

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Амуничников Антон НПИбд-01-22¹

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

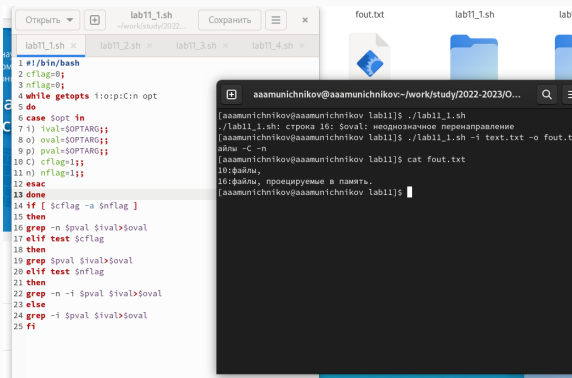
1 Выполнить 4 задания

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



The screenshot displays a Linux desktop environment. In the background, a file manager window shows a directory with files 'fout.txt', 'lab11_1.sh', and 'lab1'. In the foreground, a terminal window titled 'aaaamunichnikov@aaaamunichnikov:~/work/study/2022-2023/O...' shows the execution of a script 'lab11_1.sh'. The script contains a loop that sets a flag and performs a series of grep operations. The terminal output shows the script's execution steps and the final state of the flag.

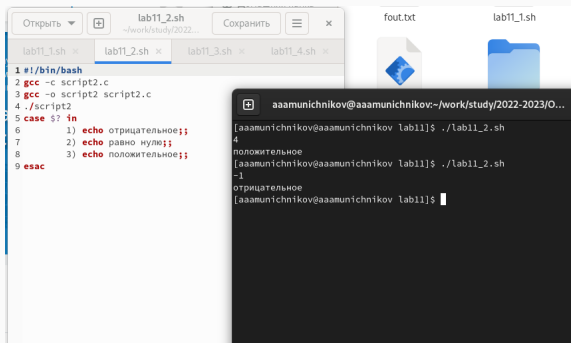
```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:o:p:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval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

```
[aaaamunichnikov@aaaamunichnikov lab11]$ ./lab11_1.sh
./lab11_1.sh: строка 16: $oval: неоднозначное перенаправление
[aaaamunichnikov@aaaamunichnikov lab11]$ ./lab11_1.sh -i text.txt -o fout.txt
айлы -C -n
[aaaamunichnikov@aaaamunichnikov lab11]$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
[aaaamunichnikov@aaaamunichnikov lab11]$
```

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 development environment with a code editor and a terminal. The code editor has tabs for lab1_1.sh, lab1_2.sh (active), lab1_3.sh, and lab1_4.sh. The active tab contains the following script:

```
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
```

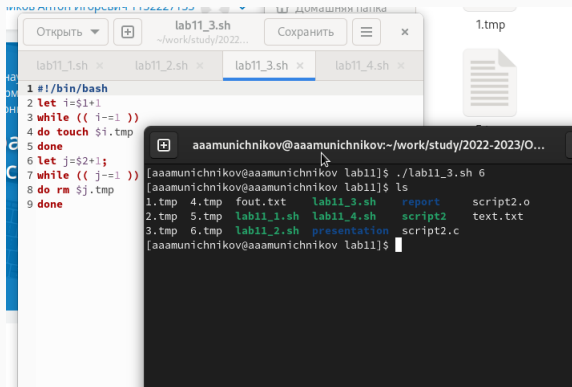
The terminal window shows the execution of the script. The user runs `./lab1_2.sh` twice. The first run outputs `положительное` (positive) because the exit code of `./script2` is 0. The second run outputs `отрицательное` (negative) because the exit code of `./script2` is 1.

```
aaamunichnikov@aaamunichnikov:~/work/study/2022-2023/O...
[aaamunichnikov@aaamunichnikov lab11]$ ./lab1_2.sh
4
положительное
[aaamunichnikov@aaamunichnikov lab11]$ ./lab1_2.sh
-1
отрицательное
[aaamunichnikov@aaamunichnikov lab11]$
```

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 a light gray title bar. The title bar contains the text "lab11_3.sh" and the path "~/work/study/2022...". Below the title bar, there are four tabs labeled "lab11_1.sh", "lab11_2.sh", "lab11_3.sh", and "lab11_4.sh". The "lab11_3.sh" tab is active. The terminal content shows a shell script being executed, followed by a command to run the script and a subsequent 'ls' command showing the files created.

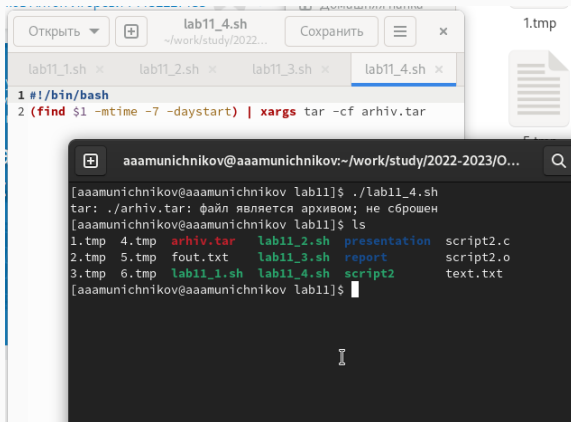
```
1 #!/bin/bash
2 let i=$1+1
3 while (( i-=1 ))
4 do touch $i.tmp
5 done
6 let j=$2+1;
7 while (( j-=1 ))
8 do rm $j.tmp
9 done

[aaamunichnikov@aaamunichnikov lab11]$ ./lab11_3.sh 6
[aaamunichnikov@aaamunichnikov lab11]$ 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
```

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 with a file explorer interface at the top. The file explorer has tabs for 'lab11_1.sh', 'lab11_2.sh', 'lab11_3.sh', and 'lab11_4.sh'. The 'lab11_4.sh' tab is active, showing the following commands:

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

Below the file explorer, a terminal window displays the output of the script execution:

```
[aaamunichnikov@aaamunichnikov lab11]$ ./lab11_4.sh
tar: ./arhiv.tar: файл является архивом; не сброшен
[aaamunichnikov@aaamunichnikov lab11]$ 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
[aaamunichnikov@aaamunichnikov lab11]$
```

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.