

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Ларина Наталья¹

24 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

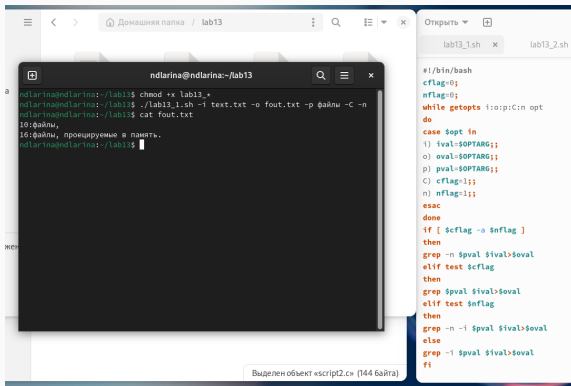
1 Выполнить 4 задания

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 'ndlarina@ndlarina:~/lab13', shows the execution of a shell script 'lab13_1.sh'. The script takes 'text.txt' as input and produces 'fout.txt' as output. The output of 'cat fout.txt' is displayed, showing the contents of 'text.txt' and the memory address '0x00000000'. The code editor, titled 'lab13_1.sh', shows the source code of the script. The script is a shell script that takes a file as input and processes it. It uses 'while' and 'case' statements to handle different options. The script is written in a shell script language.

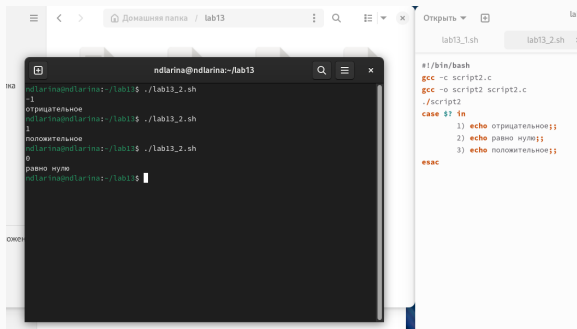
```
ndlarina@ndlarina:~/lab13
ndlarina@ndlarina:~/lab13$ chmod +x lab13_1.sh
ndlarina@ndlarina:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p 0x00000000 -C -n
ndlarina@ndlarina:~/lab13$ cat fout.txt
0x00000000,
0x00000000, процессорные в памяти.
ndlarina@ndlarina:~/lab13$
```

```
#!/bin/bash
cflag=0;
nflag=0;
while getopts i:o:p:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -o $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
```

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 'ndlarina@ndlarina:~/lab13', shows the execution of a script named 'lab13_2.sh'. The script takes an argument '1' and prints 'отрицательное' (negative). The code editor, titled 'lab13_2.sh', shows the source code of the script, which uses a case statement to print different messages based on the input argument.

```
ndlarina@ndlarina:~/lab13$ ./lab13_2.sh
1
отрицательное
ndlarina@ndlarina:~/lab13$ ./lab13_2.sh
1
положительное
ndlarina@ndlarina:~/lab13$ ./lab13_2.sh
0
равно нулю
ndlarina@ndlarina:~/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

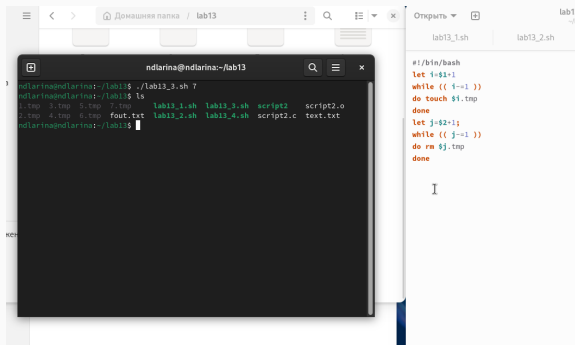
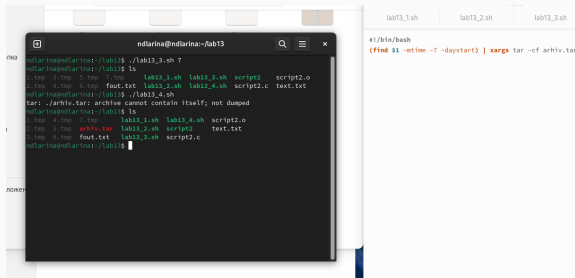


Рис. 3: Задание 3

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

```
ndlarina@ndlarina:~/lab13 ./lab13_3.sh 7
ndlarina@ndlarina:~/lab13$ ls
1.tmp  4.tmp  6.tmp  lab13_1.sh  lab13_2.sh  script2  script2.o
2.tmp  4.tmp  6.tmp  fout.txt  lab13_3.sh  lab13_4.sh  script2.c  text.txt
ndlarina@ndlarina:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
ndlarina@ndlarina:~/lab13$ ls
1.tmp  4.tmp  7.tmp  lab13_1.sh  lab13_4.sh  script2.o
2.tmp  5.tmp  arhiv.tar  lab13_2.sh  script2  text.txt
3.tmp  6.tmp  fout.txt  lab13_3.sh  script2.c
ndlarina@ndlarina:~/lab13$
```

The file explorer shows a directory named 'lab13' with three subdirectories: 'lab13_1.sh', 'lab13_2.sh', and 'lab13_3.sh'. The terminal window is open in the 'lab13' directory.

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.