

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Эрик Кириллович Станиловский НПМбд-02-20¹

27 мая, 2021, 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

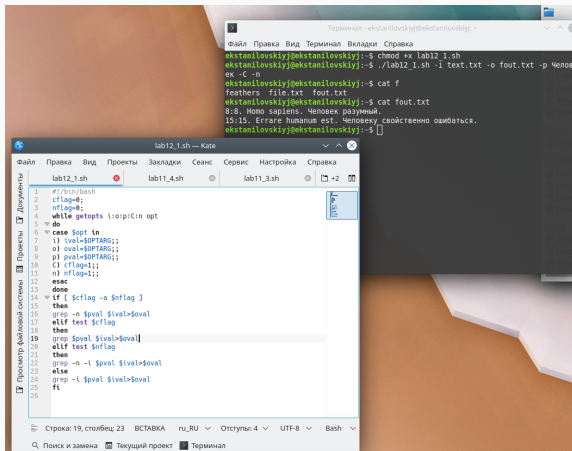
1 Выполнить 4 задания

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 terminal window and a text editor window. The terminal window, titled "Терминал - ekstanilovskiy@ekstanilovskiy: ~", displays the execution of a shell script named lab12_1.sh. The script's output includes the command "chmod +x lab12_1.sh", the execution of the script itself, and the creation of a file named "fout.txt". The output also shows the contents of "fout.txt", which is a cat command, and the contents of "cat fout.txt", which is a file named "fathers file.txt". The output ends with the text "8:8. Homo sapiens. Человек разумный. 15:15. Errare humanum est. Человеку свойственно ошибаться." and a prompt for the user to enter a command.

```
ekstanilovskiy@ekstanilovskiy:~$ chmod +x lab12_1.sh
ekstanilovskiy@ekstanilovskiy:~$ ./lab12_1.sh -i text.txt -o fout.txt -p Ченов
ек -C -n
ekstanilovskiy@ekstanilovskiy:~$ cat f
fathers file.txt fout.txt
ekstanilovskiy@ekstanilovskiy:~$ cat fout.txt
8:8. Homo sapiens. Человек разумный.
15:15. Errare humanum est. Человеку свойственно ошибаться.
ekstanilovskiy@ekstanilovskiy:~$
```

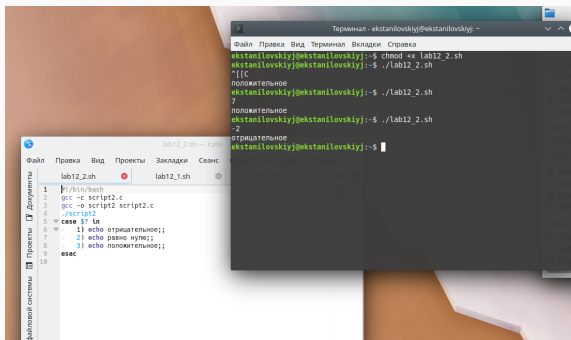
The text editor window, titled "lab12_1.sh — Kate", shows the source code of the script lab12_1.sh. The script is a Bash script that takes three arguments: a file name, an output file name, and a password. It uses a case statement to handle different options and a while loop to read the input file. It then uses the cat command to display the contents of the input file and the output file. The script ends with a prompt for the user to enter a command.

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:op:C:n opt
5 do
6     case $opt in
7         i) sval=$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 $sval>$oval
17    elif test $cflag
18    then
19        grep $pval $sval>$oval
20    elif test $nflag
21    then
22        grep -n -l $pval $sval>$oval
23    else
24        grep -t $pval $sval>$oval
25    fi
26
```

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 image shows a Linux environment with a terminal window and a code editor. The terminal window, titled "Терминал - ekstanilovskiy@ekstanilovskiy: ~", displays the following commands and output:

```
ekstanilovskiy@ekstanilovskiy:~$ chmod +x lab12_2.sh
ekstanilovskiy@ekstanilovskiy:~$ ./lab12_2.sh
+[[c
положительное
ekstanilovskiy@ekstanilovskiy:~$ ./lab12_2.sh
7
положительное
ekstanilovskiy@ekstanilovskiy:~$ ./lab12_2.sh
-2
отрицательное
ekstanilovskiy@ekstanilovskiy:~$
```

The code editor, titled "lab12_2.sh — Kate", shows the content of the script:

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

Figure 2: Задание 2

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

Выполнение работы

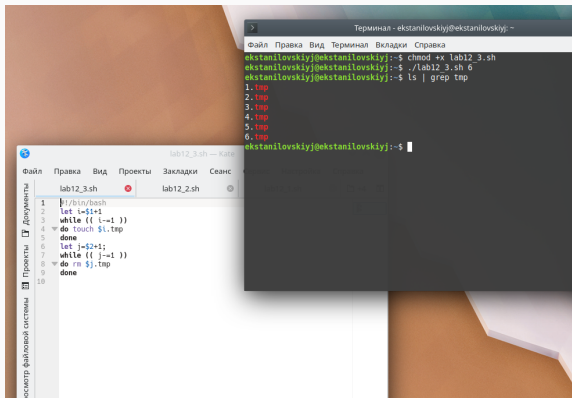


Figure 3: Задание 3

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

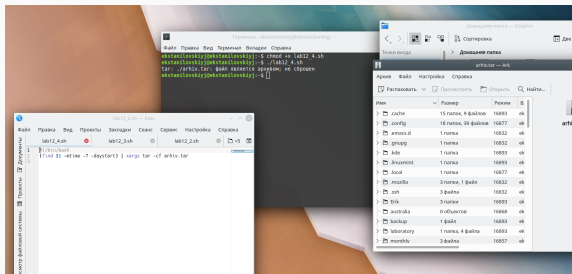


Figure 4: Задание 4

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.