

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Исаева Зарина¹

20 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

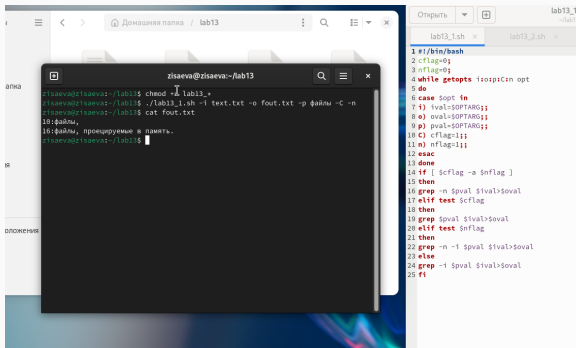
1 Выполнить 4 задания

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; `-p 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 file editor. The terminal window, titled 'zisaeva@zisaeva:~/lab13', displays the following commands and output:

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

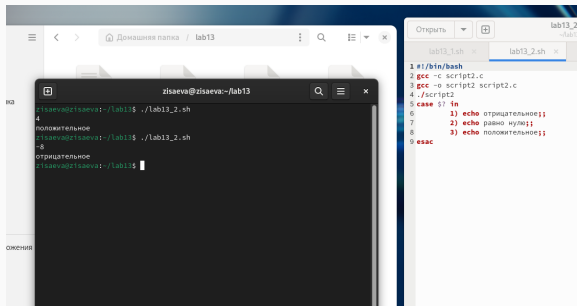
The file editor, titled 'lab13_1', shows the following shell script code:

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:osp: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
```

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 screenshot displays a Linux desktop environment. In the background, a file manager window shows the directory structure of a folder named 'lab13'. In the foreground, a terminal window titled 'zisaeva@zisaeva:~/lab13' is open. The terminal shows the execution of a script named 'lab13_2.sh' which prompts the user for input. The user has entered 'положительное' (positive), '-8', and 'отрицательное' (negative). To the right of the terminal, a code editor window titled 'lab13_2.sh' displays the script's content, which uses a 'case' statement to check if the input is negative, zero, or positive.

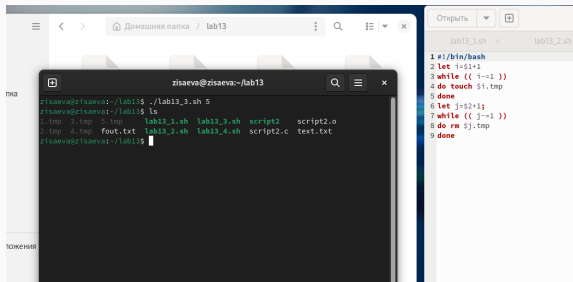
```
zisaeva@zisaeva:~/lab13$ ./lab13_2.sh
4
положительное
zisaeva@zisaeva:~/lab13$ ./lab13_2.sh
-8
отрицательное
zisaeva@zisaeva:~/lab13$
```

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

Рис. 2: Задание 2

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

Выполнение работы



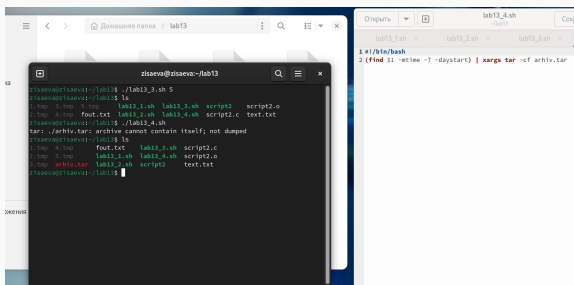
The image shows a terminal window and a file explorer. The terminal window, titled 'zisaeva@zisaeva:~/lab13', displays the execution of a script 'lab13_3.sh' with 5 arguments. The output lists several files and directories: '1.tmp', '3.tmp', '5.tmp', 'lab13_1.sh', 'lab13_3.sh', 'script2', 'script2.o', '1.tmp', '4.tmp', 'fout.txt', 'lab13_2.sh', 'lab13_4.sh', 'script2.c', and 'text.txt'. The file explorer, titled 'Домашняя папка / lab13', shows the contents of the 'lab13' directory, including 'lab13_1.sh' and 'lab13_2.sh'.

```
zisaeva@zisaeva:~/lab13$ ./lab13_3.sh 5
zisaeva@zisaeva:~/lab13$ ls
1.tmp  3.tmp  5.tmp  lab13_1.sh  lab13_3.sh  script2  script2.o
1.tmp  4.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
zisaeva@zisaeva:~/lab13$
```

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 image shows a terminal window and a file manager. The terminal window, titled 'zisaeva@zisaeva:~/lab13', displays the following commands and output:

```
zisaeva@zisaeva:~/lab13$ ./lab13_3.sh 5
zisaeva@zisaeva:~/lab13$ ls
-rtmp 3.tmp 4.tmp lab13_1.sh lab13_3.sh script2 script2.o
-rtmp 4.tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
zisaeva@zisaeva:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
zisaeva@zisaeva:~/lab13$ ls
-rtmp 4.tmp fout.txt lab13_3.sh script2.c
-rtmp 5.tmp lab13_1.sh lab13_4.sh script2.o
-rtmp arhiv.tar lab13_2.sh script2 text.txt
zisaeva@zisaeva:~/lab13$
```

The file manager window, titled 'lab13_4.sh', shows the following commands and output:

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

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.