

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Абдул Васа Файсал Ахмад¹

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

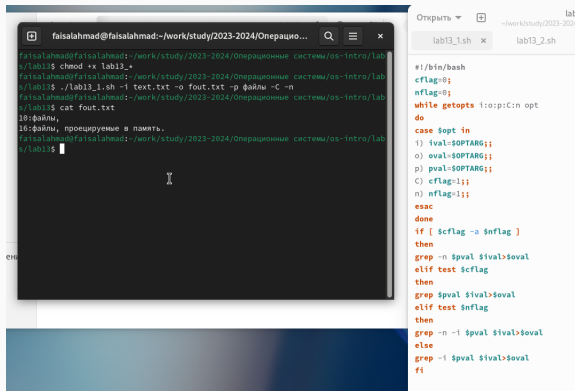
1 Выполнить 4 задания

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 on the left and a code editor on the right. The terminal window has a title bar that reads "faisalalahmad@faisalalahmad:~/work/study/2023-2024/Операционные системы/Операционные системы/os-intro/lab 3/lab13". The terminal output shows the user running "chmod +x lab13_1.sh", then ". /lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n", and finally "cat fout.txt". The output of the script is "10:файлы, 16:файлы, проецируемые в память." followed by a prompt. The code editor on the right has a title bar that reads "lab" and shows the content of "lab13_1.sh". The script is a shell script that takes options and arguments and uses "grep" to search for them in a file named "файлы".

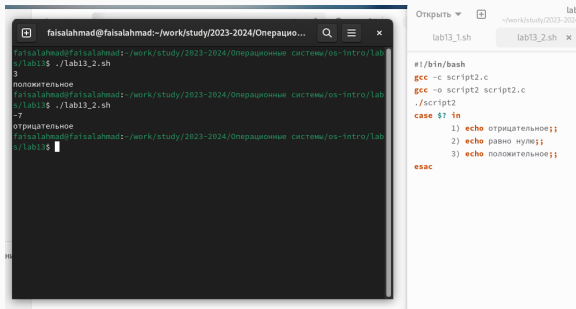
```
faisalalahmad@faisalalahmad:~/work/study/2023-2024/Операционные системы/Операционные системы/os-intro/lab 3/lab13$ chmod +x lab13_1.sh
faisalalahmad@faisalalahmad:~/work/study/2023-2024/Операционные системы/Операционные системы/os-intro/lab 3/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
faisalalahmad@faisalalahmad:~/work/study/2023-2024/Операционные системы/Операционные системы/os-intro/lab 3/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
faisalalahmad@faisalalahmad:~/work/study/2023-2024/Операционные системы/Операционные системы/os-intro/lab 3/lab13$
```

```
#!/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
```

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 `lab13_2.sh`. The user runs the script, and it outputs the number 3, followed by the word "положительное" (positive), and then -7, followed by the word "отрицательное" (negative). The code editor on the right shows the source code of the script, which includes a C program to calculate the sum of two numbers and a shell script that checks if the result is positive, zero, or negative.

```
faisalhamad@faisalhamad:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ./lab13_2.sh
3
положительное
faisalhamad@faisalhamad:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$ ./lab13_2.sh
-7
отрицательное
faisalhamad@faisalhamad:~/work/study/2023-2024/Операционные системы/os-intro/labs/lab13$
```

```
#!/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
```

Рис. 2: Задание 2

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

Выполнение работы

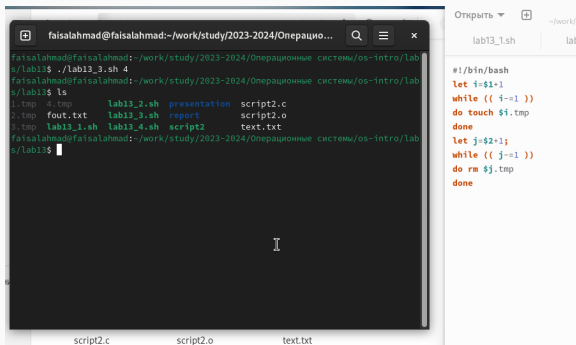


Рис. 3: Задание 3

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

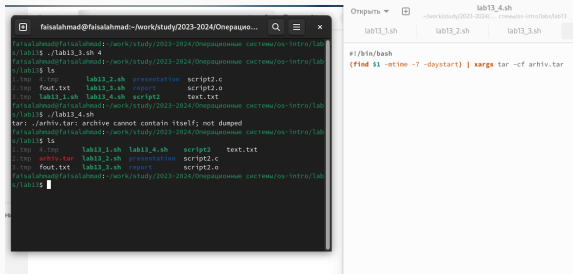


Рис. 4: Задание 4

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.