

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Тукаев Тимур Ильшатovich¹

17 апреля, 2024, 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

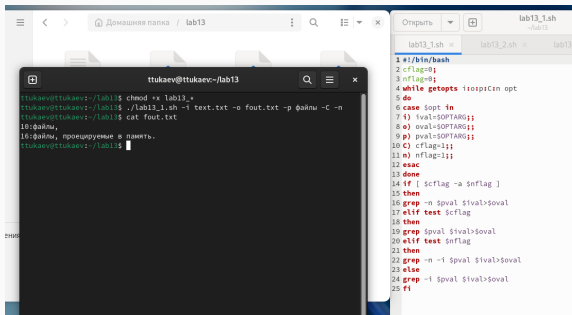
1 Выполнить 4 задания

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 code editor side-by-side. The terminal window, titled 'ttukaev@ttukaev:~/lab13', shows the execution of a shell script 'lab13.sh'. The user runs 'chmod +x lab13.sh', then './lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n'. The script outputs '10:файлы,' and '16:файлы, проецируемые в память.' The code editor, titled 'lab13_1.sh', shows the script's content, which includes a loop that iterates over command-line arguments and uses 'grep' to find patterns.

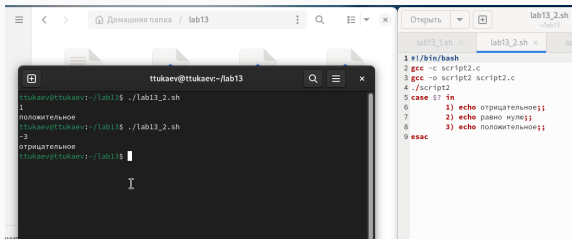
```
ttukaev@ttukaev:~/lab13$ chmod +x lab13.sh
ttukaev@ttukaev:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
ttukaev@ttukaev:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
ttukaev@ttukaev:~/lab13$
```

```
1 #!/bin/bash
2 cflag=0
3 nflag=0
4 while getopts i:opsC: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 -i $pval $sval>$oval
23 else
24 grep -i $pval $sval>$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 desktop environment with a terminal window and a file editor. The terminal window, titled 'ttukaev@ttukaev:~/lab13', displays the execution of two shell scripts. The first script, 'lab13_1.sh', prints 'положительное' (positive) and returns exit code 1. The second script, 'lab13_2.sh', prints 'отрицательное' (negative) and returns exit code -3. The file editor, titled 'lab13_2.sh', shows the source code of the second script, which uses a case statement to check if a variable is negative, zero, or positive, and prints the corresponding message.

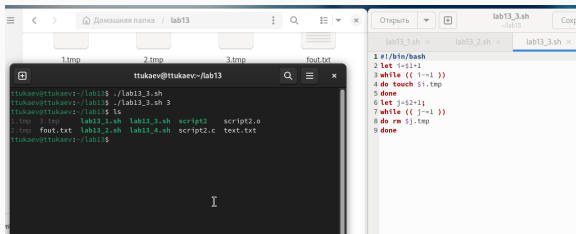
```
ttukaev@ttukaev:~/lab13$ ./lab13_1.sh
1
положительное
ttukaev@ttukaev:~/lab13$ ./lab13_2.sh
-3
отрицательное
ttukaev@ttukaev:~/lab13$
```

```
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
```

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 explorer. The terminal window, titled 'ttukaev@ttukaev:~/lab13', displays the following commands and output:

```
ttukaev@ttukaev:~/lab13$ ./lab13_3.sh
ttukaev@ttukaev:~/lab13$ ./lab13_3.sh 3
ttukaev@ttukaev:~/lab13$ ls
1.tmp  3.tmp  lab13_1.sh  lab13_3.sh  script2  script2.o
2.tmp  fout.txt lab13_2.sh  lab13_4.sh  script2.c  text.txt
ttukaev@ttukaev:~/lab13$
```

The file explorer shows the directory structure of the lab13 folder, including files 1.tmp, 2.tmp, 3.tmp, and fout.txt. The terminal window also shows the execution of the script with the argument 3, resulting in the creation of files 1.tmp, 2.tmp, 3.tmp, and fout.txt, and the execution of the script 3.

```
1 #!/bin/bash
2 let i=i+1
3 while (( i--1 ))
4 do touch $i.tmp
5 done
6 let j=i2+1
7 while (( j--1 ))
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

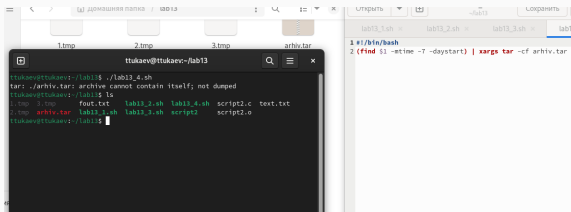


Рис. 4: Задание 4

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.