

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Ридой Хайатуннаби¹

26 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

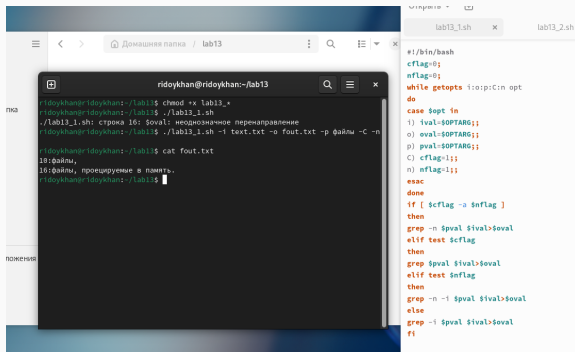
1 Выполнить 4 задания

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. The terminal window, titled 'ridoykhan@ridoykhan:~/lab13', displays the execution of a script 'lab13_1.sh'. The script takes a file path as an argument and outputs its contents. The code editor, titled 'lab13_1.sh', shows the script's source code, which is a shell script that takes a file path as an argument and outputs its contents.

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

```
#!/bin/bash
cflag=0;
nflag=0;
while getopts i:op: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 -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
```

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

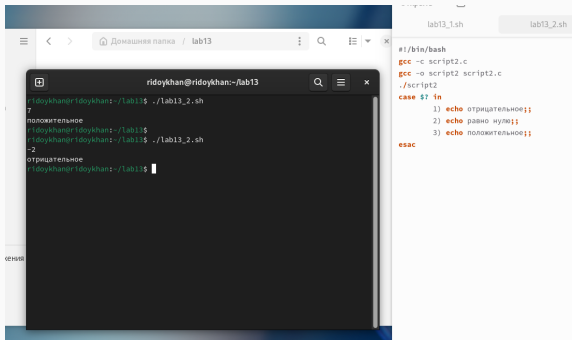
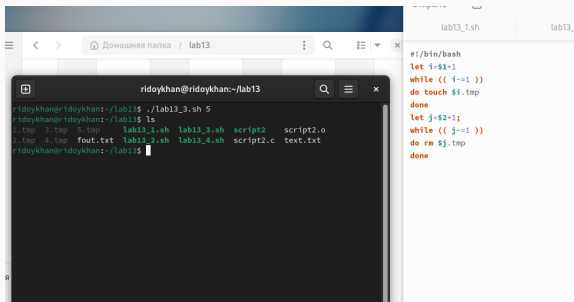


Рис. 2: Задание 2

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 'ridoykhan@ridoykhan:~/lab13', displays the following commands and output:

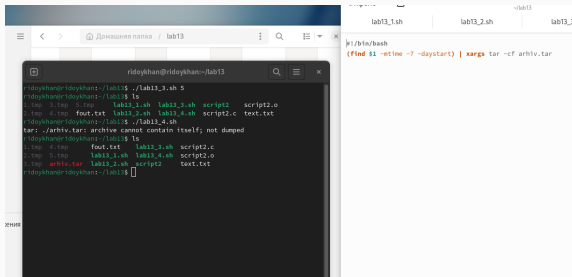
```
ridoykhan@ridoykhan:~/lab13$ ./lab13_3.sh 5
ridoykhan@ridoykhan:~/lab13$ ls
1.tmp 3.tmp 5.tmp lab13_1.sh lab13_3.sh script2 script2.o
2.tmp 4.tmp fout.txt lab13_2.sh lab13_4.sh script2.c text.txt
ridoykhan@ridoykhan:~/lab13$
```

The file explorer shows the contents of the 'lab13' directory, including files like 'lab13_1.sh', 'lab13_2.sh', 'lab13_3.sh', 'lab13_4.sh', 'script2', 'script2.c', 'script2.o', 'text.txt', and 'fout.txt'.

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



```
ridoykhan@ridoykhan:~/lab13$ ./lab13_3.sh 5
ridoykhan@ridoykhan:~/lab13$ ls
5.tap 3.tap 5.tap lab13_1.sh lab13_3.sh script2 script2.o
4.tap 4.tap foot.txt lab13_2.sh lab13_4.sh script2.c text.txt
ridoykhan@ridoykhan:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
ridoykhan@ridoykhan:~/lab13$ ls
5.tap 4.tap foot.txt lab13_1.sh lab13_3.sh script2.c
4.tap 5.tap lab13_2.sh lab13_4.sh script2.o
5.tap arhiv.tar lab13_2.sh script2 text.txt
ridoykhan@ridoykhan:~/lab13$
```

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.