

Презентация по лабораторной работе №13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Джаллох Ишмаил

10 мая 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Джаллох Ишмаил
- НКА 05-24
- факультет физико математических и естественных наук
- Российский университет дружбы народов
- 1032239614@rudn.ru
- <https://lsho-jah.github.io/ru/>

.....
.....

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 написал 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

```
while getopts "i:o:p:C:n" opt
do
case $opt in
i)inputfile="$OPTARG";;
o)outputfile="$OPTARG";;
p)template="$OPTARG";;
c)register="$OPTARG";;
n)number="";;
esac
done
```

```
grep -n "$template" "$inputfile.txt" > "$outputfile.txt"
```

```
ijalloh@ijalloh:~$ ./file1.txt -i file1 -o output -p n etconf -C -n
ijalloh@ijalloh:~$ cat output.txt
1:while getopts "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"
```

Рис. 1: Запуск file1

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

Программа завершается с помощью функции `exit(n)`, передавая информацию о коде завершения в оболочку:

```
#include <stdlib.h>
#include <stdio.h>

int main()
{
    int n;
    printf("Enter a number: ");
    scanf("%d", &n);
    if(n>0)
```

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

```
{  
    exit(1);  
}  
  
else if (n==0) {  
    exit(0);}   
else  
{  
    exit(2);  
}  
}
```

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

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

```
gcc -o cprog file2.c  
./cprog
```

```
case $? in  
0) echo "равно нулю";;  
1) echo "больше нуля";;  
2) echo "меньше нуля";;
```

```
esac
```

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

Создал исполняемый файл и запустил:

```
ijalloh@ijalloh:~$ gedit file2.c
ijalloh@ijalloh:~$ gedit command_file.sh
ijalloh@ijalloh:~$ chmod +x command_file.sh
ijalloh@ijalloh:~$ ./commad_file.sh
bash: ./commad_file.sh: No such file or directory
ijalloh@ijalloh:~$ ./command_file.sh
Enter a number: 7
больше нуля
ijalloh@ijalloh:~$
```

Рис. 2: Результаты программы

командный файл, создающий указанное число файлов

Я написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют):

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


Создал исполняемый файл и запустил:

```
ijalloh@ijalloh:~$ ./file3.sh 3
ijalloh@ijalloh:~$ ls
1.tmp      Desktop    go         my
2.tmp      Documents  helloworld.cpp 'Ne
3.tmp      Downloads  '#lab07.sh#' ou
abc1       file1      '#lab07.sh#' pi
```

Рис. 3: Создание файлов с помощью командного файла

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

создал командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывался только те файлы, которые были изменены менее недели тому назад (использовать команду find).

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

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

```
ijalloh@ijalloh:~$ gedit file4.sh
ijalloh@ijalloh:~$ chmod +x file4.sh
ijalloh@ijalloh:~$ ./file4.sh /home/ijalloh/work
ijalloh@ijalloh:~$ ls ~/work
blog os study
ijalloh@ijalloh:~$ ls
1.tmp          Desktop      git-extended  my_os
2.tmp          Documents   go            'New Directory'
3.tmp          Downloads   helloworld.cpp output.txt
abc1           file1       '#lab07.sh#'  Pictures
archive.tar    file1.txt   '#lab07.sh#~' play
australia      file2.c     lab07.sh      project1.sh
backup         file3.sh    '#LICENSE#'   project2.sh
bin            file4.sh    LICENSE       project3.sh
command_file.sh FILES.txt   may           project3.sh~
conf.txt       file.txt    monthly       project.sh
cprog         fun         Music         Public
ijalloh@ijalloh:~$
```

Рис. 4: Результаты кода

Выводы

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