

Отчёт по лабораторной работе №11

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

Леонов Владислав Александрович НПМбд-01-21

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	8
4	Контрольные вопросы	9

List of Figures

2.1	Задание 1	5
2.2	Задание 2	6
2.3	Задание 3	6
2.4	Задание 4	7

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` напишем командный файл, который анализирует командную строку с ключами и выполним его: `-i inputfile` — прочитать данные из указанного файла; `-o outputfile` — вывести данные в указанный файл; `-p шаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк;

а затем ищет в указанном файле нужные строки

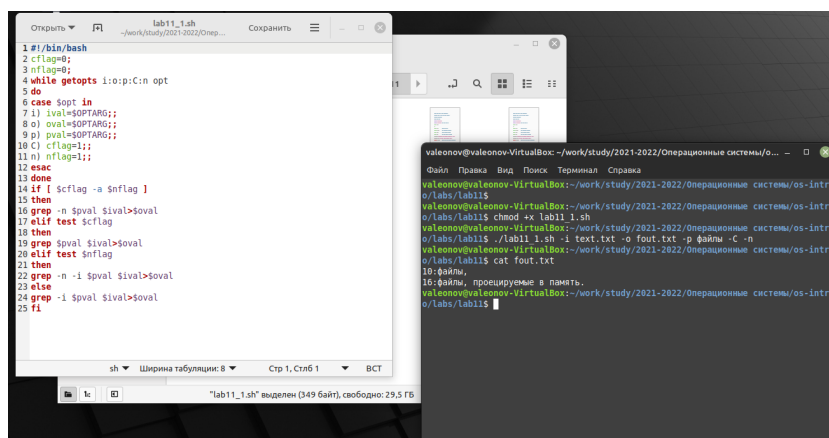


Figure 2.1: Задание 1

2. Напишем сначала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем завершим программу при помощи функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл вызовет эту программу и, проанализировав с помощью команды `$?`, выдаст сообщение о том, какое число было введено

The image shows a code editor with a file named `script2.c` and a terminal window. The code in `script2.c` is a C program that takes a number `o` as input and prints a message based on its value: positive, negative, or zero. The terminal shows the output of the program for various inputs.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main ()
5 {
6     int o;
7     scanf ("%d", &o);
8     if (o < 0) {
9         printf ("отрицательное\n");
10    } else if (o > 0) {
11        printf ("положительное\n");
12    } else {
13        printf ("равно нулю\n");
14    }
15 }
```

```
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/о...
Настраивается пакет libgcc-11-dev:amd64 (11.3.0-1ubuntu1-22.04) _
Настраивается пакет gcc-11 (11.3.0-1ubuntu1-22.04) _
Настраивается пакет libc6-dev:amd64 (2.35-0ubuntu3.1) _
Настраивается пакет libstdc++-11-dev:amd64 (11.3.0-1ubuntu1-22.04) _
Настраивается пакет g++-11 (11.3.0-1ubuntu1-22.04) _
Настраивается пакет g++ (4:11.2.0-1ubuntu1) _
update-alternatives: используется /usr/bin/g++ для предоставления /usr/bin/c++ (
c++) в автоматическом режиме
Обрабатываются триггеры для man-db (2.10.2-1) _
Обрабатываются триггеры для libc-bin (2.35-0ubuntu3) _
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intr
o/Labs/Lab11$ ./lab11_2.sh
3
положительное
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intr
o/Labs/Lab11$ ./lab11_2.sh
-8
отрицательное
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intr
o/Labs/Lab11$ ./lab11_2.sh
0
равно нулю
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intr
o/Labs/Lab11$
```

Figure 2.2: Задание 2

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N

The image shows a code editor with a file named `lab11_3.sh` and a terminal window. The script `lab11_3.sh` creates a series of files in the current directory, numbered from 1 to N. The terminal shows the execution of the script and the resulting files.

```
1 #!/bin/bash
2 let i=$1+1
3 while (( i==1 ))
4 do touch $i.tmp
5 done
6 let j=$2+1
7 while (( j==1 ))
8 do rm $j.tmp
9 done
```

```
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/о...
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intr
o/Labs/Lab11$ chmod +x lab11_3.sh
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intr
o/Labs/Lab11$ ./lab11_3.sh 3
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intr
o/Labs/Lab11$ ls
1.tmp 3.tmp lab11_1.sh lab11_3.sh presentation script2 script2.o
2.tmp fout.txt lab11_2.sh lab11_4.sh report script2.c text.txt
valeonov@valeonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intr
o/Labs/Lab11$
```

Figure 2.3: Задание 3

4. Напишем командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад.

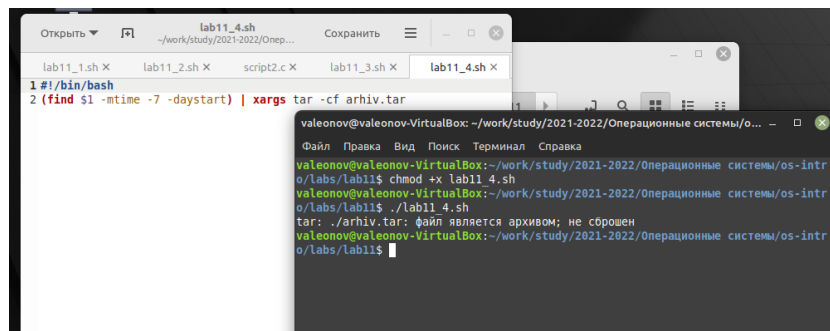


Figure 2.4: Задание 4

3 Вывод

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

4 Контрольные вопросы

1. Каково предназначение команды `getopts`? Ответ: Создание по пользовательским аргументам.
2. Какое отношение метасимволы имеют к генерации имён файлов? Ответ: Используют как файлы так и аргументы.
3. Какие операторы управления действиями вы знаете? Ответ: `If`, `else`, `elif`, `fi`, `while`, `do`, `done`, `until`, `do`, `done`, `for`, `in`, `do`, `done`, `case`, `in`, `esac`
4. Какие операторы используются для прерывания цикла? Ответ:
 - a) `for` – будет выполнять действие до тех пор, пока есть объекты для выполнения.
 - b) `while` – выполняет действие до тех пор, пока условие является истинным.
 - c) `until` – будет выполняться пока условие не станет правдиво.
5. Для чего нужны команды `false` и `true`? Ответ: `until` – будет выполняться до тех пор, пока условие не станет `true`, т.е. пока оно не станет `false`.
6. Что означает строка `if test -f mans/i.$s`, встреченная в командном файле? Ответ: Проверяет если существует файл его размерность и тип с двумя разными расширениями, заменяя через переменные.

7. Объясните различия между конструкциями while и until. Ответ:

while – выполняет действие до тех пор, пока условие является истинным.

until – будет выполняться до тех пор, пока условие не станет истинным, т.е. пока оно false.