

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Сардор Турсунбоев¹

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

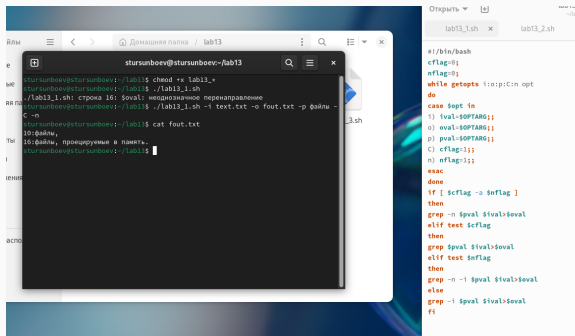
1 Выполнить 4 задания

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

```
stursunboev@stursunboev:~/lab13$ chmod +x lab13_*
stursunboev@stursunboev:~/lab13$ ./lab13_1.sh
./lab13_1.sh: строка 16: $oval: неоднозначное перенаправление
stursunboev@stursunboev:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p $valnu -C -n
stursunboev@stursunboev:~/lab13$ cat fout.txt
10:файлы
16:файлы, пропущенные в памяти.
stursunboev@stursunboev:~/lab13$
```

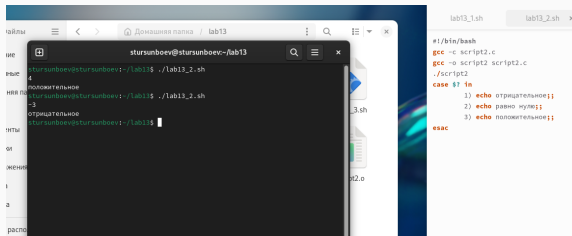
The script file, 'lab13_2.sh', contains the following code:

```
#!/bin/bash
cflag=0
nflag=0
while getopts i:o:p:C:n opt
do
case $opt in
1) sval=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $sval $oval
elif test $cflag
then
grep $pval $sval $oval
elif test $nflag
then
grep -n -i $pval $sval $oval
else
grep -i $pval $sval $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 'stursunboev@stursunboev:~/lab13', displays the execution of a script 'lab13_2.sh'. The script's output is as follows:

```
stursunboev@stursunboev:~/lab13$ ./lab13_2.sh
4
положительное
stursunboev@stursunboev:~/lab13$ ./lab13_2.sh
-3
отрицательное
stursunboev@stursunboev:~/lab13$
```

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

```
#!/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

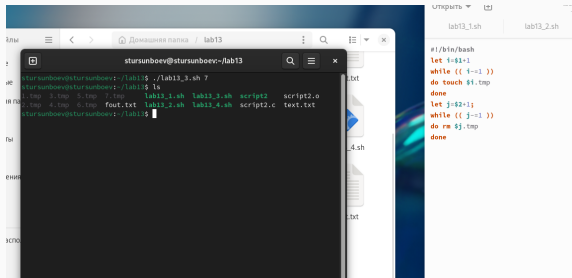
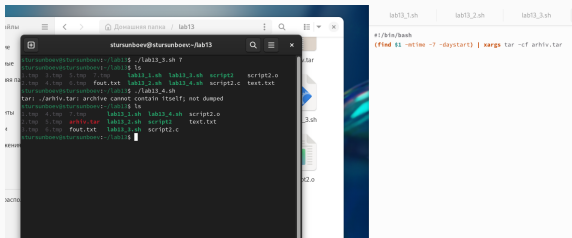


Рис. 3: Задание 3

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 manager interface. The terminal window, titled 'sturunboev@sturunboev:~/lab13', displays the following commands and output:

```
sturunboev@sturunboev:~/lab13$ ./lab13_3.sh 7
sturunboev@sturunboev:~/lab13$ ls
2.1.tar  2.1.tar  5.1.tar  7.1.tar  lab13_1.sh  lab13_3.sh  script2
2.1.tar  4.1.tar  6.1.tar  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
sturunboev@sturunboev:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
sturunboev@sturunboev:~/lab13$ ls
2.1.tar  4.1.tar  7.1.tar  lab13_1.sh  lab13_4.sh  script2.o
2.1.tar  5.1.tar  arhiv.tar  lab13_2.sh  script2    text.txt
2.1.tar  6.1.tar  fout.txt  lab13_3.sh  script2.c
sturunboev@sturunboev:~/lab13$
```

The file manager interface shows three tabs: 'lab13_1.sh', 'lab13_2.sh', and 'lab13_3.sh'. The active tab 'lab13_3.sh' contains the following code:

```
#!/bin/bash
(find $1 -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.