

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Бахи Сиди Али Темассини¹

26 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

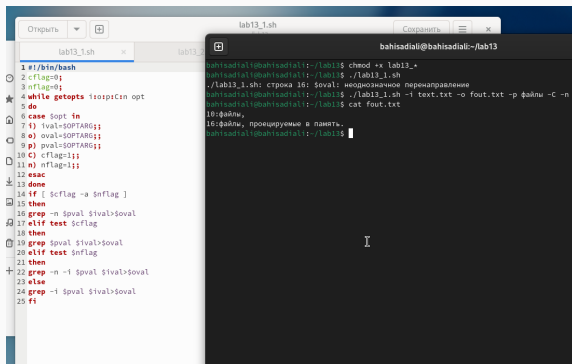
1 Выполнить 4 задания

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



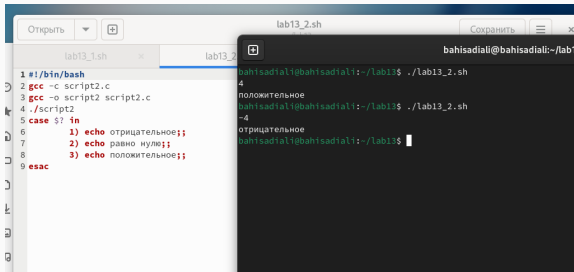
```
lab13_1.sh
1#!/bin/bash
2cflag=0;
3nflag=0;
4while getopts isospin opt
5do
6case $opt in
7i) ival=$OPTARG;;
8o) oval=$OPTARG;;
9p) pval=$OPTARG;;
10c) cflag=1;;
11n) nflag=1;;
12esac
13done
14if [ $cflag -a $nflag ]
15then
16grep -n $pval $ival>$oval
17elif test $cflag
18then
19grep $pval $ival>$oval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

```
bahisadiali@bahisadiali:~/lab13$ ./lab13_1.sh
bahisadiali@bahisadiali:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
bahisadiali@bahisadiali:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файл -C -n
bahisadiali@bahisadiali:~/lab13$ cat fout.txt
16:файл,
16:файл, проецируемые в память.
bahisadiali@bahisadiali:~/lab13$
```

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 code editor window with two tabs: 'lab13_1.sh' and 'lab13_2.sh'. The 'lab13_1.sh' tab is active, displaying a shell script. The script starts with a shebang, compiles 'script2.c' into 'script2', and then uses a 'case' statement to echo different messages based on the input. The 'lab13_2.sh' tab is also visible, showing the execution of the script with inputs '4' and '-4', resulting in the outputs 'положительное' and 'отрицательное' respectively.

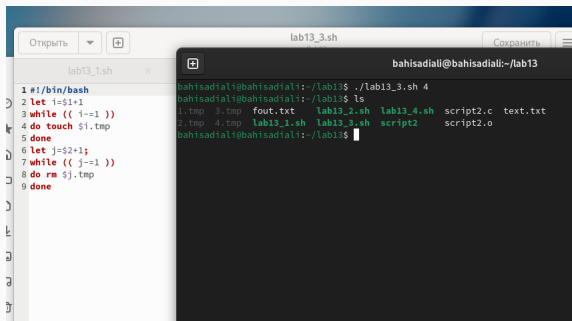
```
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6     1) echo отрицательное;;
7     2) echo равно нулю;;
8     3) echo положительное;;
9 esac
```

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

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 with two panes. The left pane displays the source code of a shell script named `lab13_1.sh`. The script contains a loop that increments a counter `i` and touches a file `$i.tmp` until `i` reaches 4. The right pane shows the execution of a script named `lab13_3.sh` in a directory `~/lab13`. The execution output shows the results of the script, including the execution of `ls` and the creation of files `fout.txt`, `script2.c`, and `script2.o`.

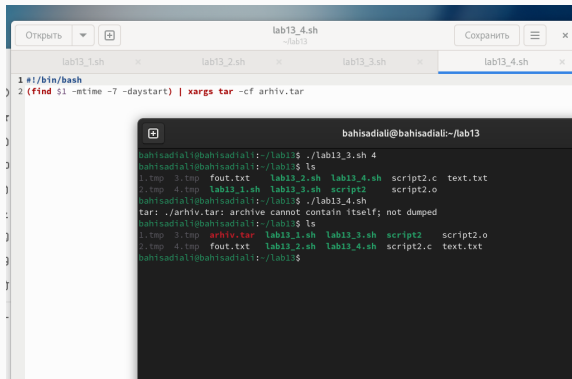
```
lab13_1.sh
1 #!/bin/bash
2 let i=$i+1
3 while (( i--=1 ))
4 do touch $i.tmp
5 done
6 let j=$2+1;
7 while (( j--=1 ))
8 do rm $j.tmp
9 done

bahisadiali@bahisadiali:~/lab13$ ./lab13_3.sh 4
bahisadiali@bahisadiali:~/lab13$ ls
1.tmp 3.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
2.tmp 4.tmp  lab13_1.sh  lab13_3.sh  script2    script2.o
bahisadiali@bahisadiali:~/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 screenshot shows a terminal window titled "lab13_4.sh" with a tab bar containing "lab13_1.sh", "lab13_2.sh", "lab13_3.sh", and "lab13_4.sh". The terminal content is as follows:

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

An inset terminal window shows the execution of the script:

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

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.