

Лабораторная работа №11. Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Дисциплина: Операционные системы

НВЕ МАНГЕ ХОСЕ. Х.М

22 апреля 2023

Группа НКАбд-03-22

Российский университет дружбы народов, Москва, Россия

Информация

- НВЕ МАНГЕ ХОСЕ. Х.М
- Студент 1-го курса, группа НКАбд-03-22
- математика и Компьютерные науки
- Российский университет дружбы народов
- Ссылка на репозиторий НВЕ МАНГЕ mmnve@dk2n

Вводная часть

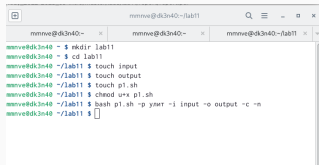
- Работа с программами позволит создать готовые скрипты для различных задач.

- Командный процессор ОС UNIX

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

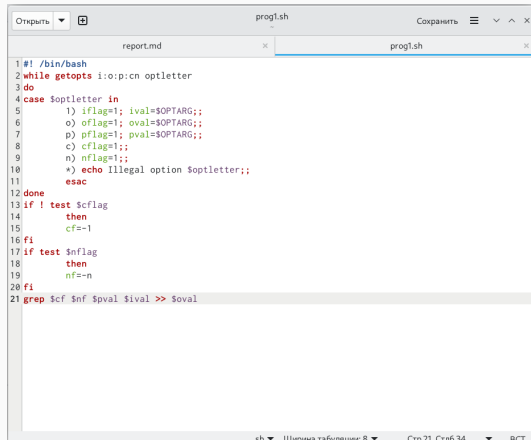
Выполнение заданий.

Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-i`inputfile — прочитать данные из указанного файла; `-o`outputfile — 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`.



```
mmnve@dk3n40:~$ mkdir lab11
mmnve@dk3n40:~$ cd lab11
mmnve@dk3n40 ~/lab11 $ touch input
mmnve@dk3n40 ~/lab11 $ touch output
mmnve@dk3n40 ~/lab11 $ touch p1.sh
mmnve@dk3n40 ~/lab11 $ chmod u+x p1.sh
mmnve@dk3n40 ~/lab11 $ bash p1.sh -p yzwr -i input -o output -c -n
mmnve@dk3n40 ~/lab11 $
```

Рис. 1: Текст программы 1



```
1 #!/bin/bash
2 while getopts i:o:p:cn optletter
3 do
4 case $optletter in
5     1) iflag=1; ival=$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 ! test $cflag
14 then
15     cf=-1
16 fi
17 if test $nflag
18 then
19     nf=-n
20 fi
21 grep $cf $nf $pval $ival >> $oval
```

Рис. 2: Файл, в котором выполнялась программа

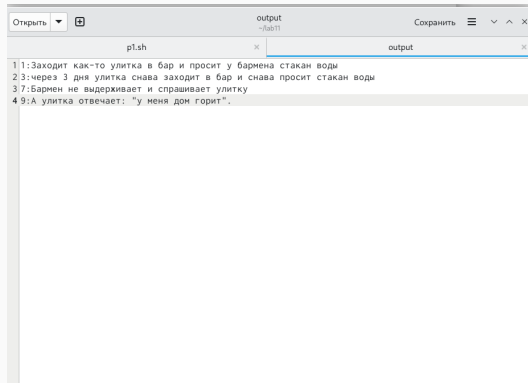
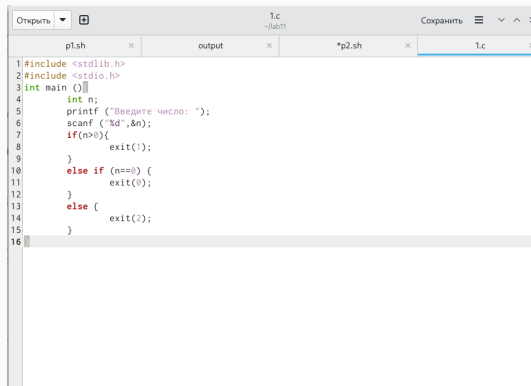
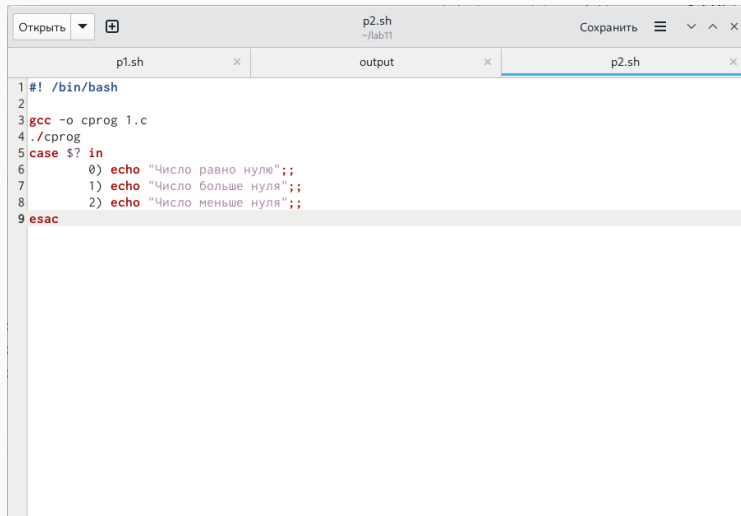


Рис. 3: Результат

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



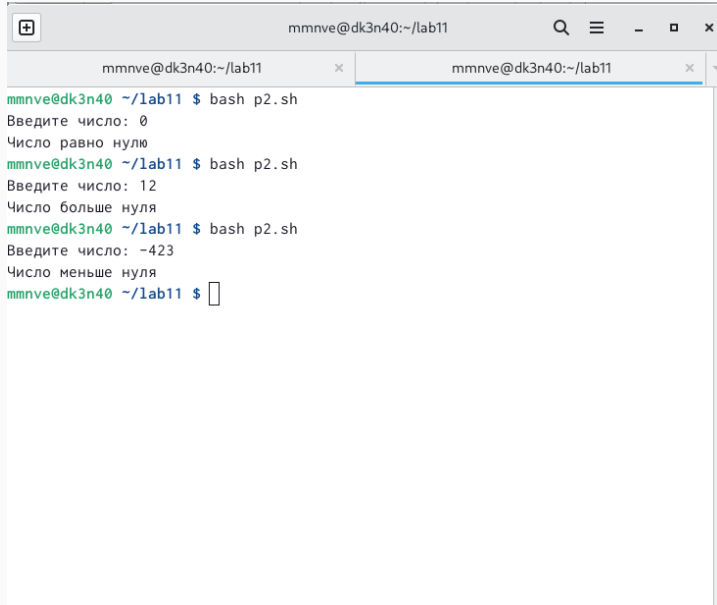
```
Открыть 1.c ~/tab11 Сохранить
p1.sh x output x *p2.sh x 1.c x
1 #include <stdlib.h>
2 #include <stdio.h>
3 int main ()
4 {
5     int n;
6     printf ("Введите число: ");
7     scanf ("%d",&n);
8     if (n>0){
9         exit(1);
10    }
11    else if (n==0) {
12        exit(0);
13    }
14    else {
15        exit(2);
16    }
```



The image shows a code editor window with three tabs: 'p1.sh', 'output', and 'p2.sh'. The 'p2.sh' tab is active and displays a shell script. The script starts with a shebang line, followed by a compilation command, a directory change, and a case statement with three branches for zero, positive, and negative numbers, each using the 'echo' command to output a Russian message. The script ends with 'esac'.

```
1 #!/bin/bash
2
3 gcc -o cprog 1.c
4 ./cprog
5 case $? in
6     0) echo "Число равно нулю";;
7     1) echo "Число больше нуля";;
8     2) echo "Число меньше нуля";;
9 esac
```

Рис. 5: Программный файл



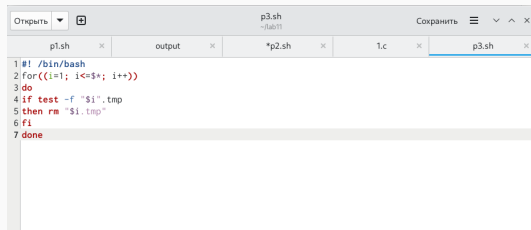
```
mmnve@dk3n40:~/lab11
mmnve@dk3n40 ~/lab11 $ bash p2.sh
Введите число: 0
Число равно нулю
mmnve@dk3n40 ~/lab11 $ bash p2.sh
Введите число: 12
Число больше нуля
mmnve@dk3n40 ~/lab11 $ bash p2.sh
Введите число: -423
Число меньше нуля
mmnve@dk3n40 ~/lab11 $
```

Рис. 6: Результат

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)



```
mmnve@dk3n40:~/lab11
mmnve@dk3n40 ~/lab11 $ bash p3.sh 5
mmnve@dk3n40 ~/lab11 $ ls
1.c  cprog  input  output  p1.sh  p2.sh  p3.sh
mmnve@dk3n40 ~/lab11 $ bash p3.sh 5
mmnve@dk3n40 ~/lab11 $ ls
1.c  cprog  input  output  p1.sh  p2.sh  p3.sh
mmnve@dk3n40 ~/lab11 $
```

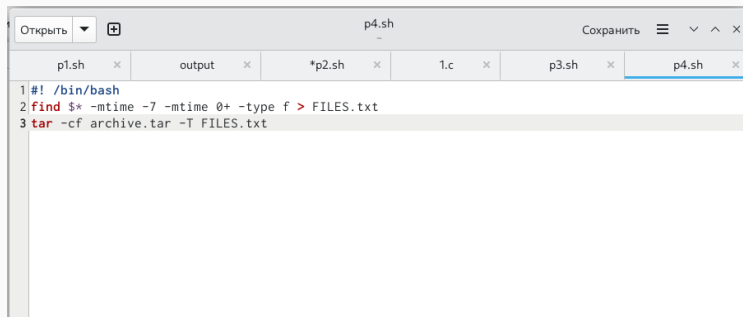


The image shows a terminal window with a light gray title bar. On the left, there is a button labeled 'Открыть' (Open) with a dropdown arrow and a plus icon. In the center, the title 'p3.sh' is displayed above the path '~/.lab/11'. On the right, there is a button labeled 'Сохранить' (Save) followed by three icons: a hamburger menu, a downward arrow, and an upward arrow. Below the title bar, there is a tab bar with five tabs: 'p1.sh', 'output', '*p2.sh', '1.c', and 'p3.sh'. The 'p3.sh' tab is selected and highlighted with a blue underline. The main area of the terminal displays a shell script with the following lines:

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

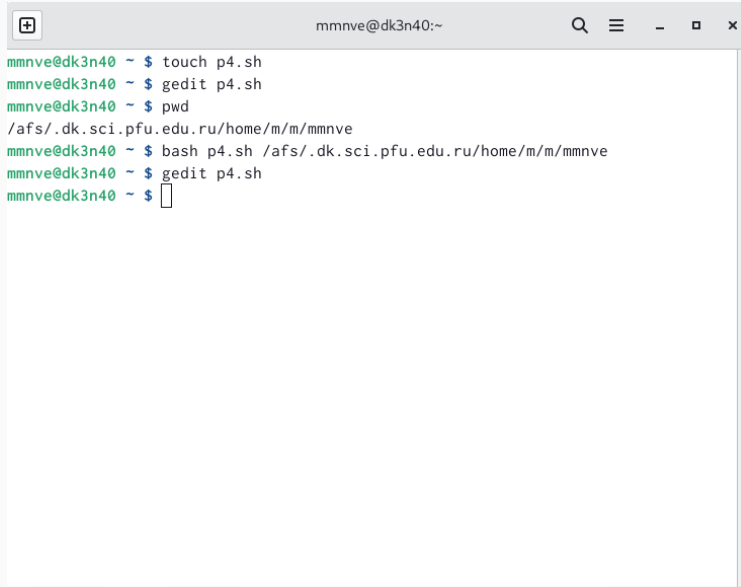
Рис. 8: Результат

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`)



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

Рис. 9: Файл и запуск



A terminal window with a title bar containing a search icon, a menu icon, and window control buttons. The title text is 'mmnve@dk3n40:~'. The terminal content shows a series of commands and their outputs:

```
mmnve@dk3n40 ~ $ touch p4.sh
mmnve@dk3n40 ~ $ gedit p4.sh
mmnve@dk3n40 ~ $ pwd
/afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve
mmnve@dk3n40 ~ $ bash p4.sh /afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve
mmnve@dk3n40 ~ $ gedit p4.sh
mmnve@dk3n40 ~ $
```

Рис. 10: Текст программы 4

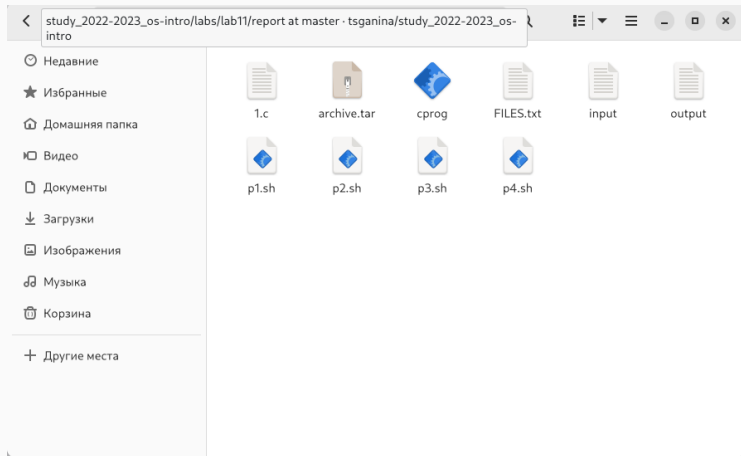


Рис. 11: Созданный архив и файл



The image shows a text editor window with a light gray header bar. On the left of the header is a button labeled 'Открыть' (Open) with a dropdown arrow and a plus icon. In the center of the header is the file name 'FILES.txt' and the path '-/lab11'. On the right is a button labeled 'Сохранить' (Save). The main area of the window contains a list of seven lines of text, each starting with a number from 1 to 7. Lines 1 through 6 are identical and show a file path: /afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve. Line 7 is a terminal prompt: mmnve@dk3n40 ~/work/study/2022-2023/System/os-intro/labs/lab11/report. The text is in a monospaced font.

```
1 /afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve
2 /afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve
3 /afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve
4 /afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve
5 /afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve
6 /afs/.dk.sci.pfu.edu.ru/home/m/m/mmnve
7 mmnve@dk3n40 ~/work/study/2022-2023/System/os-intro/labs/lab11/report
```

Рис. 12: FILES.txt

Результаты

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