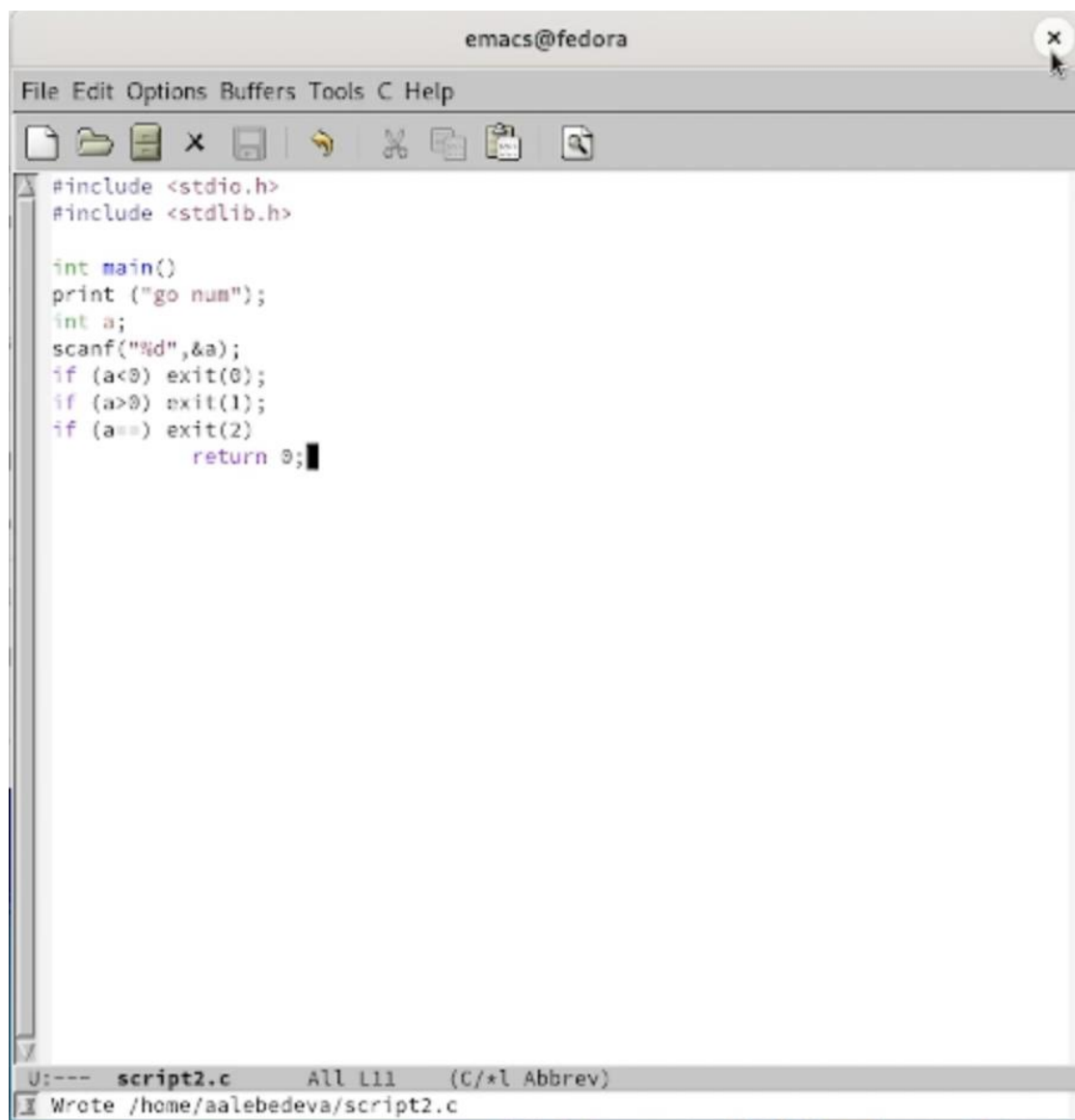
(рис. [-@fig:003])

A screenshot of a terminal window with the title "aalebedeva@fedora:~". The terminal shows a series of commands and their outputs. The commands are:   
1. `bash lab11.sh -ilab11.sh -olab11.txt -pice`  
2. `cat ~/lab11.txt` (output: `ailufgBRFLIUFGoGEFOIEUFIEUF`)  
3. `bash lab11.sh -ilab11.sh -olab11.txt -pice`  
4. `cat ~/lab11.txt` (output: `ailufgBRFLIUFGoGEFOIEUFIEUF`)  
5. `bash lab11.sh -ilab11.sh -olab11-1.txt -pice`  
6. `cat ~/lab11.txt` (output: `ailufgBRFLIUFGoGEFOIEUFIEUF`)  
7. `cat ~/lab11-1.txt` (output: `ailufgBRFLIUFGoGEFOIEUFIEUF`)  
8. `cat lab11.txt` (output: `ailufgBRFLIUFGoGEFOIEUFIEUF`)  
9. `bash lab11.sh -ilab11.sh -olab11-1.txt -pice`  
10. `cat lab11.txt` (output: `ailufgBRFLIUFGoGEFOIEUFIEUF`)  
11. `./lab11.sh -i lab11.txt -o`

терминал

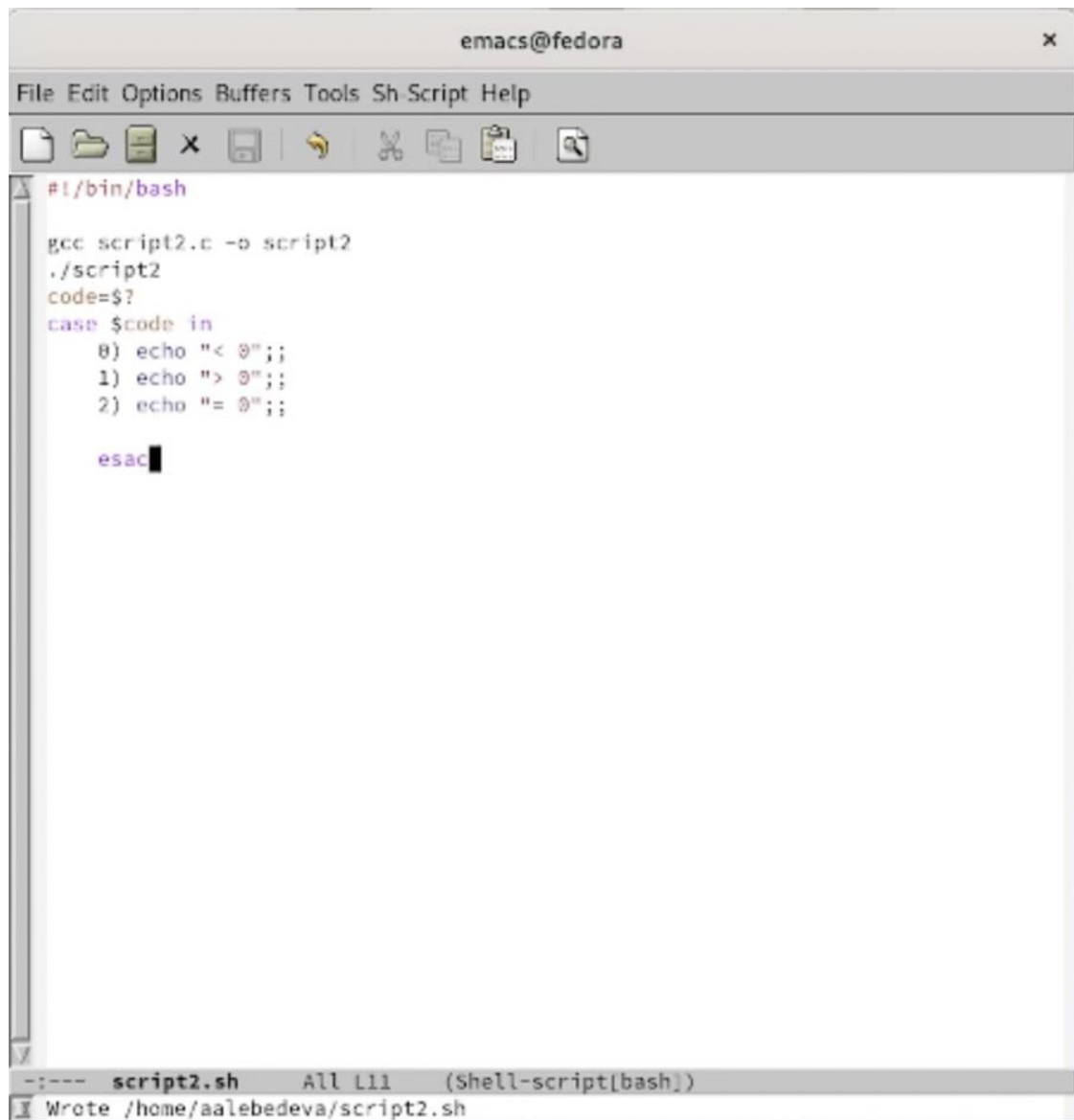
2. Написала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `?`, выдать сообщение о том, какое число было введено.

(рис. [-@fig:004])



*скриншот*

(рис. [-@fig:005])



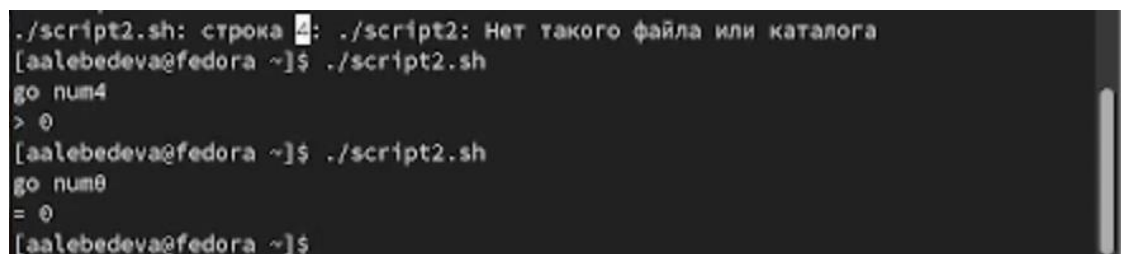
```
#!/bin/bash

gcc script2.c -o script2
./script2
code=$?
case $code in
  0) echo "< 0";;
  1) echo "> 0";;
  2) echo "= 0";;
esac
```

--:--- script2.sh All 111 (Shell-script[bash])  
Wrote /home/aalebedeva/script2.sh

скрипт

(рис. [-@fig:006])



```
./script2.sh: строка 4: ./script2: Нет такого файла или каталога
[aalebedeva@fedora ~]$ ./script2.sh
go num4
> 0
[aalebedeva@fedora ~]$ ./script2.sh
go num0
= 0
[aalebedeva@fedora ~]$
```

терминал

3. 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

(рис. [-@fig:007])



```
1 #!/bin/bash
2
3 opt=$1;
4 form=$2;
5 num=$3;
6 function Files() {
7 for ((i=1; i<=$num; i++)) do
8 file=$(echo $form | tr '0' "$i")
9 if [ $opt == "-r" ]
10 then
11 rm -f $file
12 elif [ $opt == "-c" ]
13 then
14 touch $file
15 fi
16 done
17 }
18 Files
```

скринт

(рис. [-@fig:008])

```
aalebedeva@fedora:~  
[aalebedeva@fedora ~]$ chmod +x script3.sh  
[aalebedeva@fedora ~]$ ./script3.sh -c a1.txt 3  
[aalebedeva@fedora ~]$ ls  
a1.txt      hipmuc49      longfile      script2.c      text.txt  
a2.txt      hipmuc49.pub  may           script2.sh      work  
a3.txt      lab07.sh      monthly       script2.sh-     Видео  
abc1        lab07.sh~     my_os         script3.sh      Документы  
australia   lab11-1.txt   play          script3.sh-     Загрузки  
backup      lab11.sh      presentation  script4.sh      Изображения  
bin         lab11.sh~     q             script4.sh-     Музыка  
conf.txt    lab11.txt     report        script.sh        Общедоступные  
feathers    labr          reports       script.sh-      'Рабочий стол'  
file.txt    logfile       script2       ski.plases  
[aalebedeva@fedora ~]$
```

терминал

## Контрольные вопросы

1. Команда `getopts` является встроенной командой командной оболочки `bash`, предназначенной для разбора параметров сценариев. Она обрабатывает исключительно однобуквенные параметры как с аргументами, так и без них и этого вполне достаточно для передачи сценариям любых входных данных.
2. При генерации имен используют метасимволы: произвольная (возможно пустая) последовательность символов; `?` один произвольный символ; [...] любой из символов, указанных в скобках перечислением и/или с указанием диапазона; `cat f*` выдаст все файлы каталога, начинающиеся с `"f"`; `cat f` выдаст все файлы, содержащие `"f"`; `cat program.?` выдаст файлы данного каталога с однобуквенными расширениями, скажем `"program.c"` и `"program.o"`, но не выдаст `"program.com"`; `cat [a-d]*` выдаст файлы, которые начинаются с `"a"`, `"b"`, `"c"`, `"d"`. Аналогичный эффект дадут и команды `cat [abcd]` и `cat [bdac]`.
3. Операторы `&&` и `||` являются управляющими операторами. Если в командной строке стоит `command1 && command2`, то `command2` выполняется в том, и только в том случае, если статус выхода из команды `command1` равен нулю, что говорит об успешном ее завершении. Аналогично, если командная строка имеет вид `command1 || command2`, то команда `command2` выполняется тогда, и только тогда, когда статус выхода из команды `command1` отличен от нуля.



4. Оператор break завершает выполнение ближайшего включающего цикла или условного оператора, в котором он отображается.
5. Команда true всегда возвращает ноль в качестве выходного статуса для индикации успеха. Команда false всегда возвращает не-ноль в качестве выходного статуса для индикации неудачи. Во всех управляющих конструкциях в качестве логического значения используется код возврата из программы, указанной в качестве условия. Код возврата 0 – истина, любое другое значение – ложь. Программа true – всегда завершается с кодом 0, false – всегда завершается с кодом 1.
6. Введенная строка означает условие существования файла mans/i.\$s
7. Цикл While выполняется до тех пор, пока указанное в нем условие истинно. Когда указанное условие становится ложным - цикл завершается. Цикл Until выполняется до тех пор, пока указанное в нем условие ложно.

## Выводы

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