

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Илья Колонтырский¹

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

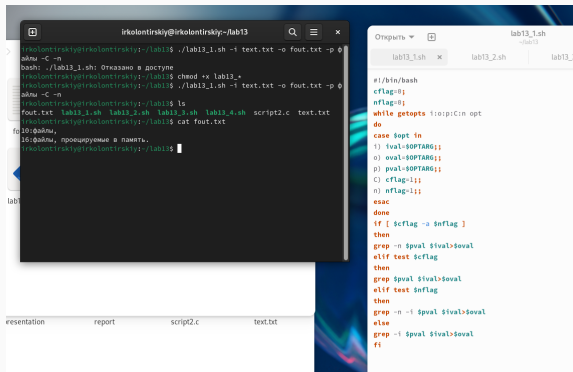
1 Выполнить 4 задания

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 'irkolontirskiy@irkolontirskiy:~/lab13', shows the execution of a shell script 'lab13_1.sh'. The script takes 'text.txt' as input and produces 'fout.txt'. The terminal output shows the script being run, the current directory being checked, and the script being executed. The code editor, titled 'lab13_1.sh', shows the script's source code, which is a shell script that takes command-line arguments and processes them.

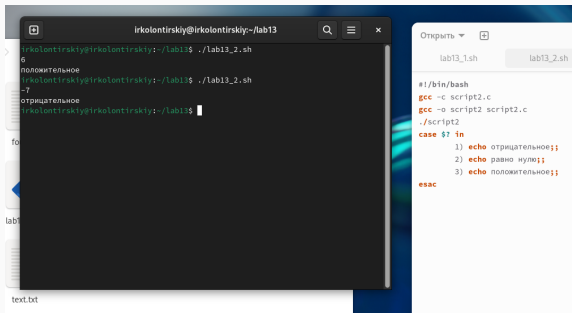
```
irkolontirskiy@irkolontirskiy:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p 0
bash: ./lab13_1.sh: Отказано в доступе
irkolontirskiy@irkolontirskiy:~/lab13$ cheed +x lab13_1
irkolontirskiy@irkolontirskiy:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p 0
bash: ./lab13_1.sh: Отказано в доступе
irkolontirskiy@irkolontirskiy:~/lab13$ ls
lab13_1.sh lab13_2.sh lab13_3.sh lab13_4.sh script2.c text.txt
irkolontirskiy@irkolontirskiy:~/lab13$ cat fout.txt
10:файлы, проецируемые в память.
irkolontirskiy@irkolontirskiy:~/lab13$
```

```
#!/bin/bash
cflag=0
nflag=0
while getopts i:oi: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 'irkolontirskiy@irkolontirskiy:~/lab13', displays the execution of a script 'lab13_2.sh'. The script outputs '6' and 'положительное' (positive), followed by '7' and 'отрицательное' (negative). The code editor, titled 'Открыть', 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 values, 'равно нулю' (equal to zero) for zero, and 'положительное' for positive values.

```
irkolontirskiy@irkolontirskiy:~/lab13$ ./lab13_2.sh
6
положительное
irkolontirskiy@irkolontirskiy:~/lab13$ ./lab13_2.sh
7
отрицательное
irkolontirskiy@irkolontirskiy:~/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

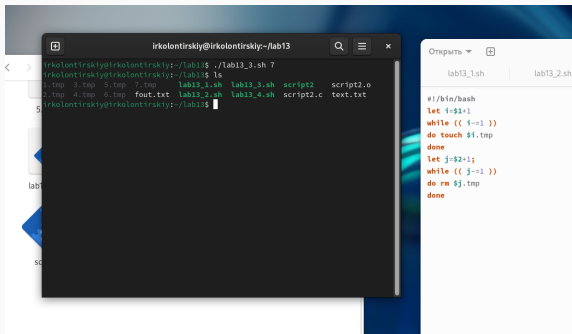


Рис. 3: Задание 3

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

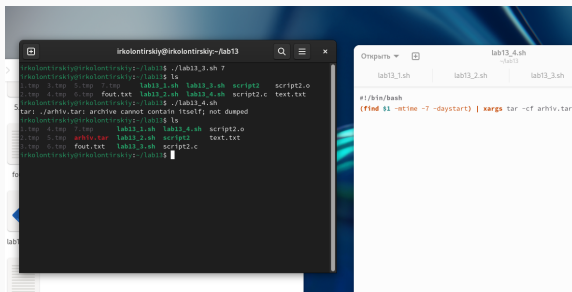


Рис. 4: Задание 4

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.