

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Мартынов Николай Алексеевич 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

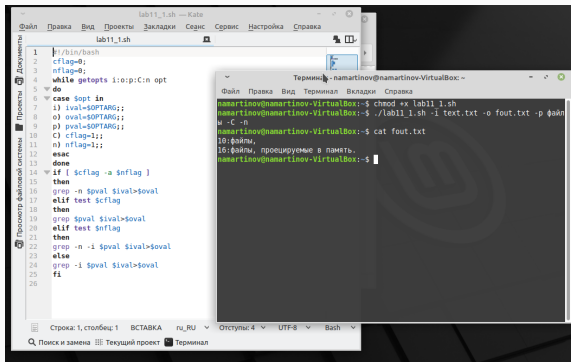
1 Выполнить 4 задания

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 screenshot displays a KVM virtual machine window titled 'lab11_1.sh - KVM'. The main window shows a shell script named 'lab11_1.sh' with the following content:

```
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
26
```

Overlaid on the script editor is a terminal window titled 'Терминал - namartinov@namartinov-VirtualBox: ~'. The terminal shows the execution of the script with the following commands and output:

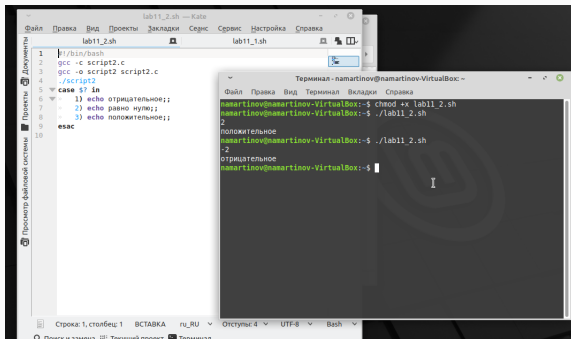
```
namartinov@namartinov-VirtualBox:~$ chmod +x lab11_1.sh
namartinov@namartinov-VirtualBox:~$ ./lab11_1.sh -i text.txt -o fout.txt -p файл
-C -n
namartinov@namartinov-VirtualBox:~$ cat fout.txt
10:файл,
10:файл, просецируемые в память.
namartinov@namartinov-VirtualBox:~$
```

The terminal output shows the script's execution with flags -i, -o, and -p, and the resulting output file 'fout.txt' containing the search results.

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 background window is a code editor titled 'lab11_2.sh' containing a shell script. The script starts with a shebang, compiles a C program 'script2.c' into 'script2', and then runs it. A 'case' statement follows, which prints 'отрицательное;' for negative numbers, 'равно нулю;' for zero, and 'положительное;' for positive numbers. The foreground window is a terminal titled 'Терминал - namartinov@namartinov-VirtualBox: ~'. It shows the user running 'chmod +x lab11_2.sh' and then './lab11_2.sh'. The terminal output shows the script's results: 'положительное' followed by '2', and 'отрицательное' followed by '2'.

```
lab11_2.sh
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 - namartinov@namartinov-VirtualBox: ~
namartinov@namartinov-VirtualBox:~$ chmod +x lab11_2.sh
namartinov@namartinov-VirtualBox:~$ ./lab11_2.sh
2
положительное
namartinov@namartinov-VirtualBox:~$ ./lab11_2.sh
2
отрицательное
namartinov@namartinov-VirtualBox:~$
```

Figure 2: Задание 2

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

Выполнение работы

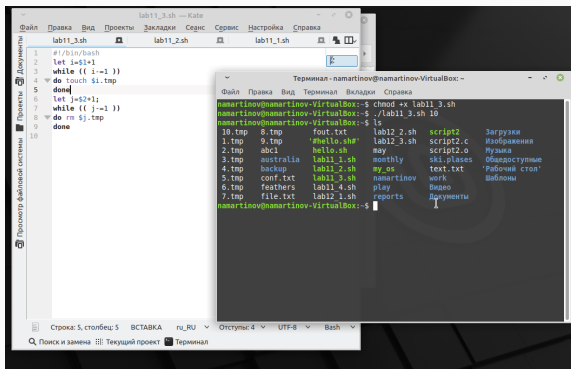


Figure 3: Задание 3

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

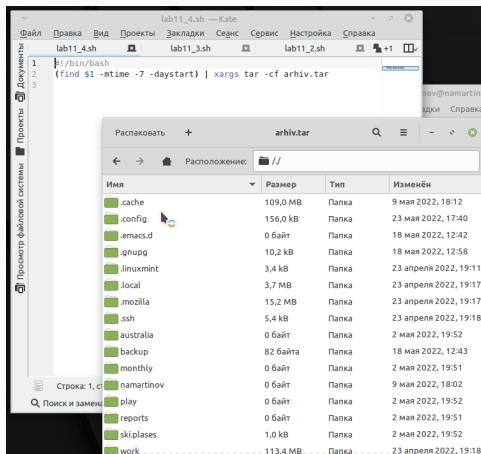


Figure 4: Задание 4

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.