

Структура научной презентации по лабораторной работе 13

Операционные системы

Гомес Лопес Теофания

05 мая 2025

Российский университет дружбы народов, Москва, Россия

Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

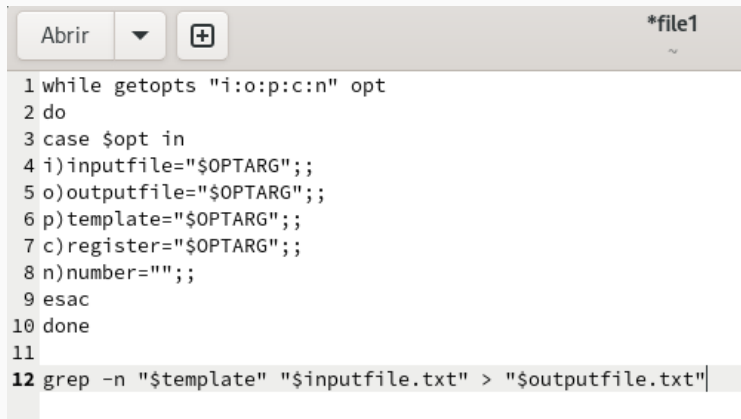
Задание

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N
4. Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории.

Выполнение лабораторной работы

Создаю файл `file1` и в нем написала код, который анализирует командную строку с ключами `-i` (прочитать данные из указанного файла), `-o` (вывести данные в указанный файл), `-p` (указать шаблон для поиска), `-C` (различать большие и малые буквы), `-n` (выдавать номера строк) используя команды `getopts` `grep`:

командный файл, который анализирует командную строку



The image shows a terminal window with a title bar containing 'Abrir', a dropdown arrow, a plus icon, and the filename '*file1'. The terminal displays a shell script with 12 lines. Line 12 is highlighted with a light blue background. The script uses a while loop to parse options and a grep command to search for a template in an input file.

```
1 while getopts "i:o:p:c:n" opt
2 do
3 case $opt in
4 i)inputfile="$OPTARG";;
5 o)outputfile="$OPTARG";;
6 p)template="$OPTARG";;
7 c)register="$OPTARG";;
8 n)number="";;
9 esac
10 done
11
12 grep -n "$template" "$inputfile.txt" > "$outputfile.txt"
```

Рис. 1: код для анализирования командной строки

командный файл, который анализирует командную строку

```
teofaniagomeslopes@teofanialopes:~$ ./file1.txt -i file1 -o output -p n etconf -  
C -n  
teofaniagomeslopes@teofanialopes:~$ cat output.txt  
1:while getopt "i:o:p:c:n" opt  
3:case $opt in  
4:i)inputfile="$OPTARG";;  
8:n)number="";;  
10:done  
12:grep -n "$template" "$inputfile.txt" > "$outputfile.txt"  
teofaniagomeslopes@teofanialopes:~$
```

Рис. 2: Запуск file1

программа, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.

Написала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.

```
1 #include <stdlib.h>
2 #include <stdio.h>
3
4 int main()
5 {
6     int n;
7     printf("Enter a number: ");
8     scanf("%d", &n);
9     if(n>0)
10    {
11        exit(1);
12    }
13    else if (n==0) {
14        exit(0);}
15    else
16    {
17        exit(2);
```

программа, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.

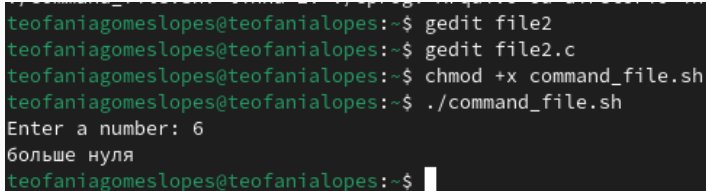
Далее создала командный файл который вызывает эту программу и, проанализировав с помощью команды \$?, выдает сообщение о том, какое число было введено:

```
1 gcc -o cprog file2.c
2 ./cprog
3
4 case $? in
5 0) echo "равно нулю";;
6 1) echo "больше нуля";;
7 2) echo "меньше нуля";;
8
9 esac
```

Рис. 4: Командный файл программы на Си

программа, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю.

Создала исполняемый файл и запустила:



```
teofaniagomeslopes@teofanialopes:~$ gedit file2
teofaniagomeslopes@teofanialopes:~$ gedit file2.c
teofaniagomeslopes@teofanialopes:~$ chmod +x command_file.sh
teofaniagomeslopes@teofanialopes:~$ ./command_file.sh
Enter a number: 6
больше нуля
teofaniagomeslopes@teofanialopes:~$
```

Рис. 5: Результаты программы

командный файл, создающий указанное число файлов

Я написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `?`.

```
1 for((i=1; i<=$*; i++))  
2 do  
3 if test -f "$i".tmp  
4 then rm "$i".tmp  
5 else touch "$i.tmp"  
6 fi  
7 done
```

Рис. 6: Командный файл для создания файлов

командный файл, создающий указанное число файлов

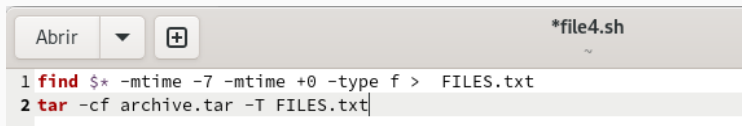
Создала исполняемый файл и запустила:

```
teofaniagomeslopes@teofanialopes:~$ chmod +x file3.sh
teofaniagomeslopes@teofanialopes:~$ ./file3.sh 3
teofaniagomeslopes@teofanialopes:~$ ls
1.tmp          Downloads      lab7.sh~      parentdir2
2.tmp          feathers       LICENSE       parentdir3
3.tmp          file1          may           Pictures
abc1           file1.txt     Milly         play
'Área de trabalho' file2         Modelos       project1.sh
'Arquitetura de computadoresre' file2.c       monthly      project2.sh
australia      file3.sh      Músicas       project3.sh
backup         file.txt      my_os         project.sh
bin            fun           'New Directory' Público
command_file.sh git-extended  ny_os         reports
conf.txt       helloworld.cpp output.txt     ski.plases
cprog          Imagens      '~p'          text.txt
Documentos    '#lab7.sh#'  parentdir     Vídeos
Documents     lab7.sh      parentdir1    work
teofaniagomeslopes@teofanialopes:~$
```

Рис. 7: Создание файлов с помощью командного файла

командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории.

создала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории.



```
1 find $* -mtime -7 -mtime +0 -type f > FILES.txt
2 tar -cf archive.tar -T FILES.txt
```

Рис. 8: Создание архива

командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории.

```
teofaniagomeslopes@teofanialopes:~$ chmod +x file4.sh
teofaniagomeslopes@teofanialopes:~$ ./file4.sh /home/teofaniagomeslopes/work
tar: Removendo "/" inicial dos nomes dos membros
tar: Removendo "/" inicial dos alvos de links físicos
teofaniagomeslopes@teofanialopes:~$ ls ~/work
arch-pc  github.io  os  study
teofaniagomeslopes@teofanialopes:~$ ls
1.tmp      Downloads      lab7.sh      parentdir2
2.tmp      feathers       lab7.sh~    parentdir3
3.tmp      file1          LICENSE     Pictures
abc1       file1.txt      may         play
archive.tar file2          Milly       project1.sh
'Área de trabalho' file2.c        Modelos     project2.sh
```

Рис. 9: Результаты кода

Выводы

При выполнении проделанной работы я научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.