Лабораторная работа №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Панченко Денис Дмитриевич

Содержание

Список иллюстраций

1 Цель работы

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

2 Выполнение лабораторной работы

Используя команды getopts grep, напишем командный файл, который анализирует командную строку с заданными ключами, а затем ищет в указанном файле нужные строки, определяемые ключом -p. (рис. 2.1 - 2.2).

```
#!/bin/bash

input_file=""
output_file=""
pattern=""
case_sensitive=""
line_number=""

while getopts "i:o:p:Cn" opt; do
  case ${opt} in
    i ) input_file=$OPTARG;;
    o ) output_file=$OPTARG;;
    p ) pattern=$OPTARG;;
    C ) case_sensitive="-i";;
    n ) line_number="-n";;
    \? ) echo "Invalid option: -$OPTARG" 1>&2;;
    : ) echo "Option -$OPTARG requires an argument." 1>&2;;
  esac
done
```

Рис. 2.1: Командный файл

```
[ddpanchenko@ddpanchenko ~]$ ./lab11.sh -i lab11.txt -p "hello"
[ddpanchenko@ddpanchenko ~]$ ./lab11.sh -i lab11.txt -p "hello" -o lab11-1.txt
[ddpanchenko@ddpanchenko ~]$ ./lab11.sh -i lab11.txt -p "Hello" -C
[ddpanchenko@ddpanchenko ~]$ ./lab11.sh -i lab11.txt -p "Hello" -n
```

Рис. 2.2: Команды

Напишем на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Командный файл должен вызывать эту программу и выдать сообщение о том, какое число было введено (рис. 2.3 - 2.5).

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

int main()
{
  printf("Введите число: ");
  int a;
  scanf("%d",&a);
  if (a<0) exit(0);
  if (a>0) exit(1);
  if (a==0) exit(2);
  return 0;
}
```

Рис. 2.3: Программа

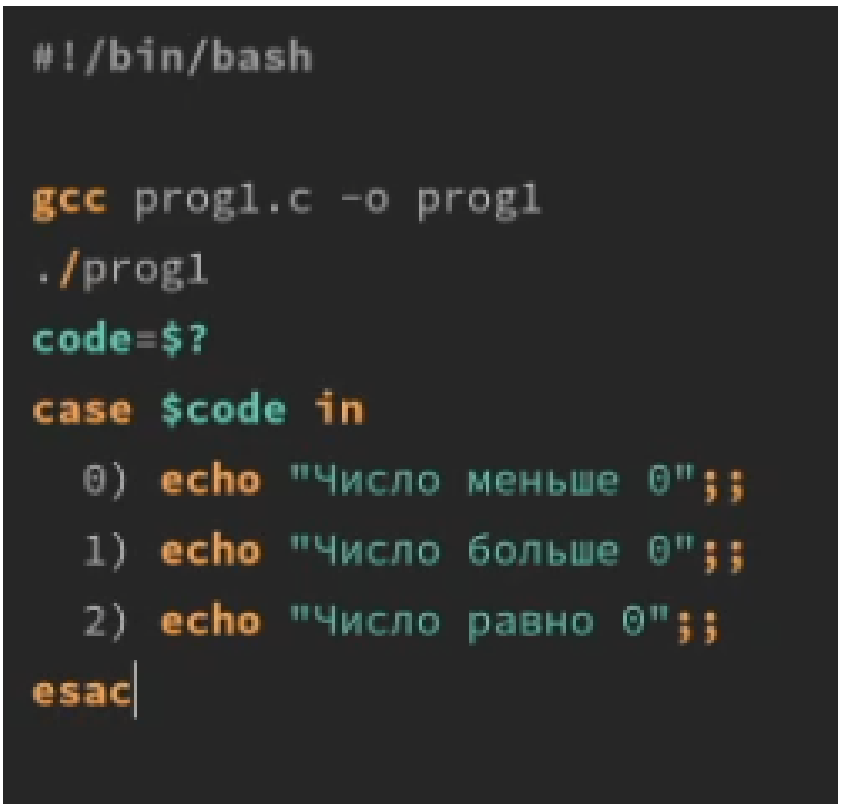

Рис. 2.4: Командный файл

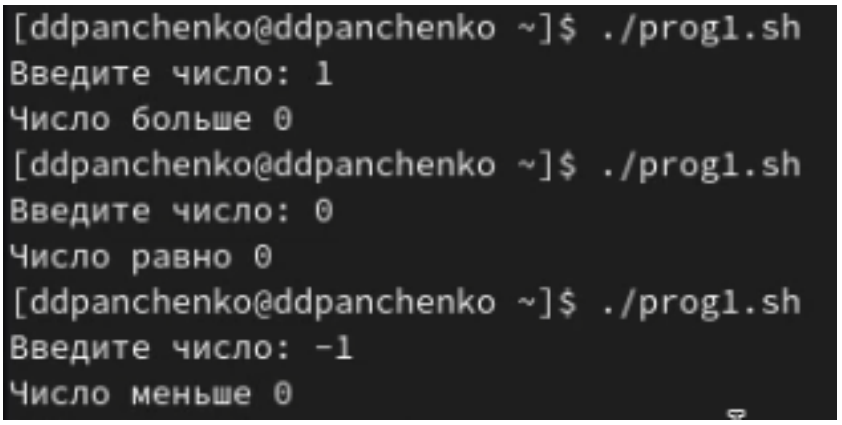

Рис. 2.5: Команда

Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Этот же командный файл должен уметь удалять все созданные им файлы (рис. 2.6 - 2.8).

```
#!/bin/bash

opt=$1;
form=$2;
num=$3;
function Files()
{
    for ((i=1; i<=$num; i++)) do
        file=$(echo $form | tr '#' "$i")
        if [ $opt == "-r" ]
        then
            rm -f $file
        elif [ $opt == "-c" ]
        then
            touch $file
        fi
    done
}
Files
```

Рис. 2.6: Командный файл

```
[ddpanchenko@ddpanchenko ~]$ ls
 australia   lab11-1.txt   prog1       script2.sh   Документы       Шаблоны
 conf.txt    lab11.sh      prog1.c     script.sh    Загрузки
 feathers    lab11.sh~     prog1.c~    ski.plases   Изображения
 file2.sh    lab11.txt     prog1.sh    text.txt     Музыка
 file.sh     my_os         prog2.sh    work         Общедоступные
 file.txt    play          prog2.sh~   Видео       'Рабочий стол'
[ddpanchenko@ddpanchenko ~]$ ./prog2.sh -c a#.txt 3
[ddpanchenko@ddpanchenko ~]$ ls
 a1.txt      file2.sh      lab11.txt   prog1.sh     text.txt      Музыка
 a2.txt      file.sh       my_os       prog2.sh     work          Общедоступные
 a3.txt      file.txt      play        prog2.sh~    Видео        'Рабочий стол'
```

Рис. 2.7: Создание файлов

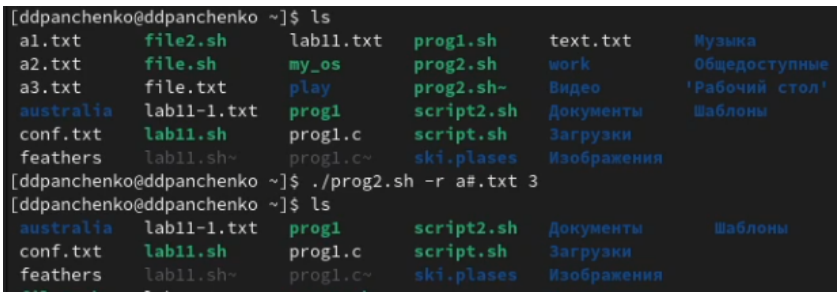

Рис. 2.8: Удаление файлов

Напишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории (рис. 2.9 - 2.12).

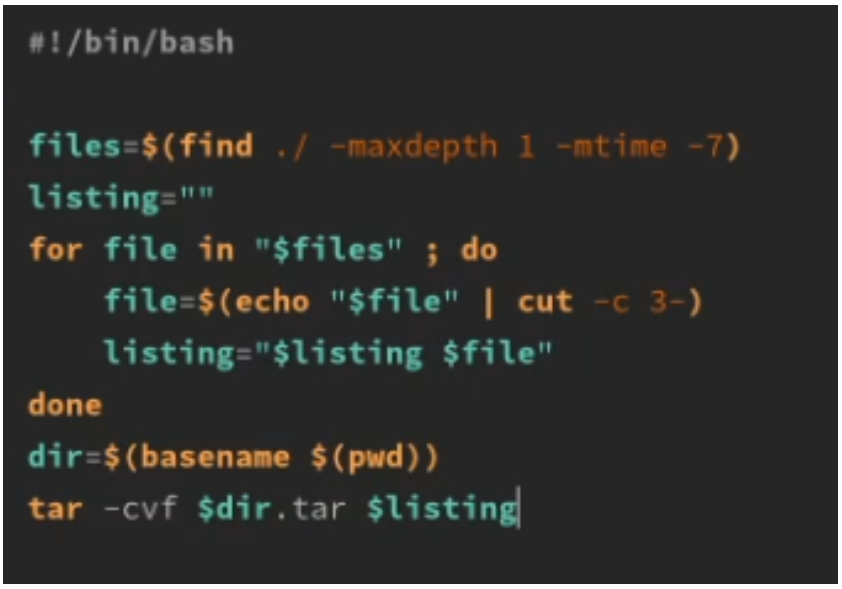

Рис. 2.9: Командный файл

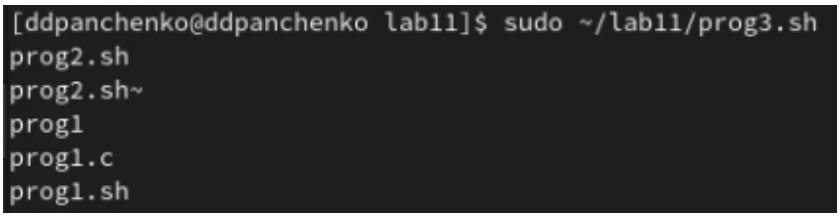

Рис. 2.10: Создание архива

```
[ddpanchenko@ddpanchenko lab11]$ tar -tf lab11.tar
prog2.sh
prog2.sh~
prog1
prog1.c
prog1.sh
```

Рис. 2.11: Создание архива

Рис. 2.12: Архив

3 Вывод

Я изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

4 Контрольные вопросы

1. Команда getopts используется для обработки опций командной строки в скриптах на языке Bash. Она позволяет скрипту распознавать и обрабатывать опции, переданные ему при запуске, и выполнять соответствующие действия.

2. Метасимволы используются в Bash для шаблонного поиска и замены файлов в командной строке, а также для генерации имен файлов. Например, символ звездочки (*) может заменять любое количество любых символов в имени файла.

3. В Bash есть несколько операторов управления действиями, таких как if, else, elif, case, for, while, until. Они используются для выполнения определенных действий в зависимости от условий, заданных в скрипте.

4. Для прерывания цикла в Bash можно использовать операторы break и continue. Оператор break прерывает выполнение цикла и переходит к следующей команде после цикла, а оператор continue прерывает текущую итерацию цикла и переходит к следующей итерации.

5. Команда false возвращает код ошибки в скрипте, что может быть полезно для тестирования и отладки. Команда true, напротив, всегда возвращает успешный код завершения, что может быть полезно, например, для создания бесконечных циклов.

6. Данная строка проверяет, существует ли файл man$s$/i.$s в текущей директории, где $s и $i - переменные, заданные в скрипте.

7. Конструкция while используется для повторения блока команд до тех пор, пока определенное условие истинно, а конструкция until - до тех пор, пока определенное условие ложно. Поэтому, если условие выполняется сразу, блок команд в while ни разу не выполнится, а в until - выполнится один раз.