

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Сотникова Виолетта Алексеевна¹

28 августа, 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

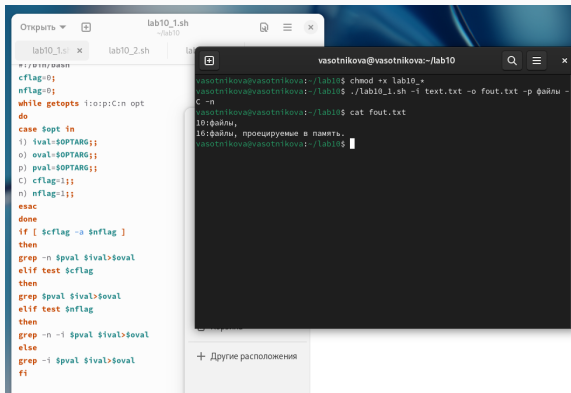
1 Выполнить 4 задания

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; `-p 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



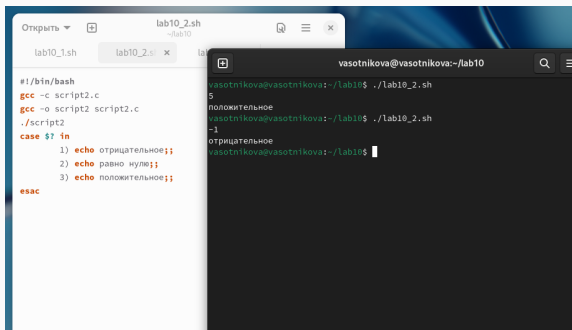
The image shows two terminal windows. The background window, titled 'lab10_1.sh', displays a shell script with a 'while' loop that reads from 'i:o:pC:n opt' and processes options. The foreground window, titled 'vasotnikova@vasotnikova:~/lab10', shows the execution of the script. The user runs 'chmod +x lab10_1.sh', then './lab10_1.sh -i text.txt -o fout.txt -p файлы -C -n'. The output of the script is shown in the foreground window, indicating that the files were processed and the results were written to 'fout.txt'.

```
vasotnikova@vasotnikova:~/lab10$ chmod +x lab10_1.sh
vasotnikova@vasotnikova:~/lab10$ ./lab10_1.sh -i text.txt -o fout.txt -p файлы -C -n
vasotnikova@vasotnikova:~/lab10$ cat fout.txt
10:файлы,
10:файлы, проецирование в память.
vasotnikova@vasotnikova:~/lab10$
```

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 two overlapping terminal windows. The background window, titled 'lab10_2.sh', displays the source code of a shell script. The script starts with a shebang, compiles a C program 'script2.c' using 'gcc', and then runs it. A 'case' statement checks the output of 'script2' and prints 'отрицательное' (negative), 'равно нулю' (equal to zero), or 'положительное' (positive) based on the value. The foreground window, titled 'vasotnikova@vasotnikova:~/lab10', shows the script being executed. The output matches the script's logic: it prints '5' followed by 'положительное' (positive), '-1' followed by 'отрицательное' (negative), and then a prompt character.

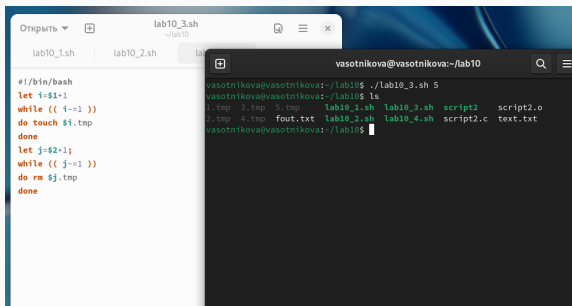
```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
  1) echo отрицательное;;
  2) echo равно нулю;;
  3) echo положительное;;
esac
```

```
vasotnikova@vasotnikova:~/lab10$ ./lab10_2.sh
5
положительное
vasotnikova@vasotnikova:~/lab10$ ./lab10_2.sh
-1
отрицательное
vasotnikova@vasotnikova:~/lab10$
```

Рис. 2: Задание 2

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

Выполнение работы



The image shows two terminal windows. The left window, titled 'lab10_3.sh', contains a shell script that iterates over files in the current directory, touching them, and then deleting them. The right window, titled 'vasotnikova@vasotnikova:~/lab10', shows the execution of 'lab10_3.sh' and the resulting directory listing.

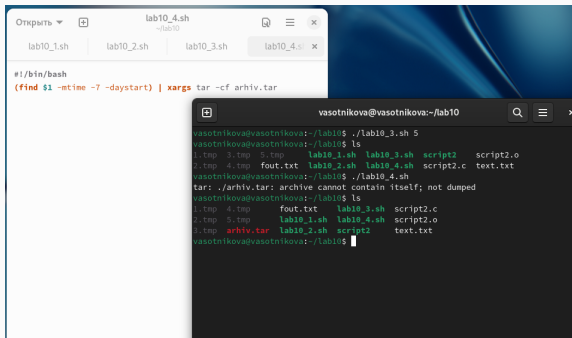
```
#!/bin/bash
let i=$#-1
while (( i>=1 ))
do touch $i.tmp
done
let j=$#-1;
while (( j>=1 ))
do rm $j.tmp
done
```

```
vasotnikova@vasotnikova:~/lab10$ ./lab10_3.sh 5
vasotnikova@vasotnikova:~/lab10$ ls
1.tmp  3.tmp  5.tmp  lab10_1.sh  lab10_3.sh  script2  script2.o
2.tmp  4.tmp  fout.txt lab10_2.sh  lab10_4.sh  script2.c  text.txt
vasotnikova@vasotnikova:~/lab10$
```

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 two terminal windows. The top window, titled 'lab10_4.sh', displays the command `(find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar`. The bottom window, titled 'vasotnikova@vasotnikova:~/lab10', shows a sequence of commands and their outputs, including `./lab10_3.sh 5`, `ls`, and `tar: ./arhiv.tar: archive cannot contain itself; not dumped`.

```
vasotnikova@vasotnikova:~/lab10$ ./lab10_3.sh 5
vasotnikova@vasotnikova:~/lab10$ ls
1.tmp 3.tmp 5.tmp lab10_1.sh lab10_3.sh script2 script2.o
2.tmp 4.tmp fout.txt lab10_2.sh lab10_4.sh script2.c text.txt
vasotnikova@vasotnikova:~/lab10$ ./lab10_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
vasotnikova@vasotnikova:~/lab10$ ls
1.tmp 4.tmp fout.txt lab10_3.sh script2.c
2.tmp 5.tmp lab10_1.sh lab10_4.sh script2.o
3.tmp arhiv.tar lab10_2.sh script2 text.txt
vasotnikova@vasotnikova:~/lab10$
```

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.