

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Роберт Кармазян ¹

21 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

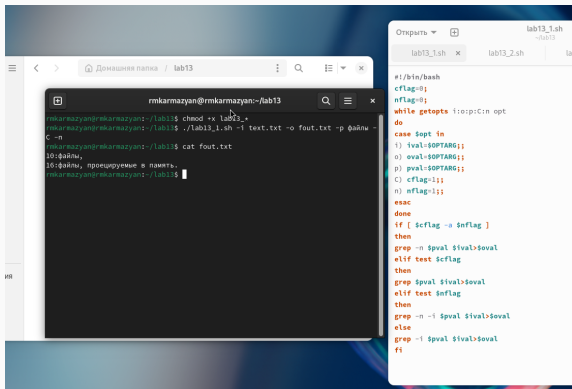
1 Выполнить 4 задания

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 terminal window and a code editor. The terminal window, titled 'rmkarmazyan@rmkarmazyan:~/lab13', displays the following commands and output:

```
rmkarmazyan@rmkarmazyan:~/lab13$ chmod +x lab13_1.sh
rmkarmazyan@rmkarmazyan:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -
C -n
rmkarmazyan@rmkarmazyan:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
rmkarmazyan@rmkarmazyan:~/lab13$
```

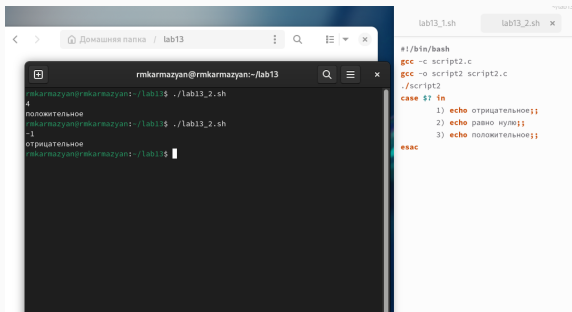
The code editor, titled 'lab13_1.sh', shows the following script:

```
#!/bin/bash
cflag=0
nflag=0
while getopts iio:p:c:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -o $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i $pval $ival>$oval
else
grep -i $pval $ival>$oval
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 image shows a terminal window and a code editor. The terminal window, titled 'rmkarmazyan@rmkarmazyan:~/lab13', displays the execution of a script named 'lab13_2.sh'. The script takes an argument '4' and prints 'положительное' (positive). It then takes an argument '-1' and prints 'отрицательное' (negative). The code editor, titled 'lab13_2.sh', shows the script's content: it uses 'gcc' to compile 'script2.c' into 'script2', then runs './script2' with a 'case' statement that echoes 'отрицательное' for negative numbers, 'равно нулю' (equal to zero) for zero, and 'положительное' for positive numbers.

```
rmkarmazyan@rmkarmazyan:~/lab13$ ./lab13_2.sh
4
положительное
rmkarmazyan@rmkarmazyan:~/lab13$ ./lab13_2.sh
-1
отрицательное
rmkarmazyan@rmkarmazyan:~/lab13$
```

```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
    1) echo отрицательное;;
    2) echo равно нулю;;
    3) echo положительное;;
esac
```

Рис. 2: Задание 2

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

Выполнение работы

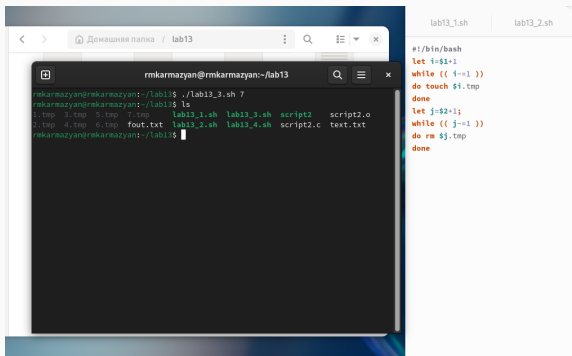
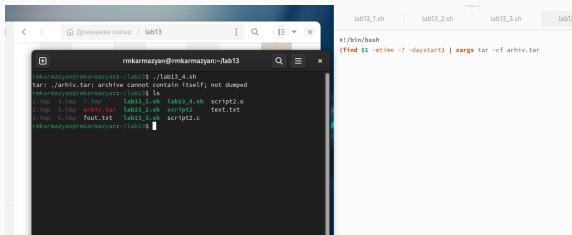


Рис. 3: Задание 3

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



The image shows two terminal windows. The left window is titled 'rmkarmazyan@rmkarmazyan:~\$' and shows the output of a tar command. The right window is titled 'lab13_1.sh' and shows a shell prompt and a command.

```
rmkarmazyan@rmkarmazyan:~$ tar -xvf arhiv.tar
tar: ./arhiv.tar: archive cannot contain itself; not dumped
rmkarmazyan@rmkarmazyan:~$ ls
1.tmp  4.tmp  7.tmp  lab13_1.sh  lab13_4.sh  script2.o
2.tmp  5.tmp  arhiv.tar  lab13_2.sh  script2  text.txt
3.tmp  6.tmp  fout.txt  lab13_3.sh  script2.c
rmkarmazyan@rmkarmazyan:~$
```

```
#!/bin/bash
(find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar
```

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.