

# Лабораторная работа №11

---

Матюшкин Денис Владимирович (НПИбд-02-21)

26.05.2022

RUDN University, Moscow, Russian Federation

## Цель работы

---

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

## Ход работы

---

## 1.1. Командный файл, который анализирует командную строку

- Используя команды `getopts` `grep`, напомним командный файл, который анализирует командную строку с ключами:
  - `-iinputfile` — прочитать данные из указанного файла;
  - `-ooutputfile` — вывести данные в указанный файл;
  - `-ршаблон` — указать шаблон для поиска;
  - `-C` — различать большие и малые буквы;
  - `-n` — выдавать номера строк.

а затем ищет в указанном файле нужные строки, определяемые ключом `-р` (рис. 1). Проверим написанный командный файл (рис. 2).

```
#!/bin/bash
while getopts "i:op:C:n" opt
do
  case $opt in
    i) input=$OPTARG;;
    o) output=$OPTARG;;
    p) shablon=$OPTARG;;
    C) reglist="";;
    n) number="";;
    *) ;;
  esac
done
grep -n "$shablon" "$input" > "$output"
```

Рис. 1: Командный файл поиска файлов

## 1.2. Проверка

```
[dvmatyushkin@dvmatyushkin ~]$ chmod +x lab11_1.sh
[dvmatyushkin@dvmatyushkin ~]$ ./lab11_1.sh -i text.txt -o test.txt -p "sh" -C -n
[dvmatyushkin@dvmatyushkin ~]$ cat test.txt
1:lab1.sh
[dvmatyushkin@dvmatyushkin ~]$ ./lab11_1.sh -i text.txt -o test.txt -p "tg" -C -n
[dvmatyushkin@dvmatyushkin ~]$ cat test.txt
5:zxc.tg
6:xyt.tg
[dvmatyushkin@dvmatyushkin ~]$ ./lab11_1.sh -i text.txt -o test.txt -p "" -C -n
[dvmatyushkin@dvmatyushkin ~]$ cat test.txt
1:lab1.sh
2:lab2.txt
3:lab3.cpp
4:abc
5:zxc.tg
6:xyt.tg
7:stifell.io
8:nice
9:good
10:well
[dvmatyushkin@dvmatyushkin ~]$
```

Рис. 2: Проверка файла

## 2.1. Программа на Си

- Напишем на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл вызывает эту программу и, проанализировав с помощью команды `$?`, выдает сообщение о том, какое число было введено (рис. 3 и рис. 4). Проверим написанный командный файл (рис. 5).

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

int main(){
    int n;
    printf("Введите число: "); scanf("%d",&n);
    if (n > 0) printf("Число больше нуля\n", n);
    else if (n < 0) printf("Число меньше нуля\n", n);
    else printf("Число равно нулю\n", n);
    printf("Введеное число: "); exit(n);
    return n;
}
```

Рис. 3: Программа на Си

## 2.2. Проверка

```
#!/bin/bash

gcc lab11.c -o lab11
./lab11
echo $?
```

Рис. 4: Командный файл

```
[dvmatyushkin@dvmatyushkin ~]$ emacs lab11.c
[dvmatyushkin@dvmatyushkin ~]$ emacs lab11_2.sh
[dvmatyushkin@dvmatyushkin ~]$ chmod +x lab11_2.sh
[dvmatyushkin@dvmatyushkin ~]$ ./lab11_2.sh
Введите число: 5
Число больше нуля
Введенное число: 5
[dvmatyushkin@dvmatyushkin ~]$ ./lab11_2.sh
Введите число: 0
Число равно нулю
Введенное число: 0
[dvmatyushkin@dvmatyushkin ~]$
```

Рис. 5: Проверка файла



## 3.1. Командный файл, который создает указанное число файлов

- Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые создается, передаётся в аргументы командной строки. Этот же командный файл удаляет все созданные им файлы (если они существуют) (рис. 6). Проверим написанный командный файл (рис. 7).

```
#!/bin/bash

while getopts "c:r" opt
do
    case $opt in
        c)n=$OPTARG;
        for i in $(seq 1 $n);
        do
            touch "$i.tmp";
            done;;
        r)for i in $(find -name "*.tmp");
        do
            rm $i;
            done;;
    esac
done
```

Рис. 6: Командный файл для создание файлов

## 3.2. Проверка

```
[dvmatyushkin@dvmatyushkin ~]$ chmod +x lab11_3.sh
[dvmatyushkin@dvmatyushkin ~]$ ./lab11.sh -c 10
bash: ./lab11.sh: Нет такого файла или каталога
[dvmatyushkin@dvmatyushkin ~]$ ./lab11_3.sh -c 10
[dvmatyushkin@dvmatyushkin ~]$ ls
10.tmp  5.tmp  bin      lab11_2.sh~  lab11.c~    resources  themes  Изображени
1.tmp   6.tmp  lab11    lab11_3.sh  lab11.cpp~  snap       work    Музыка
2.tmp   7.tmp  lab11_1.sh  lab11_3.sh~  opt        test.txt   Видео   Общедоступ
3.tmp   8.tmp  lab11_1.sh~ lab11_4.sh  output     text.txt   Документы 'Рабочий ст
4.tmp   9.tmp  lab11_2.sh  lab11.c     outputfile  text.txt~  Загрузки  Шаблоны

[dvmatyushkin@dvmatyushkin ~]$ ./lab11_3.sh -r
find: './.local/share/Trash/files/monthly/play': Отказано в доступе
[dvmatyushkin@dvmatyushkin ~]$ ls
bin      lab11_2.sh  lab11_4.sh  opt      snap      themes  Загрузки
lab11    lab11_2.sh~ lab11.c     output   test.txt  work    Изображения
lab11_1.sh  lab11_3.sh  lab11.c~    outputfile  text.txt  Видео   Музыка
lab11_1.sh~ lab11_3.sh~ lab11.cpp~  resources  text.txt~ Документы  Общедоступные
[dvmatyushkin@dvmatyushkin ~]$
```

Рис. 7: Проверка файла

## 4.1. Командный файл архиватор

- 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 (используем команду find) (рис. 8). Проверим написанный командный файл (рис. 9 и рис. 10).

```
#!/bin/bash

while getopts "p:" opt
do
    case $opt in
        p)dir=$OPTARG;;
    esac
done

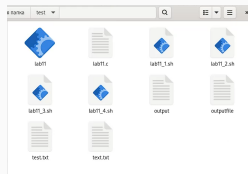
find $dir -maxdepth 1 -ctime -7 -type f -print0 | xargs -0 tar rfv test.tar
```

Рис. 8: Командный файл архиватор

## 4.2. Проверка

```
[dvmatyushkin@dvmatyushkin ~]$ emacs lab11_4.sh
[dvmatyushkin@dvmatyushkin ~]$ chmod +x lab11_4.sh
[dvmatyushkin@dvmatyushkin ~]$ ./lab11_4.sh
./vboxclient-clipboard.pid
./vboxclient-seamless.pid
./vboxclient-draganddrop.pid
./vboxclient-display-svg-x11.pid
./bash_history
./outputfile
./output
./lab11_1.sh~
./lab11_1.sh
./lab11.cpp~
./lab11_2.sh~
./lab11_2.sh
./lab11_4.sh
./text.txt~
./text.txt
./test.txt
./lab11.c~
./lab11.c
./lab11
./lab11_3.sh~
./lab11_3.sh
[dvmatyushkin@dvmatyushkin ~]$
```

Рис. 9: Проверка файла



## Вывод

---

- В ходе этой лабораторной работы мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Спасибо за внимание!