

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Ахмед Нурыев¹

17 апреля, 2024, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

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

Задачи лабораторной работы

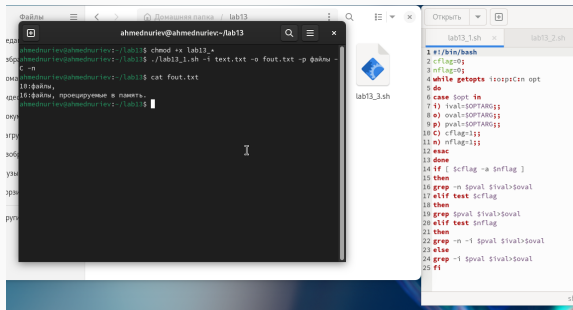
1 Выполнить 4 задания

Процесс выполнения 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; `-r шаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк;

а затем ищет в указанном файле нужные строки

Выполнение работы



The image shows a desktop environment with a terminal window and a file editor. The terminal window, titled 'ahmednuriev@ahmednuriev:~/lab13', displays the following commands and output:

```
ahmednuriev@ahmednuriev:~/lab13$ chmod +x lab13.sh
ahmednuriev@ahmednuriev:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p 0x00000000 -c -n
ahmednuriev@ahmednuriev:~/lab13$ cat fout.txt
10:0x00000000, прочитываем в память.
ahmednuriev@ahmednuriev:~/lab13$
```

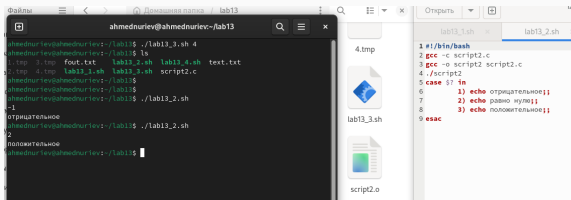
The file editor, titled 'lab13_1.sh', shows the following shell script code:

```
#!/bin/bash
1 cflag=0
2 nflag=0
3 while getopts isoprcin opt
4 do
5 case $opt in
6 i) ival=$OPTARG;;
7 f) 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. 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

Выполнение работы



The screenshot shows a terminal window and a file explorer. The terminal window, titled 'ahmednuriev@ahmednuriev:~/lab13', displays the following commands and output:

```
ahmednuriev@ahmednuriev:~/lab13$ ./lab13_3.sh 4
ahmednuriev@ahmednuriev:~/lab13$ ls
1.tmp  4.tmp  fout.txt  lab13_2.sh  lab13_4.sh  text.txt
ahmednuriev@ahmednuriev:~/lab13$ ./lab13_3.sh script2.c
ahmednuriev@ahmednuriev:~/lab13$
ahmednuriev@ahmednuriev:~/lab13$ ./lab13_2.sh
-1
отрицательное
ahmednuriev@ahmednuriev:~/lab13$ ./lab13_2.sh
2
положительное
ahmednuriev@ahmednuriev:~/lab13$
```

The file explorer shows the contents of the 'lab13_2.sh' script:

```
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. 2: Задание 2

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

Выполнение работы

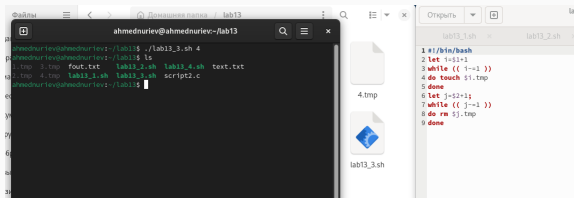


Рис. 3: Задание 3

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.

Выполнение работы

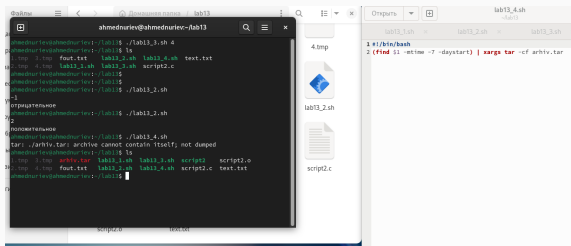


Рис. 4: Задание 4

Выводы по проделанной работе

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