

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Иван Волобуев¹

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

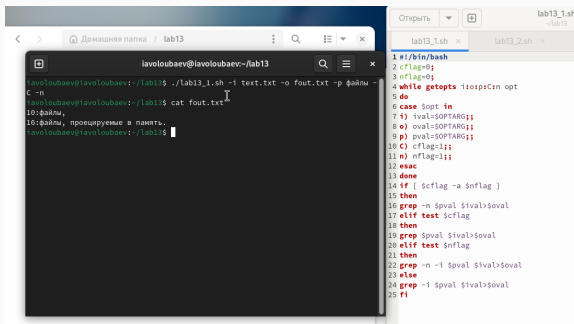
1 Выполнить 4 задания

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 script file. The terminal window is titled 'iavoloubaev@iavoloubaev:~/lab13' and shows the execution of a script 'lab13_1.sh' with arguments '-i text.txt -o fout.txt -p файлы -C -n'. The output of the script is displayed in the terminal, showing the contents of 'fout.txt' which are '10:файлы,' and '10:файлы, проецируемые в память.' The script file 'lab13_1.sh' is shown on the right, containing a shell script that processes command-line arguments and uses 'grep' to search for patterns in a file.

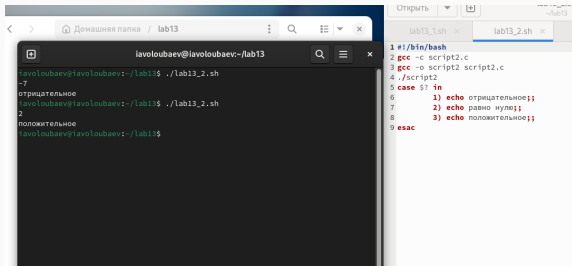
```
iavoloubaev@iavoloubaev:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
iavoloubaev@iavoloubaev:~/lab13$ cat fout.txt
10:файлы,
10:файлы, проецируемые в память.
iavoloubaev@iavoloubaev:~/lab13$
```

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:osp: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. 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 on the left and a code editor on the right. The terminal window is titled 'iavouloubaev@iavouloubaev:~/lab13' and shows the execution of a script named 'lab13_2.sh'. The script's output is displayed in the terminal: '-7 отрицательное', '2', and 'положительное'. The code editor on the right shows the source code of 'lab13_2.sh', which is a shell script that compiles a C program and runs it with different arguments.

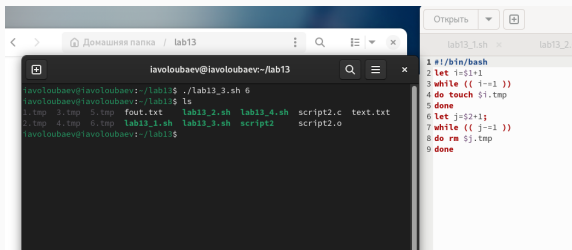
```
iavouloubaev@iavouloubaev:~/lab13$ ./lab13_2.sh
-7
отрицательное
iavouloubaev@iavouloubaev:~/lab13$ ./lab13_2.sh
2
положительное
iavouloubaev@iavouloubaev:~/lab13$
```

```
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



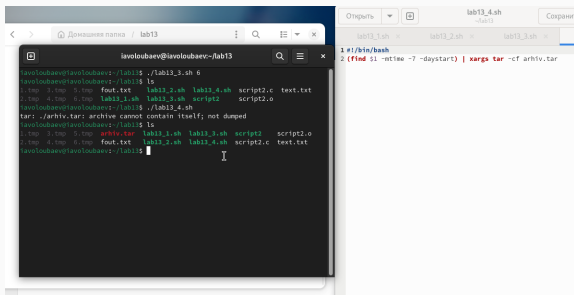
The image shows a terminal window and a file explorer. The terminal window, titled 'iavoloubaev@iavoloubaev:~/lab13', displays the execution of a script 'lab13_3.sh' with argument '6'. The script's output lists files and directories: '1.tmp', '2.tmp', '3.tmp', '4.tmp', '5.tmp', '6.tmp', 'fout.txt', 'lab13_1.sh', 'lab13_2.sh', 'lab13_3.sh', 'lab13_4.sh', 'script2', 'script2.c', 'script2.o', and 'text.txt'. The file explorer, titled 'Домашняя папка / lab13', shows the contents of the 'lab13' directory, including files 'lab13_1.sh' and 'lab13_2.sh'.

```
iavoloubaev@iavoloubaev:~/lab13$ ./lab13_3.sh 6
iavoloubaev@iavoloubaev:~/lab13$ ls
1.tmp  2.tmp  3.tmp  4.tmp  5.tmp  6.tmp  fout.txt  lab13_1.sh  lab13_2.sh  lab13_3.sh  lab13_4.sh  script2  script2.c  script2.o  text.txt
iavoloubaev@iavoloubaev:~/lab13$
```

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 a terminal window and a file manager. The terminal window, titled 'lavoloubaev@lavoloubaev:~/lab13', displays the following commands and output:

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

The file manager, titled 'lab13_4.sh', shows the following commands:

```
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.