

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Сальников Даниил Александрович НБИбд-02-21¹

26 мая, 2022, 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

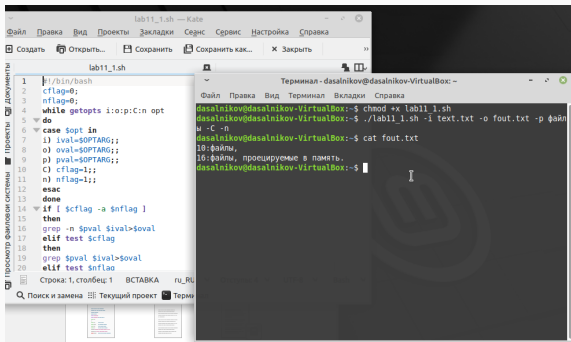
1 Выполнить 4 задания

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 screenshot of a development environment. On the left, a code editor window titled 'lab11_1.sh' contains a shell script. The script starts with a shebang, sets flags, and uses a while loop to process command-line options. It then uses a case statement to handle different options like -i, -o, -p, -C, and -n, and finally uses a series of if-then-else statements to process the flags and output the results. On the right, a terminal window titled 'Терминал - dasalnikov@dasalnikov-VirtualBox: -' shows the execution of the script. The user runs 'chmod +x lab11_1.sh', then './lab11_1.sh -i text.txt -o fout.txt -p файл'. The output shows the script processing the flags and outputting the results.

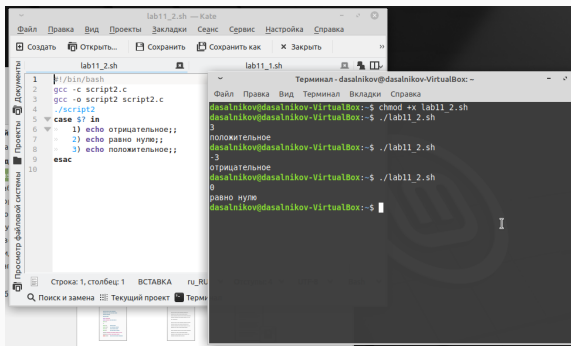
```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:op:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval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

```
dasalnikov@dasalnikov-VirtualBox:~$ chmod +x lab11_1.sh
dasalnikov@dasalnikov-VirtualBox:~$ ./lab11_1.sh -i text.txt -o fout.txt -p файл
-C -n
dasalnikov@dasalnikov-VirtualBox:~$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
dasalnikov@dasalnikov-VirtualBox:~$
```

Figure 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 screenshot displays a development environment with two windows. The left window, titled 'lab11_2.sh — Kate', shows a C program 'script2.c' that uses a switch statement to check if an input number is negative, zero, or positive. The right window is a terminal titled 'Терминал - dasalnikov@dasalnikov-VirtualBox: ~' where the program is compiled and executed. The terminal shows the execution of 'lab11_2.sh' with inputs 3, -3, 0, and a blank line, resulting in the corresponding Russian output: 'положительное', 'отрицательное', 'равно нулю', and a blank line.

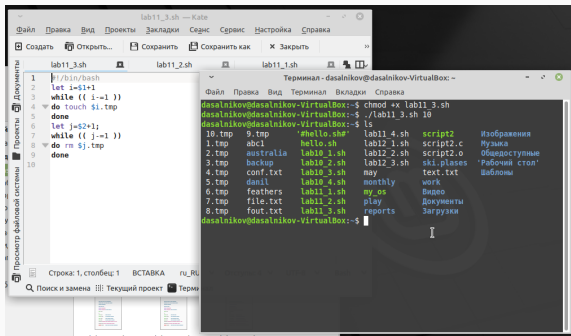
```
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6   1) echo отрицательное;;
7   2) echo равно нулю;;
8   3) echo положительное;;
9   esac
10
```

```
Терминал - dasalnikov@dasalnikov-VirtualBox: ~
Файл  Правка  Вид  Терминал  Вкладки  Справка
dasalnikov@dasalnikov-VirtualBox:~$ chmod +x lab11_2.sh
dasalnikov@dasalnikov-VirtualBox:~$ ./lab11_2.sh
3
положительное
dasalnikov@dasalnikov-VirtualBox:~$ ./lab11_2.sh
-3
отрицательное
dasalnikov@dasalnikov-VirtualBox:~$ ./lab11_2.sh
0
равно нулю
dasalnikov@dasalnikov-VirtualBox:~$
```

Figure 2: Задание 2

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N

Выполнение работы



The screenshot displays a text editor window titled 'lab11_3.sh — Kate' with a menu bar (Файл, Правка, Вид, Проекты, Закладки, Сессии, Сервис, Настройка, Справка) and a toolbar. The editor contains the following shell script:

```
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
10
```

Below the editor is a terminal window titled 'Терминал - dasalnikov@dasalnikov-VirtualBox: ~'. It shows the execution of the script:

```
dasalnikov@dasalnikov-VirtualBox:~$ chmod +x lab11_3.sh
dasalnikov@dasalnikov-VirtualBox:~$ ./lab11_3.sh 10
dasalnikov@dasalnikov-VirtualBox:~$ ls
10.tmp 9.tmp 'hello.sh' lab11_4.sh script2 Изображения
1.tmp abc1 hello.sh lab12_1.sh script2.c Музыка
2.tmp australia lab10_1.sh lab12_2.sh script2.o Общедоступные
3.tmp backup lab10_2.sh lab12_3.sh ski.places 'Рабочий стол'
4.tmp conf.txt lab10_3.sh may text.txt Виблоны
5.tmp danil lab10_4.sh monthly work
6.tmp feathers lab11_1.sh my_os Видео
7.tmp file.txt lab11_2.sh play Документы
8.tmp fout.txt lab11_3.sh reports Загрузки
dasalnikov@dasalnikov-VirtualBox:~$
```

Figure 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

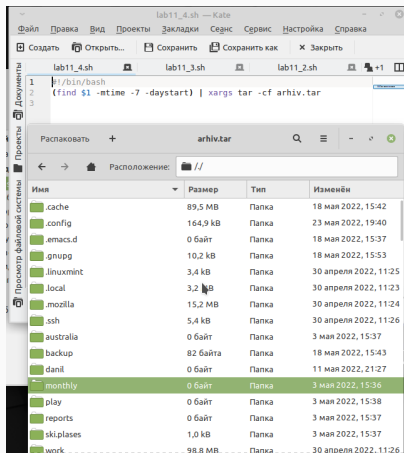


Figure 4: Задание 4

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.