

Лабораторная работа №11

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

Извекова Мария Петровна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
	Список литературы	13

Список иллюстраций

3.1	код 1	8
3.2	команда <code>bash</code>	8
3.3	результат	9
3.4	код в <code>си</code>	10
3.5	код 2	11
3.6	результат	11
3.7	код 3	11

Список таблиц

1 Цель работы

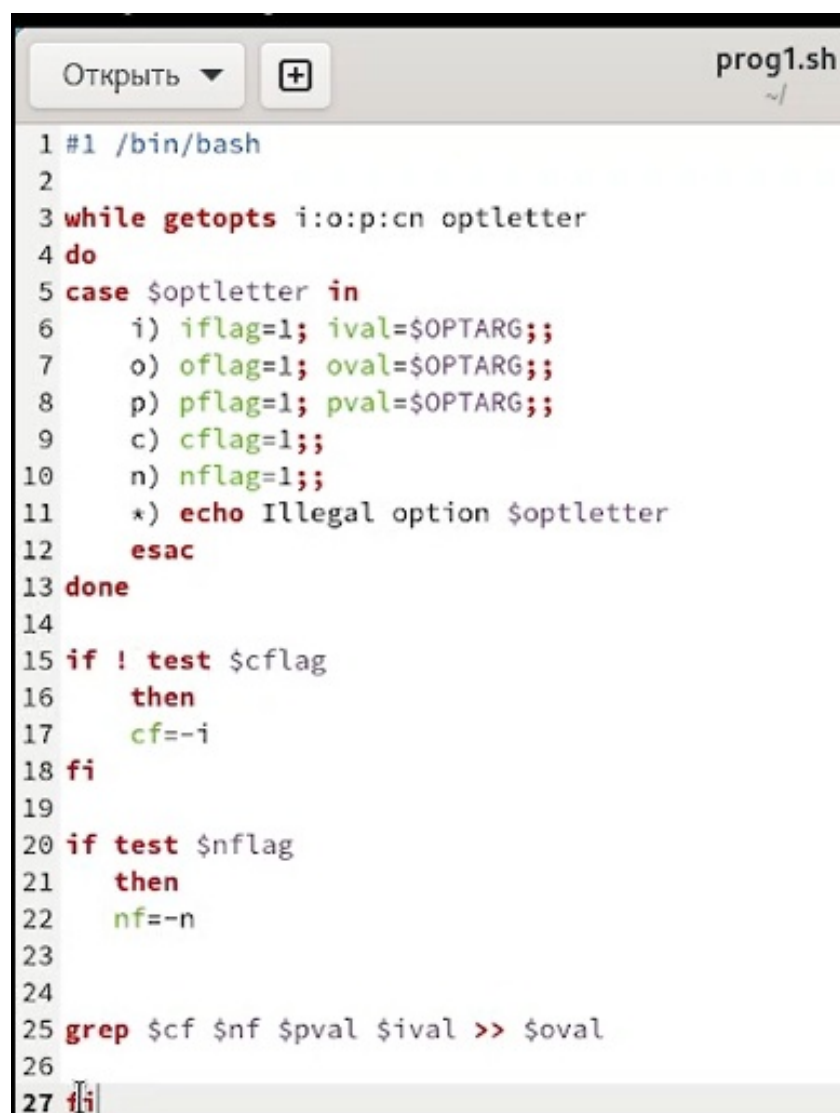
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

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; `-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`.
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 ∞ (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

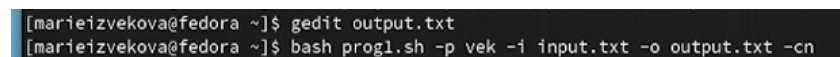
1. Создаются файлы расширения sh, в которые мы будем записывать код, выводящий на экран требуемую информацию из заданий.
2. В первом коде мы пишем команды, которые выводят нам результаты по запрашиваемой части и также строку, где эта часть находится



```
1 #! /bin/bash
2
3 while getopts i:o:p:cn optletter
4 do
5     case $optletter in
6         i) iflag=1; ival=$OPTARG;;
7         o) oflag=1; oval=$OPTARG;;
8         p) pflag=1; pval=$OPTARG;;
9         c) cflag=1;;
10        n) nflag=1;;
11        *) echo Illegal option $optletter
12        esac
13    done
14
15    if ! test $cflag
16    then
17        cf=-i
18    fi
19
20    if test $nflag
21    then
22        nf=-n
23    fi
24
25    grep $cf $nf $pval $ival >> $oval
26
27 fi
```

Рис. 3.1: код 1

В командной строке даем команду для создания результата



```
[marieizvekova@fedora ~]$ gedit output.txt
[marieizvekova@fedora ~]$ bash prog1.sh -p vek -i input.txt -o output.txt -cn
```

Рис. 3.2: команда bash

Результат записывается в отдельный файл

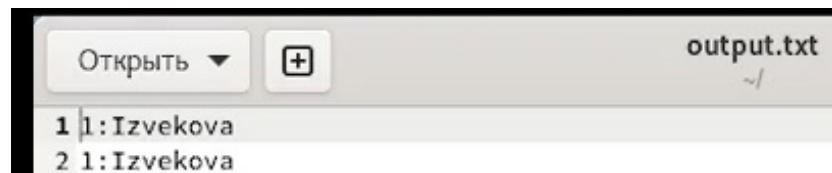


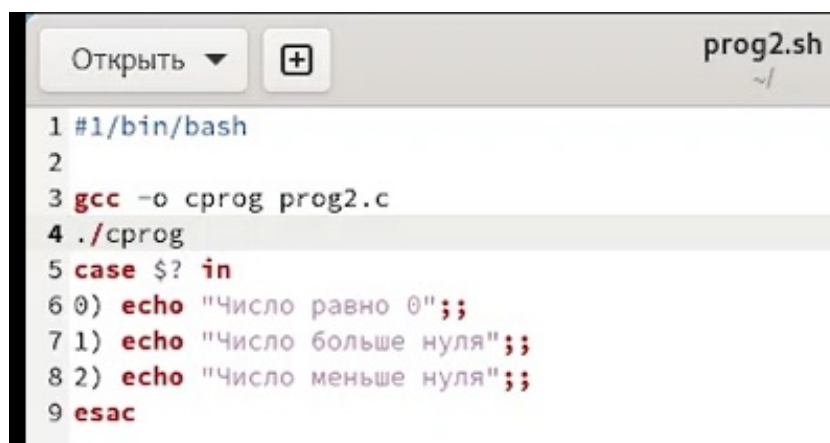
Рис. 3.3: результат

3. Отдельно создаем файл с расширением c, в котором пишем код, для вывода результата: число - больше, меньше или равно 0

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main()
5 {
6
7     int n;
8     printf("number");
9     scanf("%d", &n);
10    if (n>0) {
11        exit(1);
12    }
13    else if (n=0) {
14        exit(0);
15    }
16    else {
17        exit(2);
18    }
19
20 }
```

Рис. 3.4: код в си

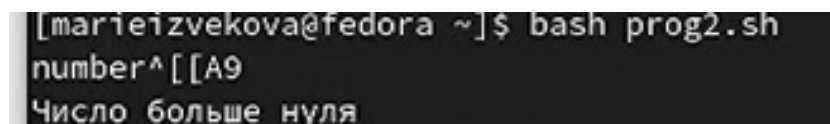
в отдельном файле пишем код, для вывода результата на экран



```
Открыть ▾ + prog2.sh ~/
1 #!/bin/bash
2
3 gcc -o cprog prog2.c
4 ./cprog
5 case $? in
6 0) echo "число равно 0";;
7 1) echo "число больше нуля";;
8 2) echo "число меньше нуля";;
9 esac
```

Рис. 3.5: код 2

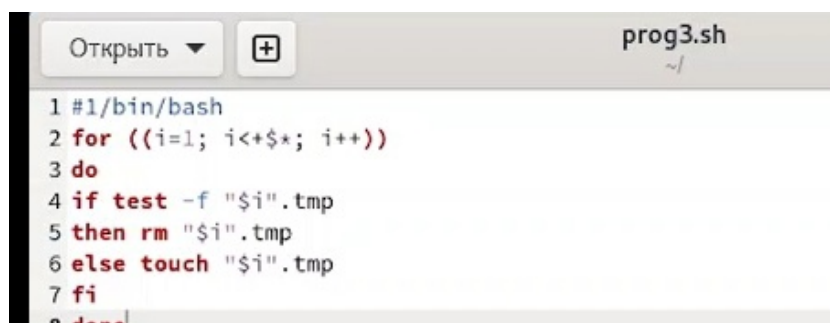
Результат:



```
[marieizvekova@fedora ~]$ bash prog2.sh
number^[A9
Число больше нуля
```

Рис. 3.6: результат

4. Пишем код для создания файлов под определенными номерами



```
Открыть ▾ + prog3.sh ~/
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. 3.7: код 3

Результат:

```

[marieizvekova@fedora ~]$ geotit prog3.sh
[marieizvekova@fedora ~]$ bash prog3.sh 3
[marieizvekova@fedora ~]$ ls
1.tmp  code1.sh  input.txt  output.txt  prog3.sh  Видео  Общедоступные
2.tmp  code2.sh  '#lab07.sh#'  prog1.sh  ski.places  Документы  'Рабочий стол'
abc1   code3.sh  lab07.sh  prog2.c  texput.log  Загрузки  Шаблоны
backup code4.sh  lab07.sh~  prog2.cpp  text.txt  Изображения
bin    cprog    monthly    prog2.sh  work      Музыка

```

Выводы

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

Список литературы