

Отчет по лабораторной работе №13

Дисциплина:Операционные системы

Абрамова Ульяна Михайловна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	13
	Список литературы	14

Список иллюстраций

3.1	1	7
3.2	2	8
3.3	3	9
3.4	4	10
3.5	5	10
3.6	6	11
3.7	7	11
3.8	8	11
3.9	9	12

Список таблиц

1 Цель работы

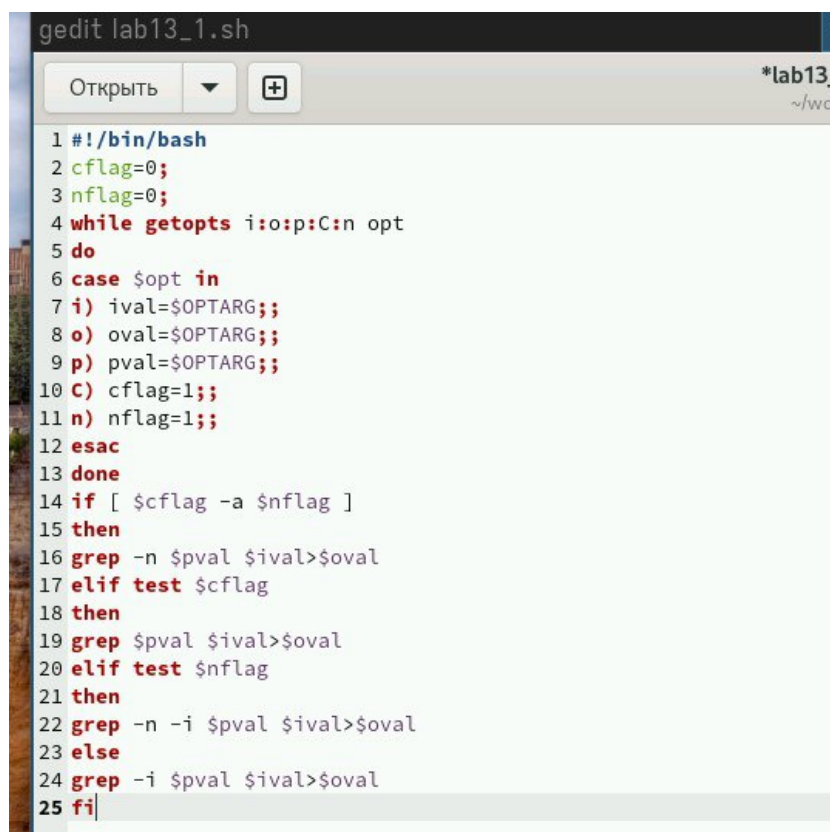
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

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; `-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`.
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 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).
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. 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 (рис. 3.1, 3.2).



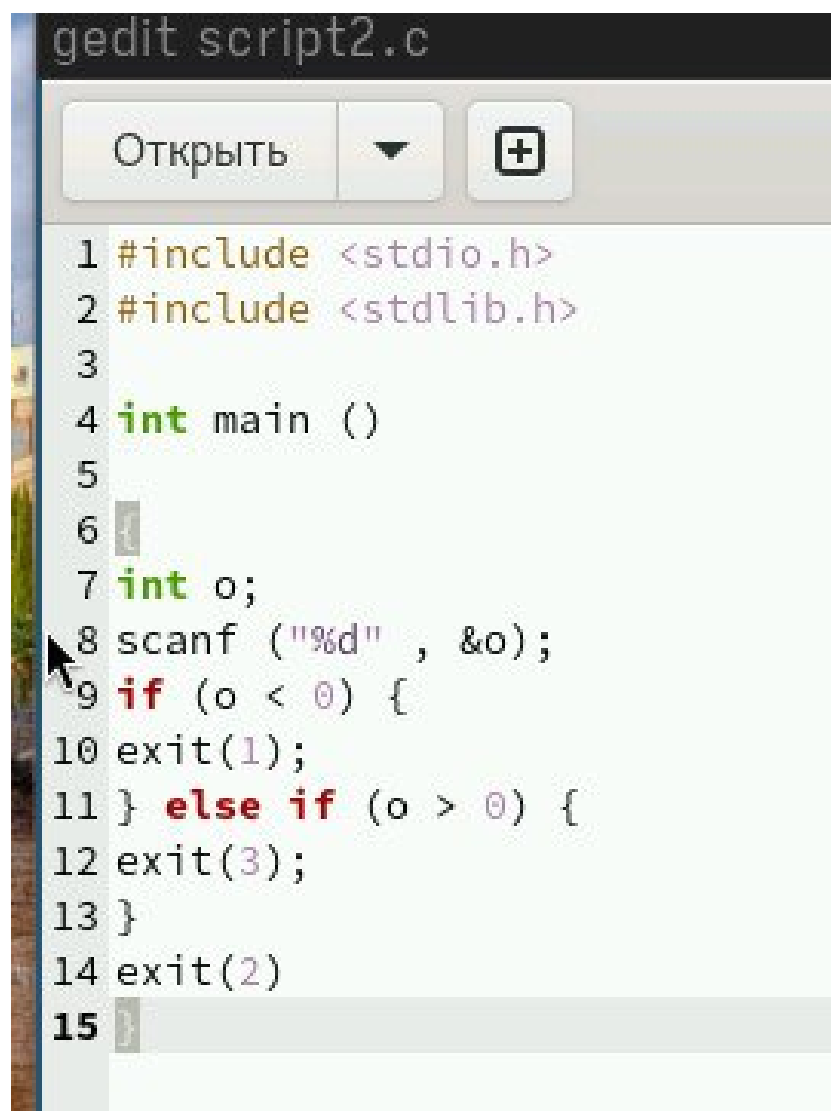
```
gedit lab13_1.sh
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:o:p:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval $ival>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $ival>$oval
23 else
24 grep -i $pval $ival>$oval
25 fi
```

Рис. 3.1: 1

```
[umabramova@umabramova work]$ ./lab13_1.sh -i conf.txt -o fout.txt -p files -C -n
grep: conf.txt: Нет такого файла или каталога
[umabramova@umabramova work]$ ls
fout.txt  git-extended  lab12_1.sh  lab12_2.sh  lab12_3.sh  lab12_4.sh  lab13_1.sh
[umabramova@umabramova work]$ touch lab13_2.sh
```

Рис. 3.2: 2

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 (рис. 3.3, 3.4, 3.5).

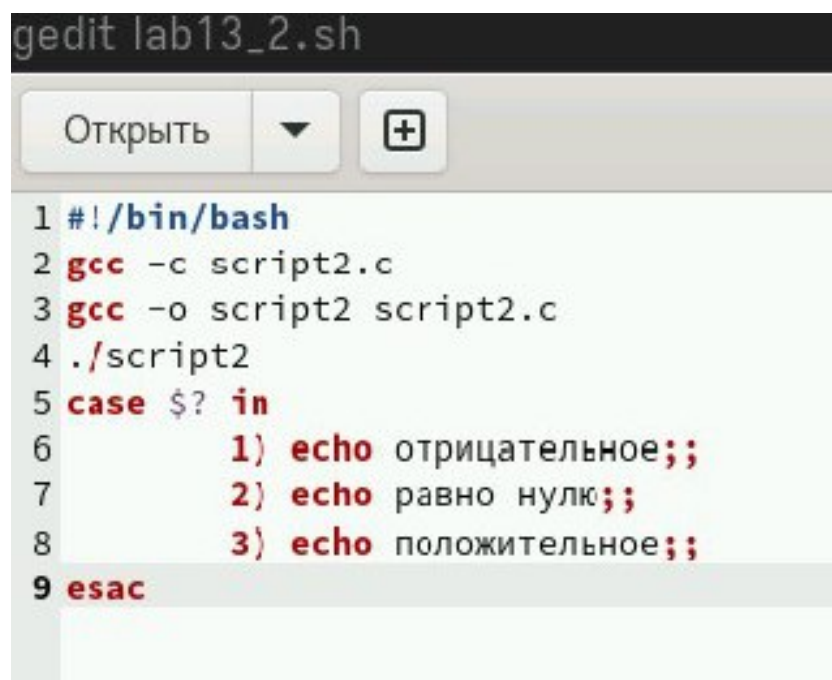


```
gedit script2.c

Открыть ▼ +

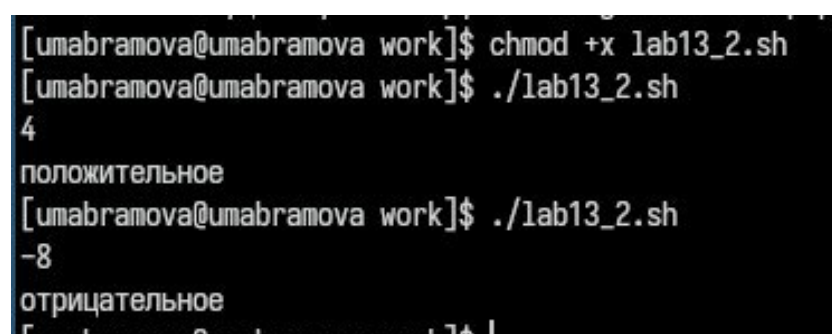
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main ()
5
6 {
7     int o;
8     scanf ("%d" , &o);
9     if (o < 0) {
10 exit(1);
11 } else if (o > 0) {
12 exit(3);
13 }
14 exit(2)
15 }
```

Рис. 3.3: 3



```
gedit lab13_2.sh
Открыть ▼ +
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6     1) echo отрицательное;;
7     2) echo равно нулю;;
8     3) echo положительное;;
9 esac
```

Рис. 3.4: 4



```
[umabramova@umabramova work]$ chmod +x lab13_2.sh
[umabramova@umabramova work]$ ./lab13_2.sh
4
положительное
[umabramova@umabramova work]$ ./lab13_2.sh
-8
отрицательное
[umabramova@umabramova work]$
```

Рис. 3.5: 5

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (рис. 3.6, 3.7).


```
[umabramova@umabramova work]$ chmod +x lab13_4.sh
[umabramova@umabramova work]$ ./lab13_4.sh
tar: ./archiv.tar: archive cannot contain itself; not dumped
[umabramova@umabramova work]$ ls
1.tmp  3.tmp  5.tmp      fout.txt  lab12_1.sh  lab12_3.s
2.tmp  4.tmp  archiv.tar  git-extended  lab12_2.sh  lab12_4.s
```

Рис. 3.9: 9

4 Выводы

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

Список литературы

1. Операционные системы