

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Тумуреева Галина Аркадьевна¹

23 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

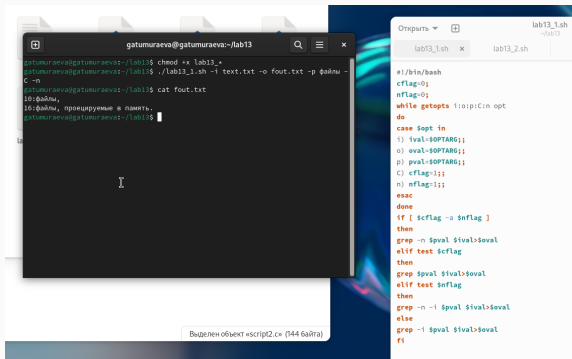
1 Выполнить 4 задания

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 code editor. The terminal window, titled 'gatumuraeva@gatumuraeva:~/lab13', displays the following commands and output:

```
gatumuraeva@gatumuraeva:~/lab13$ cheod *x lab13_*
gatumuraeva@gatumuraeva:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p 0a0mu -
C -n
gatumuraeva@gatumuraeva:~/lab13$ cat fout.txt
10:0a0mu,
10:0a0mu, процируемые в памяти.
gatumuraeva@gatumuraeva:~/lab13$
```

The code editor, titled 'lab13_1.sh', shows the following script:

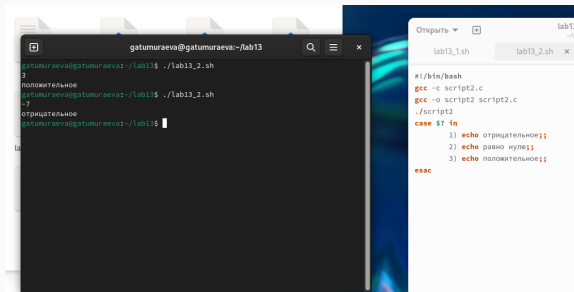
```
#!/bin/bash
cflag=0;
nflag=0;
while getopts i:op:Cin opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival$oval
elif test $cflag
then
grep $pval $ival$oval
elif test $nflag
then
grep -n -i $pval $ival$oval
else
grep -i $pval $ival$oval
fi
```

At the bottom of the terminal window, a message states: 'Выделен объект «script2.c» (144 байта)'.

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 terminal window and a code editor. The terminal window, titled 'gatumuraeva@gatumuraeva:~/lab13', displays the execution of a script 'lab13_2.sh'. The script takes an argument '3' and prints 'положительное' (positive). The user then runs the script with argument '-7', and it prints 'отрицательное' (negative). The code editor, titled 'lab13', shows the source code of 'lab13_2.sh'. The script is a Bash script that compiles 'script2.c' with 'gcc', runs it, and uses a 'case' statement to print 'отрицательное' for negative numbers, 'равно нулю' (equal to zero) for zero, and 'положительное' for positive numbers.

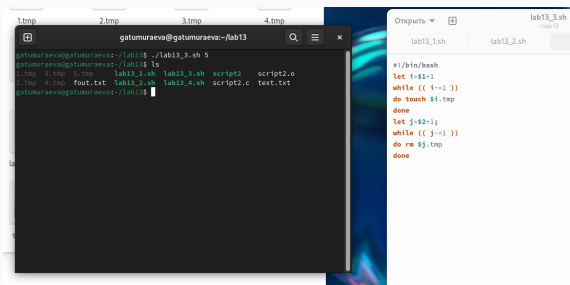
```
gatumuraeva@gatumuraeva:~/lab13$ ./lab13_2.sh
3
положительное
gatumuraeva@gatumuraeva:~/lab13$ ./lab13_2.sh
-7
отрицательное
gatumuraeva@gatumuraeva:~/lab13$
```

```
#!/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



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

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

The script file, titled 'lab13_3.sh', contains the following code:

```
#!/bin/bash
let i=$1+1
while (( i<=1 ))
do touch $i.tmp
done
let j=$2+1;
while (( j<=1 ))
do rm $j.tmp
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

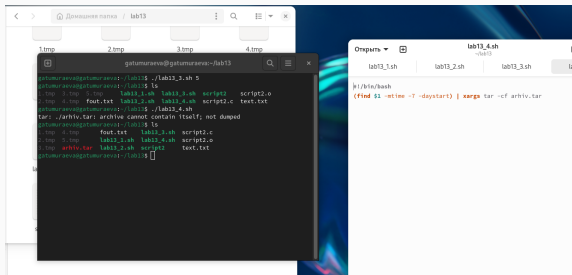


Рис. 4: Задание 4

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.