Лабораторная работа №12 по предмету Операционные системы

Группа НПМбв-02-19

Нечаева Виктория Алексеевна

Содержание

# 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 с ключами:  
   – -iinputfile — прочитать данные из указанного файла;  
   – -ooutputfile — вывести данные в указанный файл;  
   – -pшаблон — указать шаблон для поиска;  
   – -C — различать большие и малые буквы;  
   – -n — выдавать номера строк.  
   а затем ищет в указанном файле нужные строки, определяемые ключом -p.
2. Написать 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 должен вызывать эту программу и, проанализировав с помощью команды ?, выдать сообщение о том, какое число было введено.
3. Написать 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 необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
4. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).

# Выполнение лабораторной работы

## Задача 1

Сверху задается getops с возможными аргументами при запуске программы в терминале. Внизу в блоках if проверяется соответствие условиям и grep ищет по формату pattern подходящие строки. В последнем блоке if делается то же самое, но вывод результата происходит в файл output.txt



Рисунок 1

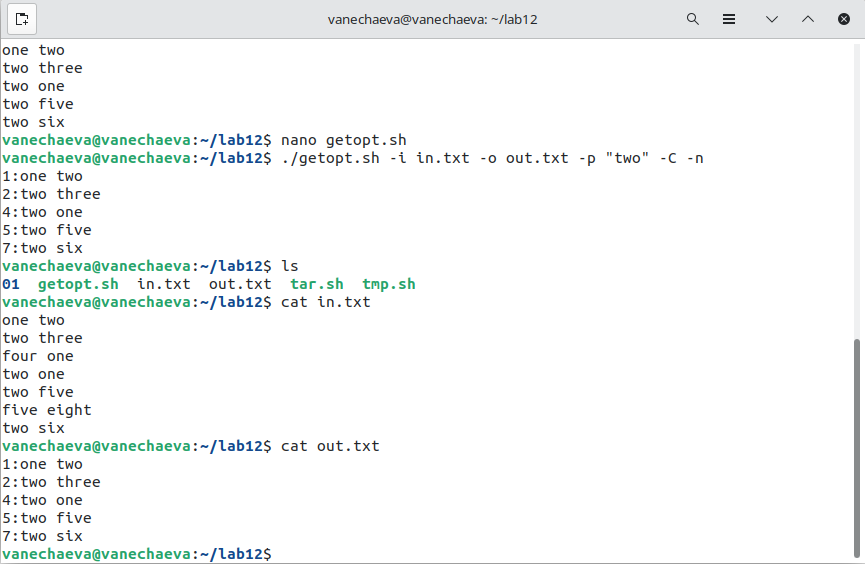


Рисунок 2

## Задача 2

Программа запрашивает ввод числа, считывает его, сравнивает согласно условиям и выводит результат. bash файл запускает программу на выполнение и echo & выводит num который мы задали в программе (берет последнее значение)

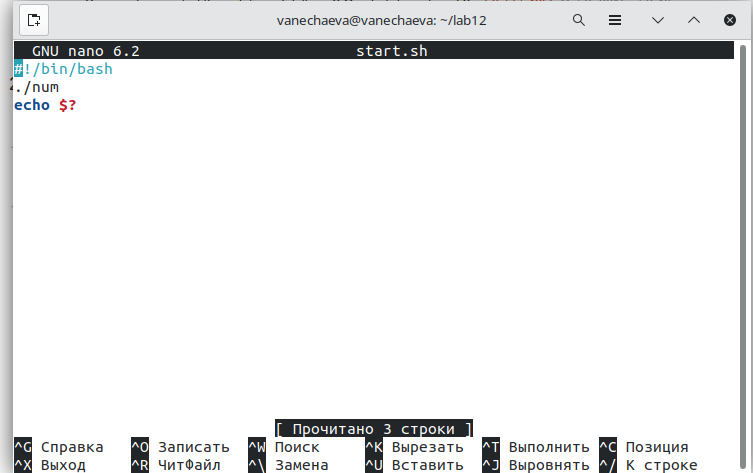


Рисунок 3



Рисунок 4

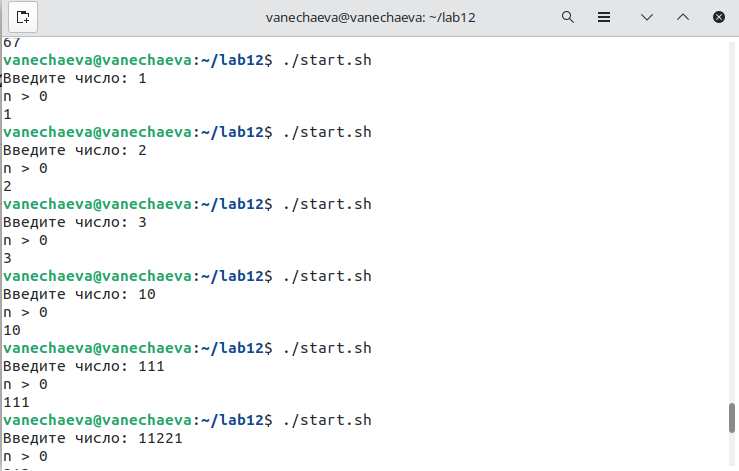


Рисунок 5

## Задача 3

В аргументе задается количество файлов, которое мы ходим создать. Цико проходит от 1 до i= 1 и создает такое количество файлов. Далее программа спрашивает пользователя, хочет ли он удалить созданные файлы, если пользователь отвечает “y”, то только что созданные файлы удаляются в цикле по одному, если “n” (или любую другую букву), то файлы не удаляются.



Рисунок 6

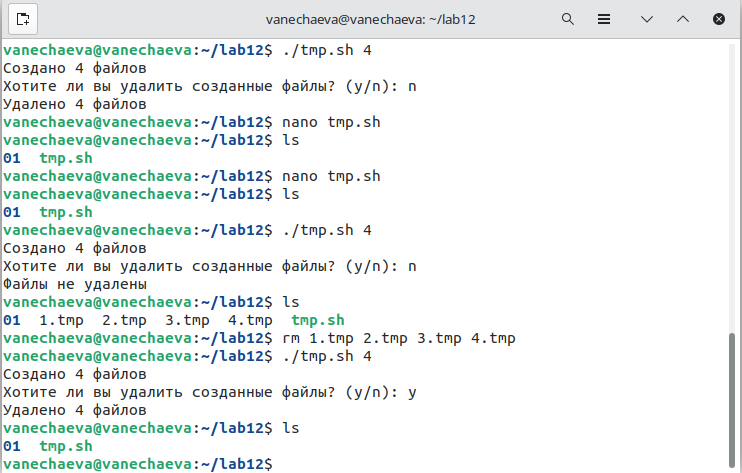


Рисунок 7

## Задача 4

Программа проверяет указан ли в строке вызова аргумент. Если нет, то программа завершается, если да, то создается .tar.gz

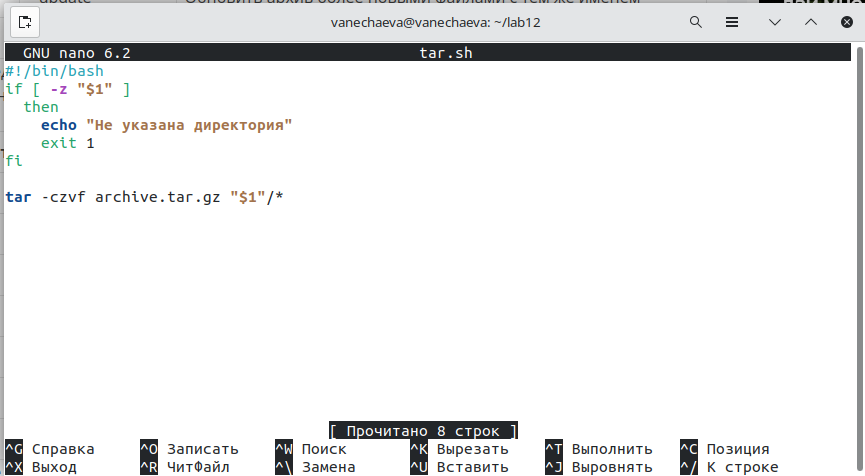


Рисунок 8

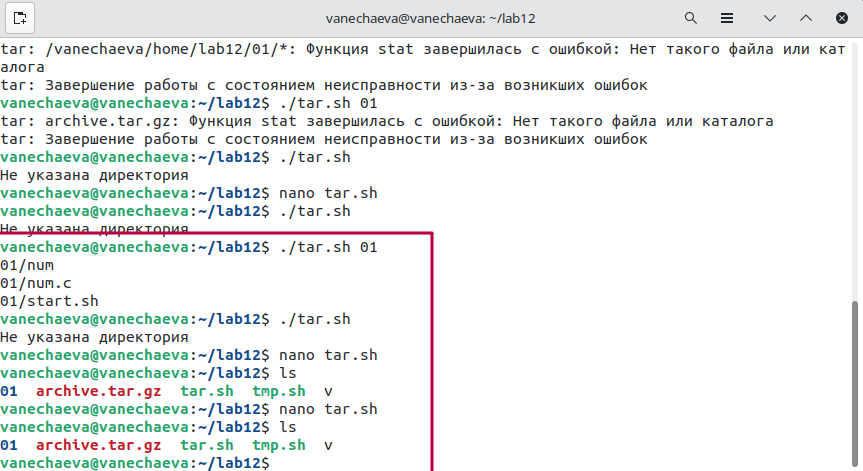


Рисунок 9

Аналогично, только еще если условие поиска по файлам, которые были изменены менее недели назад.

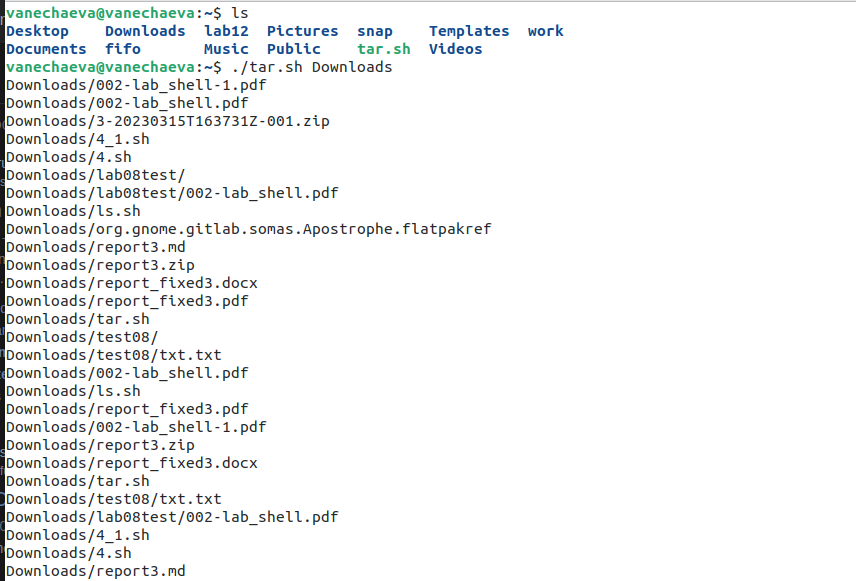


Рисунок 10



Рисунок 11

# Выводы

В ходе данной лабораторной работы мною были написаны более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Контрольные вопросы

1. Каково предназначение команды getopts?

Команда getopts в UNIX-подобных операционных системах используется для разбора аргументов командной строки. Она позволяет программистам написать скрипты, которые могут обрабатывать опции и параметры командной строки, переданные пользователем.

1. Какое отношение метасимволы имеют к генерации имён файлов?

Метасимволы в UNIX-подобных операционных системах используются для генерации имен файлов и путей к ним. Например, символ звездочки (\*) соответствует любой последовательности символов в имени файла, а символ вопросительного знака (?) соответствует одному любому символу. Также метасимволы используются для выполнения поиска и замены текста в файле при использовании утилит, таких как sed и awk.

1. Какие операторы управления действиями вы знаете?

* Операторы управления действиями в UNIX-подобных операционных системах включают в себя:
* if/else: позволяет выполнить определенное действие, если условие истинно, и другое действие, если условие ложно. for: используется для выполнения набора команд или действий для каждого элемента в списке или диапазоне значений. while: выполняет определенные действия, пока условие истинно. case: позволяет выполнить одно действие из нескольких в зависимости от значения переменной. break: используется для выхода из текущего цикла. continue: прерывает текущую итерацию цикла и переходит к следующей.

1. Какие операторы используются для прерывания цикла?

* Для прерывания цикла в UNIX-подобных операционных системах используются следующие операторы:
* break: используется для немедленного выхода из цикла. continue: используется для пропуска текущей итерации цикла и перехода к следующей.

1. Для чего нужны команды false и true?

Команды false и true являются базовыми утилитами в UNIX-подобных операционных системах, которые возвращают код возврата 1 и 0 соответственно. Обычно эти команды используются в комбинации с другими командами и операторами для контроля над процессами, запускаемыми в скриптах.

1. Что означает строка if test -f mani.s, встреченная в командном файле?

Данная строка используется для проверки наличия файла в определенном месте в файловой системе. Она проверяет, существует ли файл с именем mani.$s, где $s, $i - переменные, заданные ранее в скрипте.

1. Объясните различия между конструкциями while и until

Конструкция while используется для выполнения блока кода, пока условие истинно. В то время как конструкция until используется для выполнения блока кода до тех пор, пока условие не станет истинным. Другими словами, while выполняет блок кода, пока условие не станет ложным, а until выполняет блок кода, пока условие не станет истинным.