

# **Лабораторная работа No 11**

**Программирование в командном процессоре ОС UNIX. Ветвления и  
циклы**

Акопян Сатеник Манвеловна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>11</b>
	<b>Список литературы</b>	<b>12</b>

## Список иллюстраций

2.1	рисунок 2 . . . . .	6
2.2	рисунок 3 . . . . .	7
2.3	рисунок 4 . . . . .	7
2.4	рисунок 5 . . . . .	7
2.5	рисунок 6 . . . . .	8
2.6	рисунок 7 . . . . .	8
2.7	рисунок 8 . . . . .	8
2.8	рисунок 9 . . . . .	9
2.9	рисунок 10 . . . . .	9
2.10	рисунок 11 . . . . .	9
2.11	рисунок 12 . . . . .	10

## 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 Выполнение лабораторной работы

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; `-rшаблон` — 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`.

рисунок 1



```
1 #! /bin/bash
2 while getopts i:o:p:cn optletter
3 do case $optletter in
4   i) iflag=1; ival=$OPTARG;;
5   o) oflag=1; oval=$OPTARG;;
6   p) pflag=1; pval=$OPTARG;;
7   C) cflag=1;;
8   N) nflag=1;;
9   *) echo Illegal option $optletter
10  esac
11 done
12 if ! test $cflag
13 then
14   cf=-i
15 fi
16
17 if test $nflag
18 then
19   nf=-n
20 fi
21
22 grep $cf $nf $pval $ival >> $oval
```

Рис. 2.1: рисунок 2

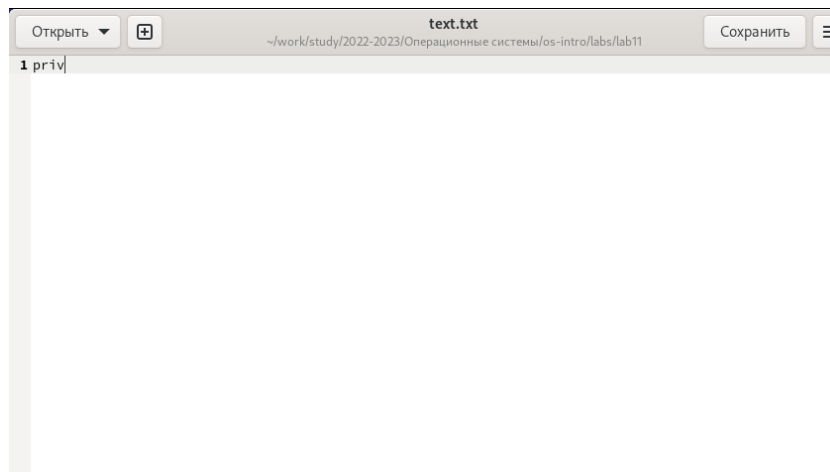


Рис. 2.2: рисунок 3

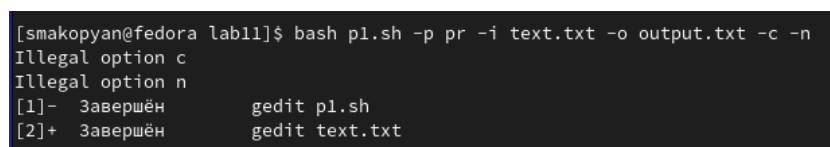


Рис. 2.3: рисунок 4

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. Команд- ный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

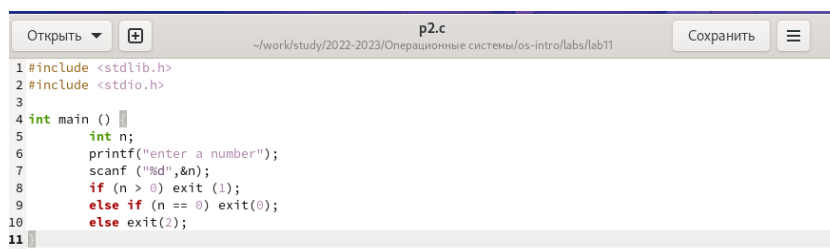
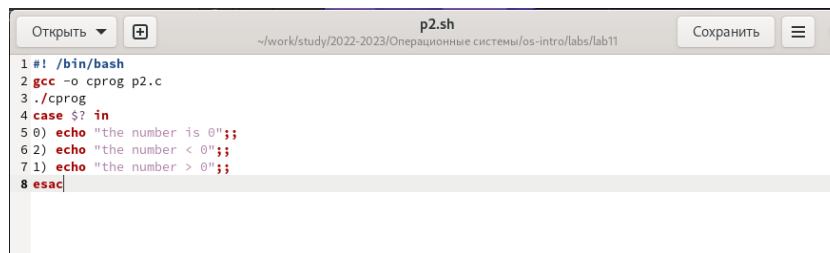
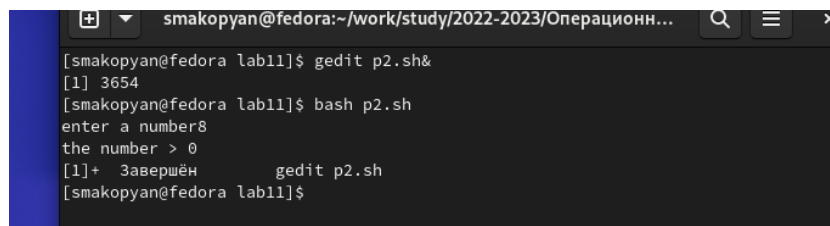


Рис. 2.4: рисунок 5



```
1 #! /bin/bash
2 gcc -o cprog p2.c
3 ./cprog
4 case $? in
5 0) echo "the number is 0";;
6 2) echo "the number < 0";;
7 1) echo "the number > 0";;
8 esac
```

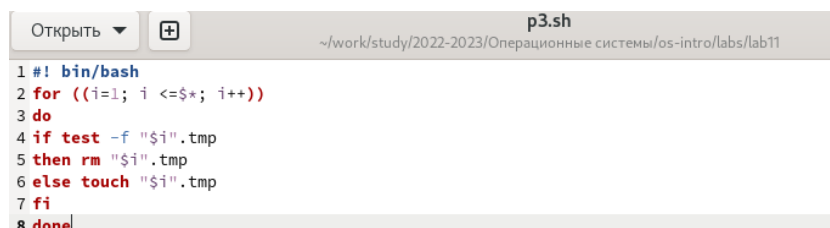
Рис. 2.5: рисунок 6



```
[smakopyan@fedora lab11]$ gedit p2.sh&
[1] 3654
[smakopyan@fedora lab11]$ bash p2.sh
enter a number8
the number > 0
[1]+  Завершён      gedit p2.sh
[smakopyan@fedora lab11]$
```

Рис. 2.6: рисунок 7

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до  $\infty$  (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).



```
1 #! bin/bash
2 for ((i=1; i <=$*; i++))
3 do
4 if test -f "$i".tmp
5 then rm "$i".tmp
6 else touch "$i".tmp
7 fi
8 done
```

Рис. 2.7: рисунок 8



```
[smakopyan@fedora lab11]$ bash p3.sh 3
[smakopyan@fedora lab11]$
```

Рис. 2.8: рисунок 9

```
[smakopyan@fedora lab11]$ bash p3.sh 3
[smakopyan@fedora lab11]$ ls
1.tmp  3.tmp  output.txt  p2.c    p2.sh  presentation  text.txt
2.tmp  cprog  p1.sh      p2.cpp  p3.sh  report
[smakopyan@fedora lab11]$
```

Рис. 2.9: рисунок 10

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`).

```
Открыть  + p4.sh
~/work/study/2022-2023/Операционные системы/os-intro/labs/lab11
1 #! bin/bash
2 find * -mtime -7 -mtime +6 -type f > FILES.txt
3 tar -cv archive.tar -T FILES.txt
```

Рис. 2.10: рисунок 11

/*..	-BBERX-	фев 15 21:07	/afs	0	авг 9 2022
/presentation	56	фев 15 21:07	~bin	7	авг 9 2022
/report	62	фев 15 21:07	/boot	4096	фев 13 14:33
1.tmp	0	апр 22 21:50	/dev	3920	апр 22 21:20
2.tmp	0	апр 22 21:50	/etc	4692	апр 6 15:08
3.tmp	0	апр 22 21:50	/home	18	авг 9 2022
FILES.txt	0	апр 22 21:56	~lib	7	авг 9 2022
*cprog	25040	апр 22 21:46	~lib64	9	авг 9 2022
output.txt	5	апр 22 21:34	/lost+found	0	мая 5 2022
pl.sh	327	апр 22 21:31	/media	0	авг 9 2022

Рис. 2.11: рисунок 12

## **3 Выводы**

В результате данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## **Список литературы**