

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Маныев Ресулбег¹

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

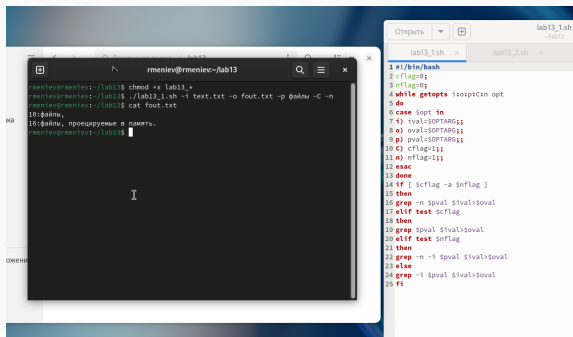
1 Выполнить 4 задания

Процесс выполнения лабораторной работы

1. Используя команды `getopts` `grep` напомним 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 script file. The terminal window, titled 'rmeniev@rmeniev:~/lab13', displays the following commands and output:

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

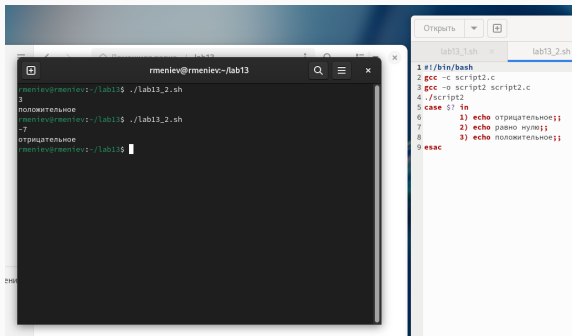
The script file, titled 'lab13_1.sh', contains the following code:

```
1 #!/bin/bash
2 cflag=0
3 nflag=0
4 while getopts 'io:psC:n opt'
5 do
6 case $opt in
7 i) sval=$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 $sval>$oval
17 elif test $cflag
18 then
19 grep $pval $sval>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $sval>$oval
23 else
24 grep -i $pval $sval>$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 image shows a terminal window and a code editor. The terminal window, titled 'rmeniev@rmeniev:~/lab13', displays the execution of a script 'lab13_2.sh'. The script takes an argument and prints 'положительное' (positive) for '3', 'отрицательное' (negative) for '-7', and nothing for '0'. The code editor, titled 'lab13_1.sh' and 'lab13_2.sh', shows the source code of 'lab13_2.sh'.

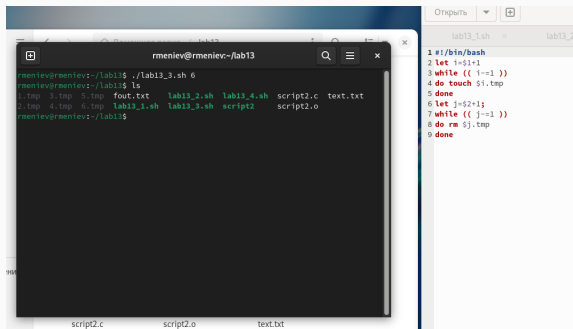
```
rmeniev@rmeniev:~/lab13$ ./lab13_2.sh 3
положительное
rmeniev@rmeniev:~/lab13$ ./lab13_2.sh -7
отрицательное
rmeniev@rmeniev:~/lab13$
```

```
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



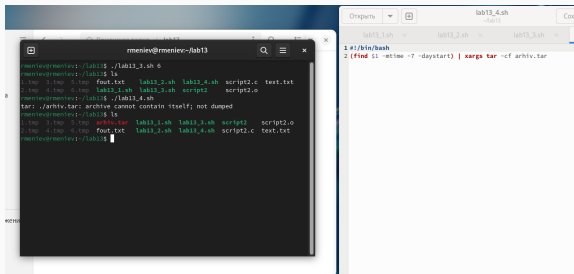
The image shows a terminal window and a code editor. The terminal window, titled 'rmeniev@rmeniev:~/lab13', displays the execution of a shell script 'lab13_3.sh' with argument '6'. The script's output lists files in the current directory: '1.tmp', '3.tmp', '5.tmp', 'fout.txt', 'lab13_2.sh', 'lab13_4.sh', 'script2.c', and 'text.txt'. The code editor, titled 'lab13_1.sh' and 'lab13_2...', shows the following script content:

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

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 windows from a Linux desktop environment. The left window is a terminal titled 'rmeniev@rmeniev:~/lab13'. It displays the execution of a script 'lab13_3.sh' with arguments '6', followed by 'ls' and 'tar' commands. The right window is a file manager titled 'lab13_4.sh' showing the contents of a file named 'arhiv.tar'.

```
rmeniev@rmeniev:~/lab13$ ./lab13_3.sh 6
rmeniev@rmeniev:~/lab13$ ls
1.tmp  2.tmp  3.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
2.tmp  4.tmp  5.tmp  lab13_3.sh  lab13_3.sh  script2.c
rmeniev@rmeniev:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
rmeniev@rmeniev:~/lab13$ ls
1.tmp  3.tmp  4.tmp  arhiv.tar  lab13_1.sh  lab13_3.sh  script2.c  script2.o
2.tmp  4.tmp  5.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
rmeniev@rmeniev:~/lab13$
```

```
lab13_4.sh
-rw-r--r-- 1 rmeniev rmeniev 4096 10.11.2023 10:11 arhiv.tar
1 #!/bin/bash
2 (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.