

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Кенан Гашимов¹

19 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

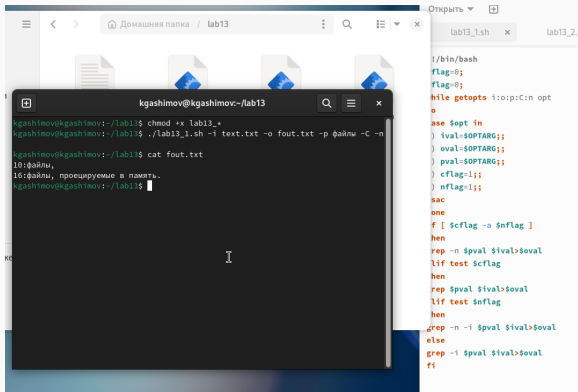
1 Выполнить 4 задания

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



```
kgashimov@kgashimov:~/lab13$ chmod +x lab13_*
kgashimov@kgashimov:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлн -C -n

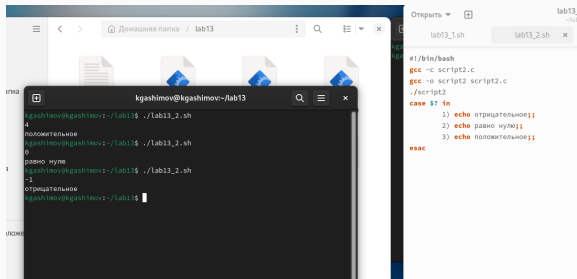
kgashimov@kgashimov:~/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
kgashimov@kgashimov:~/lab13$
```

```
#!/bin/bash
flag=0;
flag=0;
while getopts i:o:p:C:n opt
do
case $opt in
) ival=$OPTARG;;
) oval=$OPTARG;;
) pval=$OPTARG;;
) cflag=1;;
) nflag=1;;
) sac
one
f [ $cflag -a $nflag ]
hen
rep -n $pval $ival>$oval
lif test $cflag
hen
rep $pval $ival>$oval
lif test $nflag
hen
rep -n -i $pval $ival>$oval
else
rep -i $pval $ival>$oval
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 contents of a directory named 'lab13'. It contains several files, including 'lab13_1.sh' and 'lab13_2.sh'. In the foreground, a terminal window is open, showing the execution of a script. The terminal prompt is 'kgashimov@kgashimov:~/lab13'. The user has entered './lab13_2.sh', and the script has executed several lines of code, including 'echo 4', 'echo положительное', 'echo 0', 'echo равно нулю', 'echo -1', and 'echo отрицательное'. To the right of the terminal, a code editor window is visible, showing the source code of the script. The code includes a shebang line '#!/bin/bash', compilation commands 'gcc -c script2.c' and 'gcc -o script2 script2.c', and a case statement that prints different messages based on the input value.

```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
    1) echo отрицательное;;
    2) echo равно нулю;;
    3) echo положительное;;
esac
```

Рис. 2: Задание 2

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

Выполнение работы

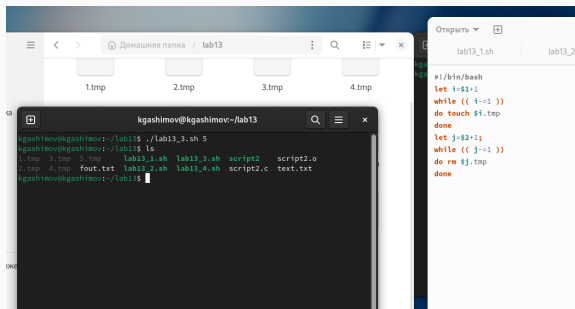


Рис. 3: Задание 3

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

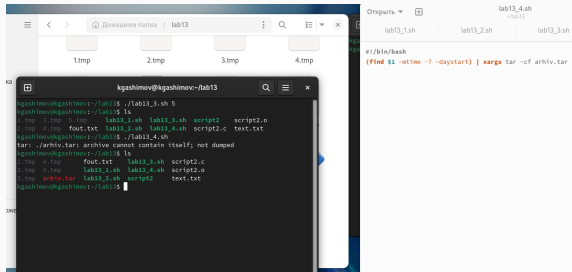


Рис. 4: Задание 4

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.