

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Хамди Мохаммад¹

27 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

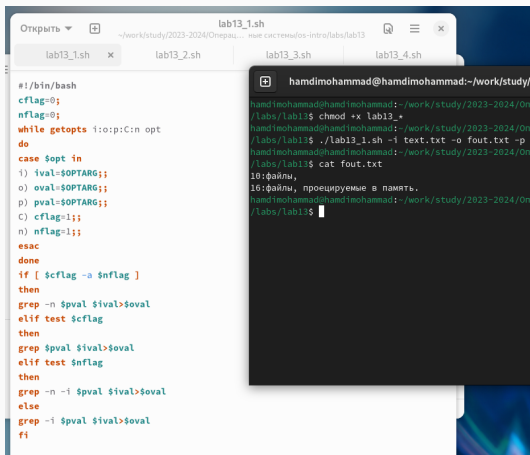
1 Выполнить 4 задания

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



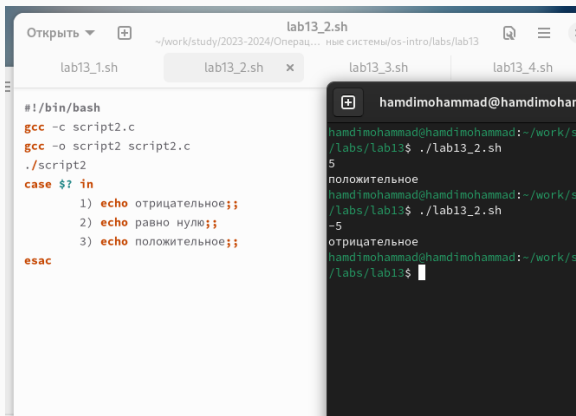
```
#!/bin/bash
cflag=0;
nflag=0;
while getopts i:o:p:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i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

```
hamdimohammad@hamdimohammad:~/work/study/
hamdimohammad@hamdimohammad:~/work/study/2023-2024/On
/labs/lab13$ chmod +x lab13_*
hamdimohammad@hamdimohammad:~/work/study/2023-2024/On
/labs/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p
hamdimohammad@hamdimohammad:~/work/study/2023-2024/On
/labs/lab13$ cat fout.txt
10:Файлы,
16:Файлы, проецируемые в память.
hamdimohammad@hamdimohammad:~/work/study/2023-2024/On
/labs/lab13$
```

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 code editor with four tabs: lab13_1.sh, lab13_2.sh (active), lab13_3.sh, and lab13_4.sh. The active tab displays a shell script named lab13_2.sh. The script starts with a shebang, compiles script2.c, and then uses a case statement to check if a number is positive, zero, or negative. An inset window shows the terminal output of running the script with the number 5, which correctly identifies it as 'положительное' (positive).

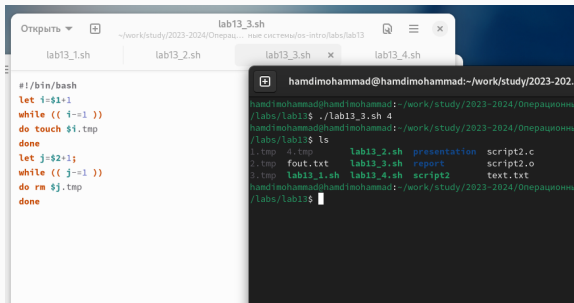
```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
    1) echo отрицательное;;
    2) echo равно нулю;;
    3) echo положительное;;
esac
```

```
hamdimohammad@hamdimohammad:~/work/s
/labs/lab13$ ./lab13_2.sh
5
положительное
hamdimohammad@hamdimohammad:~/work/s
/labs/lab13$ ./lab13_2.sh
-5
отрицательное
hamdimohammad@hamdimohammad:~/work/s
/labs/lab13$
```

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 with a dark theme. The top bar indicates the current directory is `~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13`. The terminal has several tabs open: `lab13_1.sh`, `lab13_2.sh`, `lab13_3.sh` (active), and `lab13_4.sh`. The active tab displays the following shell script:

```
#!/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
```

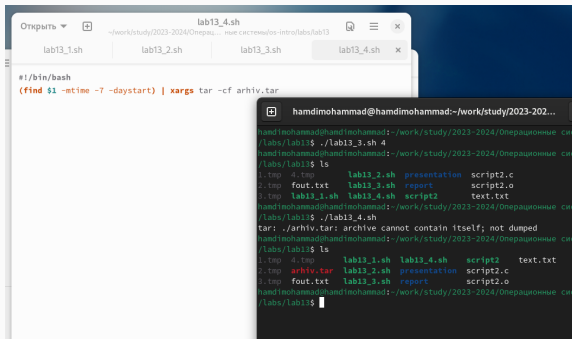
The terminal shows the execution of the script with the following output:

```
handimohammad@handimohammad: ~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ./lab13_3.sh 4
handimohammad@handimohammad: ~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ls
1.tmp  4.tmp      lab13_2.sh  presentation  script2.c
2.tmp  fout.txt   lab13_3.sh  report        script2.o
3.tmp  lab13_1.sh lab13_4.sh  script2       text.txt
handimohammad@handimohammad: ~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$
```

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



```
Открыть ▾ + lab13_4.sh
~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13
lab13_1.sh lab13_2.sh lab13_3.sh lab13_4.sh x

#!/bin/bash
(find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar

hamdimohammad@hamdimohammad:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ./lab13_3.sh 4
hamdimohammad@hamdimohammad:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ls
1.tmp 4.tmp lab13_2.sh presentation script2.c
2.tmp fout.txt lab13_3.sh report script2.o
3.tmp lab13_1.sh lab13_4.sh script2 text.txt
hamdimohammad@hamdimohammad:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
hamdimohammad@hamdimohammad:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ls
1.tmp 4.tmp lab13_1.sh lab13_4.sh script2 text.txt
2.tmp arhiv.tar lab13_2.sh presentation script2.c
3.tmp fout.txt lab13_3.sh report script2.o
hamdimohammad@hamdimohammad:~/work/study/2023-2024/Операционные системы/os-intro/labs/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.