

Отчет по лабораторной работе №11

Дисциплина Операционные системы

Колобова Елизавета, гр. НММбд-01-22

Содержание

1	Цель работы	5
2	Контрольные вопросы	10
	Список литературы	12

Список иллюстраций

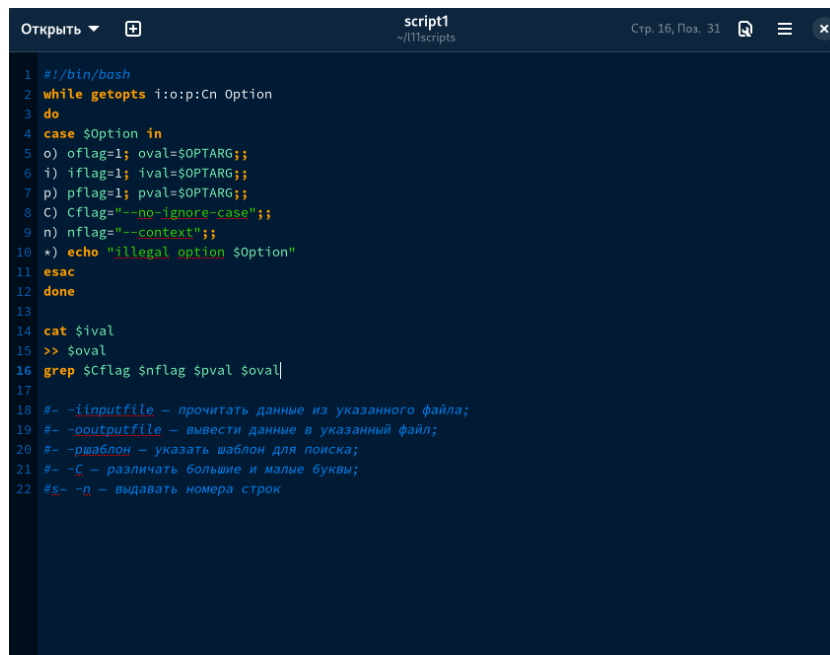
1.1	Рис. 1. Скрипт поиска по шаблону в задаваемом файле	6
1.2	Рис. 2. Программа для сравнения числа с нулем на Си	7
1.3	Рис. 3. Скрипт определения введенного числа	7
1.4	Рис. 4. Скрипт создания заданного количества файлов	8
1.5	Рис. 5. Скрипт архивирования файлов указанной директории . . .	9

Список таблиц

1 Цель работы

Цель работы - изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать сложные командные файлы с использованием логических управляющих конструкций и циклов. # Выполнение лабораторной работы

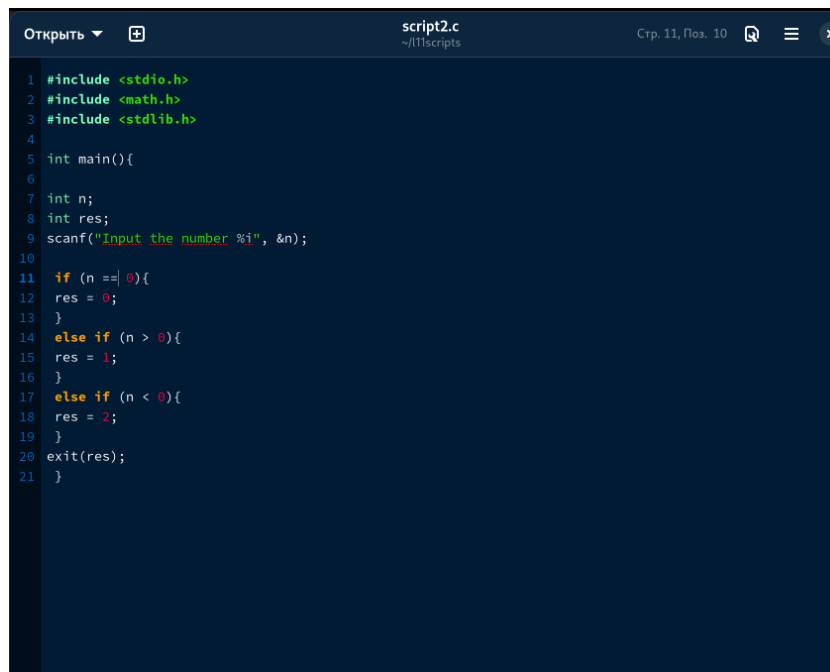
1. Используя команды `getopts` `grep`, напомним командный файл, который анализирует командную строку с ключами: `-i`inputfile — прочитать данные из указанного файла; `-o`outputfile — вывести данные в указанный файл; `-r`шаблон — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-p`. (рис. [1.1])



```
1 #!/bin/bash
2 while getopts i:o:p:Cn Option
3 do
4 case $Option in
5 o) oflag=1; oval=$OPTARG;;
6 i) iflag=1; ival=$OPTARG;;
7 p) pflag=1; pval=$OPTARG;;
8 C) Cflag="--no-ignore-case";
9 n) nflag="--context";
10 *) echo "illegal option $Option"
11 esac
12 done
13
14 cat $ival
15 >> $oval
16 grep $Cflag $nflag $pval $oval
17
18 #- -inputfile - прочитать данные из указанного файла;
19 #- -outputfile - вывести данные в указанный файл;
20 #- -ршаблон - указать шаблон для поиска;
21 #- -C - различать большие и малые буквы;
22 #- -n - выдавать номера строк
```

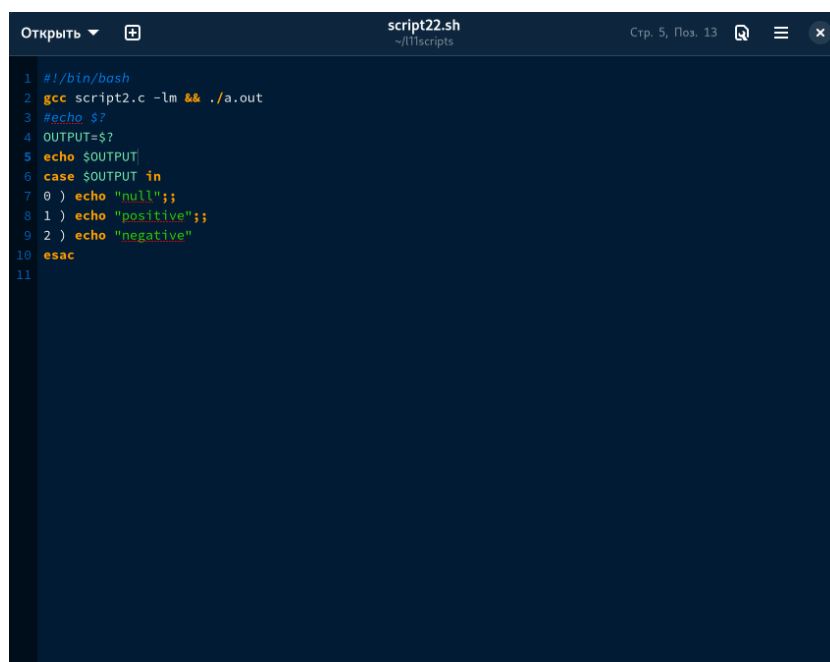
Рис. 1.1: Рис. 1. Скрипт поиска по шаблону в задаваемом файле

2. Напишем 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 вызывает эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено. (положительное, отрицательное или нуль) (рис. [1.2], [1.3])



```
1 #include <stdio.h>
2 #include <math.h>
3 #include <stdlib.h>
4
5 int main(){
6
7     int n;
8     int res;
9     scanf("Input the number %i", &n);
10
11     if (n == 0){
12         res = 0;
13     }
14     else if (n > 0){
15         res = 1;
16     }
17     else if (n < 0){
18         res = 2;
19     }
20     exit(res);
21 }
```

Рис. 1.2: Рис. 2. Программа для сравнения числа с нулем на Си

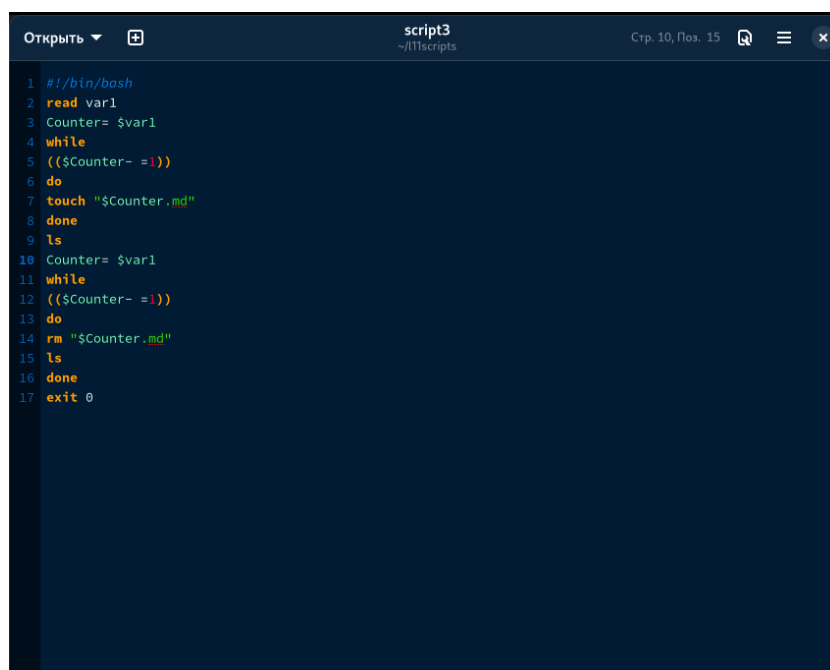


```
1 #!/bin/bash
2 gcc script2.c -lm && ./a.out
3 #echo $?
4 OUTPUT=$?
5 echo $OUTPUT
6 case $OUTPUT in
7 0 ) echo "null";;
8 1 ) echo "positive";;
9 2 ) echo "negative"
10 esac
11
```

Рис. 1.3: Рис. 3. Скрипт определения введенного числа

3. Напишем командный файл, создающий указанное число файлов, пронуме-

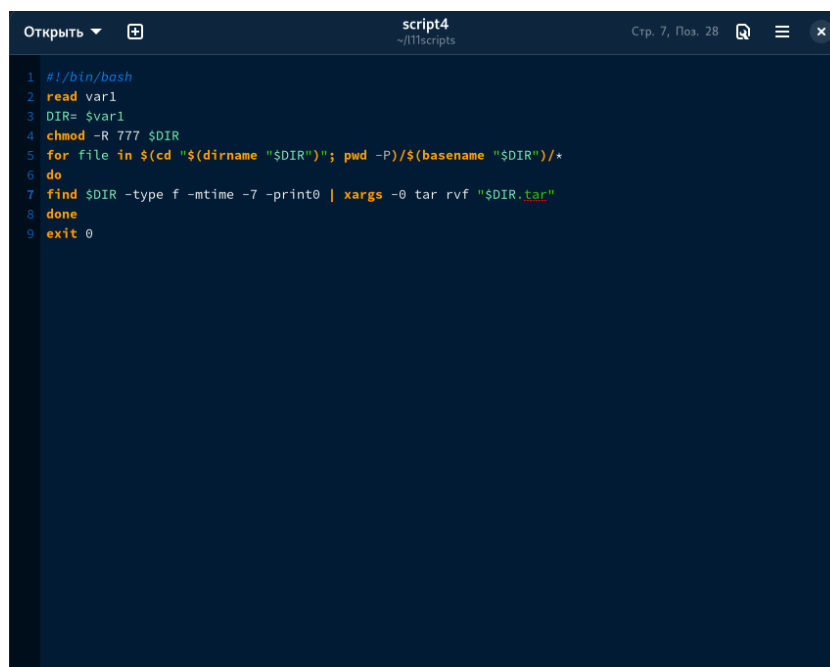
рованных последовательно от 1 до N (1.tmp, 2.tmp, 3.tmp,4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же ко- мандный файл удаляет все созданные им файлы (если они существуют). (рис. [1.4])



```
1 #!/bin/bash
2 read var1
3 Counter= $var1
4 while
5 (($Counter- =1))
6 do
7 touch "$Counter.md"
8 done
9 ls
10 Counter= $var1
11 while
12 (($Counter- =1))
13 do
14 rm "$Counter.md"
15 ls
16 done
17 exit 0
```

Рис. 1.4: Рис. 4. Скрипт создания заданного количества файлов

4. Напишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Он модифицирован так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад . (рис. [1.5])



```
1 #!/bin/bash
2 read var1
3 DIR= $var1
4 chmod -R 777 $DIR
5 for file in $(cd "$(dirname "$DIR"); pwd -P)/$(basename "$DIR")/*
6 do
7 find $DIR -type f -mtime -7 -print0 | xargs -0 tar rvf "$DIR.tar"
8 done
9 exit 0
```

Рис. 1.5: Рис. 5. Скрипт архивирования файлов указанной директории

2 Контрольные вопросы

1. Каково предназначение команды `getopts`? осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.
2. Какое отношение метасимволы имеют к генерации имён файлов? После всех подстановок в каждом слове команды ищутся символы `?`, `*` и `t.d`. Если находится хотя бы один из них, то это слово рассматривается как шаблон имен файлов и заменяется именами файлов, удовлетворяющих данному шаблону (в алфавитном порядке). Если ни одно имя файла не удовлетворяет шаблону, то он остается неизменным. Значения указанных символов: любая строка, включая и пустую `?` один любой символ [...] любой из указанных между ними символов. Пара символов, разделенных знаком `-`, означает любой символ, который находится между ними, включая и их самих. Если первым символом после `[` идет `!`, то указанные символы не должны входить в имя файла
3. Какие операторы управления действиями вы знаете? Точка с запятой (`;`), Амперсанд (`&`), Символ доллара со знаком вопроса (`$?`), Двойной амперсанд (`&&`), Двойная вертикальная черта (`||`), Знак фунта (`#`), Экранирование специальных символов (`\`)
4. Какие операторы используются для прерывания цикла? Команда `break` завершает выполнение цикла, а команда `continue` завершает данную итерацию блока операторов.
5. Для чего нужны команды `false` и `true`? команда `true` всегда возвращает код

завершения, равный нулю (т.е. истина), команда `false` всегда возвращает код завершения, не равный нулю (т. е. ложь).

6. Что означает строка `if test -f mans/i.$s`, встреченная в командном файле?
7. Объясните различия между конструкциями `while` и `until`. При замене в операторе цикла `while` служебного слова `while` на `until` условие, при выполнении которого осуществляется выход из цикла, меняется на противоположное. В остальном оператор цикла `while` и оператор цикла `until` идентичны. #

Выводы

Результатом проделанной работы является изучение основ программирования в оболочке ОС UNIX/Linux.

Список литературы