

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Дев Авинаш¹

18 апреля, 2023, 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

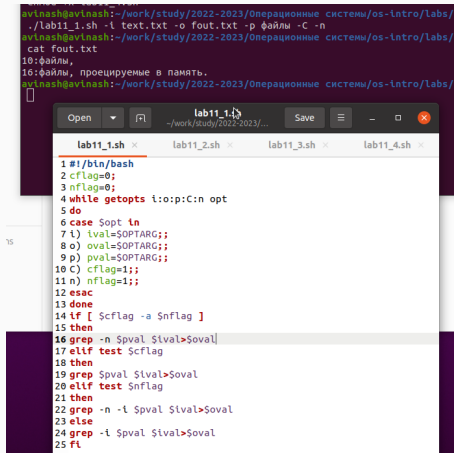
1. Используя команды `getopts` `grep` напишем командный файл, который анализирует командную строку с ключами и выполним его: `-i inputfile` — прочитать данные из указанного файла; `-o outputfile` — вывести данные в указанный файл; `-r шаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк;

а затем ищет в указанном файле нужные строки

Выполнение работы

```
avlnash@avlnash:~/work/study/2022-2023/Операционные системы/os-intro/labs/
./lab11_1.sh -l text.txt -o fout.txt -p файлы -C -n
avlnash@avlnash:~/work/study/2022-2023/Операционные системы/os-intro/labs/
cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
avlnash@avlnash:~/work/study/2022-2023/Операционные системы/os-intro/labs/

```

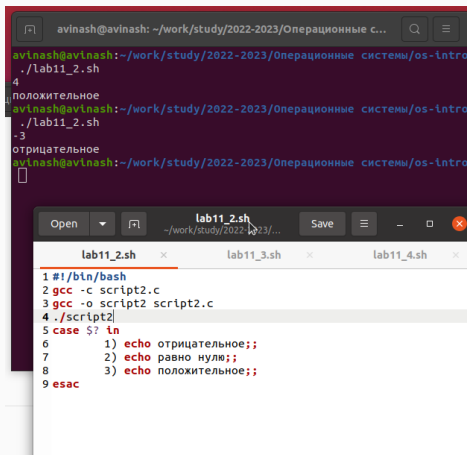


```
lab11_1.sh x lab11_2.sh x lab11_3.sh x lab11_4.sh x
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:o:p:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$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 and a script editor. The terminal window, titled 'avinash@avinash: ~/work/study/2022-2023/Операционные с...', displays the execution of a script named 'lab11_2.sh'. The script's output is as follows:

```
avinash@avinash:~/work/study/2022-2023/Операционные системы/os-intro$ ./lab11_2.sh
4
положительное
avinash@avinash:~/work/study/2022-2023/Операционные системы/os-intro$ ./lab11_2.sh
-3
отрицательное
avinash@avinash:~/work/study/2022-2023/Операционные системы/os-intro$
```

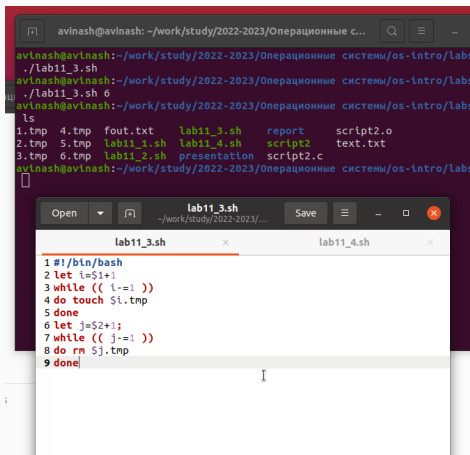
The script editor, titled 'lab11_2.sh', shows the following code:

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

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 editor. The terminal window displays the execution of a shell script named `lab11_3.sh`. The script creates a directory `tmp` and generates files `1.tmp` through `6.tmp`, each containing a specific command or file name. The file editor shows the content of `lab11_3.sh`, which is a bash script that uses nested loops to create these files.

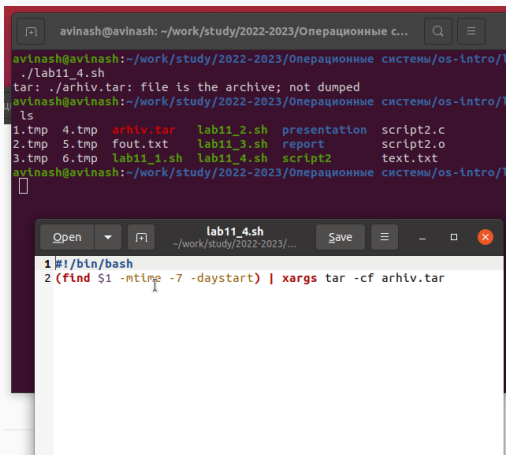
```
avlnash@avlnash: ~/work/study/2022-2023/Операционные системы/os-intro/labs
avlnash@avlnash:~/work/study/2022-2023/Операционные системы/os-intro/labs
./lab11_3.sh
avlnash@avlnash:~/work/study/2022-2023/Операционные системы/os-intro/labs
./lab11_3.sh 6
avlnash@avlnash:~/work/study/2022-2023/Операционные системы/os-intro/labs
ls
1.tmp 4.tmp fout.txt lab11_3.sh report script2.o
2.tmp 5.tmp lab11_1.sh lab11_4.sh script2 text.txt
3.tmp 6.tmp lab11_2.sh presentation script2.c
avlnash@avlnash:~/work/study/2022-2023/Операционные системы/os-intro/labs

lab11_3.sh
1 #!/bin/bash
2 let i=$1+1
3 while (( i-- ))
4 do touch $i.tmp
5 done
6 let j=$2+1
7 while (( j-- ))
8 do rm $j.tmp
9 done
```

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 editor. The terminal window, titled 'avinash@avinash: ~/work/study/2022-2023/Операционные с...', displays the following commands and output:

```
avinash@avinash:~/work/study/2022-2023/Операционные системы/os-intro/
./lab11_4.sh
tar: ./arhiv.tar: file is the archive; not dumped
avinash@avinash:~/work/study/2022-2023/Операционные системы/os-intro/
ls
1.tmp  4.tmp  arhiv.tar  lab11_2.sh  presentation  script2.c
2.tmp  5.tmp  fout.txt  lab11_3.sh  report        script2.o
3.tmp  6.tmp  lab11_1.sh lab11_4.sh  script2       text.txt
avinash@avinash:~/work/study/2022-2023/Операционные системы/os-intro/
```

The file editor, titled 'lab11_4.sh', shows the following script content:

```
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.