

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Зокиров Аллохназар¹

19 мая, 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

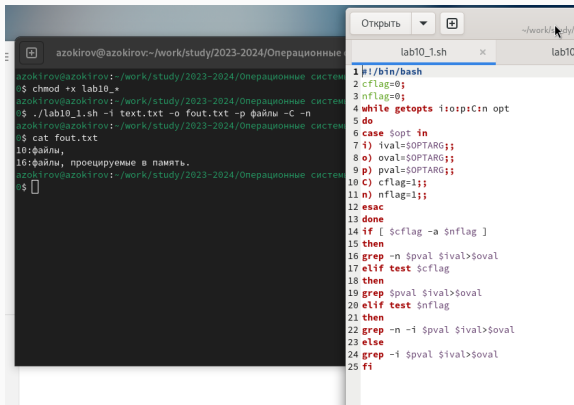
1 Выполнить 4 задания

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 on the left and a code editor on the right. The terminal window displays the execution of a script named `lab10_1.sh`. The user `azokirov` is in the directory `~/work/study/2023-2024/Операционные системы`. The script is executed with arguments `-i text.txt -o fout.txt -p файлы -C -n`. The output shows that the script processes the file `text.txt` and writes the result to `fout.txt`. The code editor on the right shows the source code of `lab10_1.sh`, which is a shell script that sets flags and processes input files.

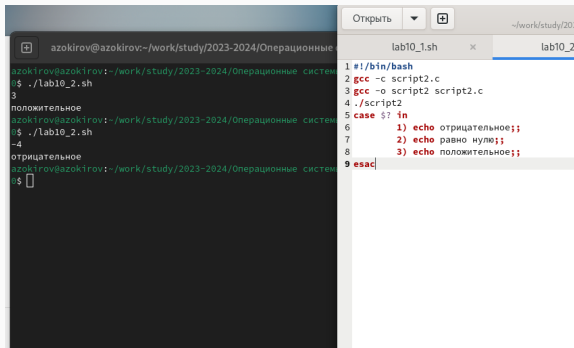
```
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
0$ chmod +x lab10_*
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
0$ ./lab10_1.sh -i text.txt -o fout.txt -p файлы -C -n
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
0$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
0$
```

```
1#!/bin/bash
2cflag=0;
3nflag=0;
4while getopts i:op:C:n opt
5do
6case $opt in
7i) ival=$OPTARG;;
8o) oval=$OPTARG;;
9p) pval=$OPTARG;;
10C) cflag=1;;
11n) nflag=1;;
12esac
13done
14if [ $cflag -a $nflag ]
15then
16grep -n $pval $ival>$oval
17elif test $cflag
18then
19grep $pval $ival>$oval
20elif test $nflag
21then
22grep -n -i $pval $ival>$oval
23else
24grep -i $pval $ival>$oval
25fi
```

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 on the left and a code editor on the right. The terminal window displays the execution of a script named `lab10_2.sh`. The user runs the script twice, first with a positive argument and then with a negative argument. The code editor shows the source code of `lab10_2.sh`, which uses `gcc` to compile `script2.c` and then runs `script2` with a `case` statement to print different messages based on the input.

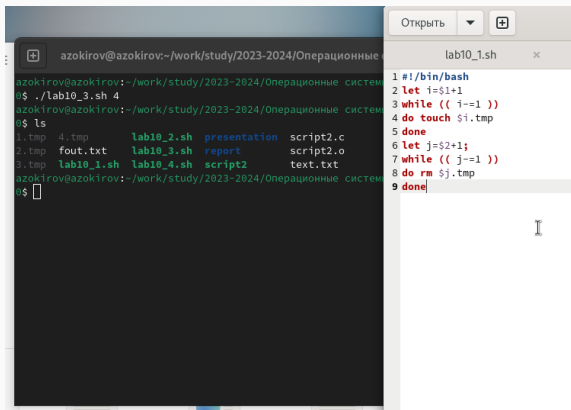
```
azokirov@azokirov:~/work/study/2023-2024/Операционные системы$ ./lab10_2.sh
3
положительное
azokirov@azokirov:~/work/study/2023-2024/Операционные системы$ ./lab10_2.sh
-4
отрицательное
azokirov@azokirov:~/work/study/2023-2024/Операционные системы$
```

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

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 on the left and a file editor on the right. The terminal window has a title bar with a plus icon and the text "azokirov@azokirov:~/work/study/2023-2024/Операционные системы". The terminal content shows the user running a script and listing files. The file editor has a title bar with "Открыть", a dropdown arrow, and a plus icon, and a tab titled "lab10_1.sh". The editor content shows a shell script with a while loop and touch command.

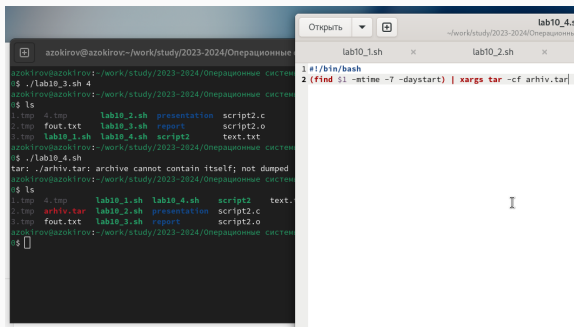
```
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
0$ ./lab10_3.sh 4
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
0$ ls
1.tmp 4.tmp lab10_2.sh presentation script2.c
2.tmp fout.txt lab10_3.sh report script2.o
3.tmp lab10_1.sh lab10_4.sh script2 text.txt
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
0$
```

```
1 #!/bin/bash
2 let i=$1+1
3 while (( i-=1 ))
4 do touch $i.tmp
5 done
6 let j=$2+1;
7 while (( j-=1 ))
8 do rm $j.tmp
9 done
```

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 window. The terminal window displays the following commands and output:

```
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
$ ./lab10_3.sh 4
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
$ ls
1.tmp  4.tmp      lab10_2.sh  presentation  script2.c
2.tmp  fout.txt    lab10_3.sh  report        script2.o
3.tmp  lab10_1.sh  lab10_4.sh  script2       text.txt
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
$ ./lab10_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
$ ls
1.tmp  4.tmp      lab10_1.sh  lab10_4.sh  script2  text.
2.tmp  arhiv.tar  lab10_2.sh  presentation  script2.c
3.tmp  fout.txt  lab10_3.sh  report        script2.o
azokirov@azokirov:~/work/study/2023-2024/Операционные системы
$
```

The file manager window shows the contents of the `lab10_4.sh` file:

```
1 #!/bin/bash
2 (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.