

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Евгений Шехавцов¹

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

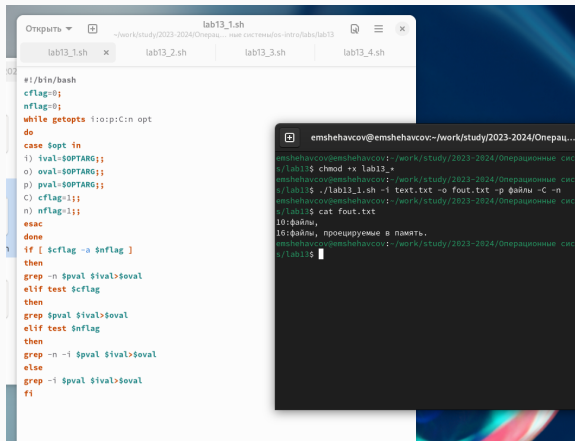
1 Выполнить 4 задания

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



The image shows a terminal window with a light blue header bar. The title bar reads "lab13_1.sh" and the path is "~work/study/2023-2024/Операц... ние системы/os-infra/labs/lab13". There are four tabs: "lab13_1.sh", "lab13_2.sh", "lab13_3.sh", and "lab13_4.sh". The terminal content is a shell script for a lab exercise. It uses a while loop with getopt to parse options -f, -o, -p, and -c. It then uses grep to check if the provided arguments match the expected values. The script is executed in a separate terminal window, where the user runs "chmod +x lab13_1.sh", then "s/lab1: ./lab13_1.sh -f text.txt -o fout.txt -p файлы -C -n", and finally "cat fout.txt". The output of the script is displayed in the terminal.

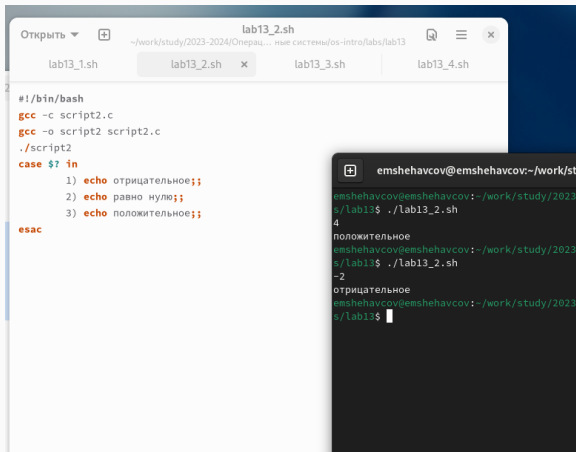
```
#!/bin/bash
cflag=0;
nflag=0;
while getopts f:op:c:n opt
do
case $opt in
f) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i $pval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

```
emshehacov@emshehacov:~/work/study/2023-2024/Операц...
s/lab1:~$ chmod +x lab13_1.sh
emshehacov@emshehacov:~/work/study/2023-2024/Операц...
s/lab1:~$ ./lab13_1.sh -f text.txt -o fout.txt -p файлы -C -n
emshehacov@emshehacov:~/work/study/2023-2024/Операц...
s/lab1:~$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
emshehacov@emshehacov:~/work/study/2023-2024/Операц...
s/lab1:~$
```

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 window titled 'lab13_2.sh' with tabs for 'lab13_1.sh', 'lab13_2.sh', 'lab13_3.sh', and 'lab13_4.sh'. The code in 'lab13_2.sh' is as follows:

```
#!/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
```

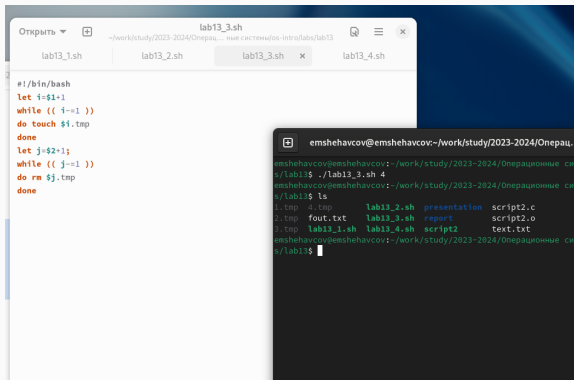
Below the code editor is a terminal window showing the execution of the script. The user runs './lab13_2.sh' and the output is '4'.

```
emshehavgov@emshehavgov:~/work/st
emshehavgov@emshehavgov: ~/work/study/2023
s/lab13$ ./lab13_2.sh
4
положительное
emshehavgov@emshehavgov: ~/work/study/2023
s/lab13$ ./lab13_2.sh
-2
отрицательное
emshehavgov@emshehavgov: ~/work/study/2023
s/lab13$
```

Рис. 2: Задание 2

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N

Выполнение работы



The image displays two terminal windows. The left window, titled 'lab13_3.sh', shows the source code of a shell script. The script uses nested while loops to create and touch files in a directory named 'tmp'. The right window, titled 'emshehacov@emshehacov:~/work/study/2023-2024/Операц..', shows the execution of the script. It displays the command './lab13_3.sh 4' and the output of the 'ls' command, which lists the files created in the 'tmp' directory.

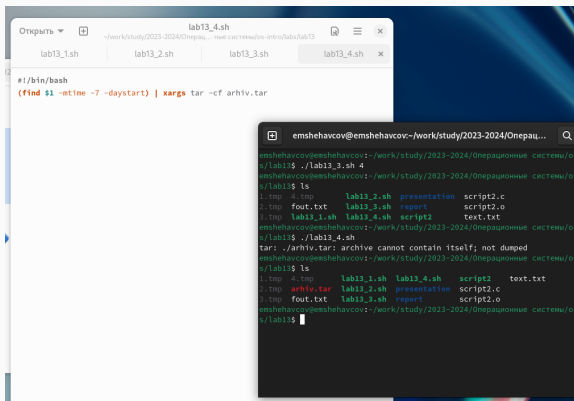
```
#!/bin/bash
let i=$i+1
while (( i=1 ))
do touch $i.tmp
done
let j=$j+1;
while (( j=1 ))
do rm $j.tmp
done
```

```
emshehacov@emshehacov:~/work/study/2023-2024/Операц..
s/lab13$ ./lab13_3.sh 4
emshehacov@emshehacov:~/work/study/2023-2024/Операцонные см
s/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
emshehacov@emshehacov:~/work/study/2023-2024/Операцонные см
s/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



The image shows a terminal window with a dark background and light-colored text. The window title is "lab13_4.sh". The prompt is "#!/bin/bash". The command being executed is "(find \$1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar". The output of the command is a list of files and directories: "1.tmp 4.tmp lab13_2.sh presentation script2.c", "2.tmp fout.txt lab13_3.sh report script2.o", and "3.tmp lab13_1.sh lab13_4.sh script2 text.txt". The prompt then changes to "emshehacov@emshehacov:~/work/study/2023-2024/Операционные системы/ос/". The command being executed is "tar: ./arhiv.tar: archive cannot contain itself; not dumped". The output of the command is a list of files and directories: "1.tmp 4.tmp lab13_1.sh lab13_4.sh script2 text.txt", "2.tmp arhiv.tar lab13_2.sh presentation script2.c", and "3.tmp fout.txt lab13_3.sh report script2.o". The prompt then changes to "emshehacov@emshehacov:~/work/study/2023-2024/Операционные системы/ос/". The command being executed is "ls". The output of the command is a list of files and directories: "1.tmp 4.tmp lab13_1.sh lab13_4.sh script2 text.txt", "2.tmp arhiv.tar lab13_2.sh presentation script2.c", and "3.tmp fout.txt lab13_3.sh report script2.o".

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

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

tar: ./arhiv.tar: archive cannot contain itself; not dumped
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
```

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.