

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Уржиндорж Мягмар¹

20 июня, 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

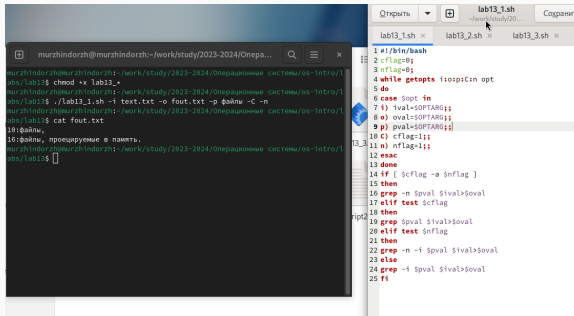
1 Выполнить 4 задания

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 shell script named `lab13_1.sh`. The user `murzhindorzh` is in the directory `~/work/study/2023-2024/Операционные системы/os-intro/lab13`. The script performs the following actions:

- Creates a directory `lab13_1` using `mkdir -p lab13_1`.
- Runs `./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n`.
- Displays the output of `cat fout.txt`, which shows: `16:файлы, проецируемые в память.`

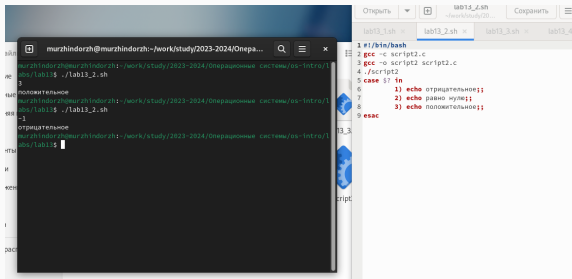
The code editor on the right shows the source code of `lab13_1.sh`. The script is a Bash script that takes command-line options and processes them. The code is as follows:

```
1 #!/bin/bash
2 cflag=0
3 nflag=0
4 while getopts i:osp:C:n opt
5 do
6 case $opt in
7 i) ival=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval $ival>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $ival>$oval
23 else
24 grep -i $pval $ival>$oval
25 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 'murzhindorzh@murzhindorzh: ~/work/study/2023-2024/Опера...', displays the execution of a script named 'lab13_2.sh'. The script's output is as follows:

```
murzhindorzh@murzhindorzh: ~/work/study/2023-2024/Опера...  
$ ./lab13_2.sh  
3  
положительное  
murzhindorzh@murzhindorzh: ~/work/study/2023-2024/Опера...  
$ ./lab13_2.sh  
-1  
отрицательное  
murzhindorzh@murzhindorzh: ~/work/study/2023-2024/Опера...  
$ ./lab13_2.sh
```

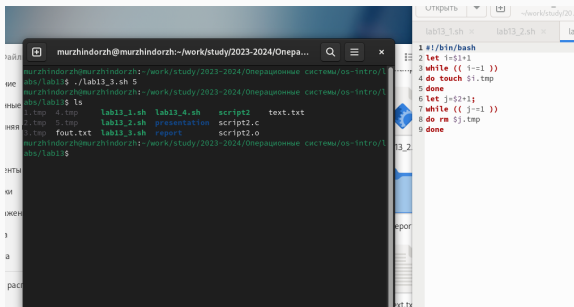
The code editor, titled 'lab13_2.sh', shows the following script content:

```
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 and a file explorer. The terminal window, titled 'murzhindorzh@murzhindorzh:~/work/study/2023-2024/Операционные системы/ос-интро/...', displays the following commands and output:

```
murzhindorzh@murzhindorzh:~/work/study/2023-2024/Операционные системы/ос-интро/abs/lab13$ ./lab13_3.sh 5
murzhindorzh@murzhindorzh:~/work/study/2023-2024/Операционные системы/ос-интро/abs/lab13$ ls
1.tmp  4.tmp      lab13_1.sh  lab13_4.sh  script2    text.txt
2.tmp  5.tmp      lab13_2.sh  presentation script2.c
3.tmp  fout.txt   lab13_3.sh  report      script2.o
murzhindorzh@murzhindorzh:~/work/study/2023-2024/Операционные системы/ос-интро/abs/lab13$
```

The file explorer, titled 'Открыть' and showing the path '~/work/study/20...', displays the following files:

```
1#!/bin/bash
2let i=51+1
3while (( i-->1 ))
4do touch $i.tmp
5done
6let j=52+1;
7while (( j-->1 ))
8do rm $j.tmp
9done
```

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

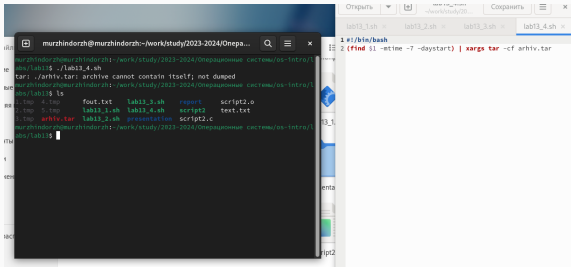


Рис. 4: Задание 4

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.