Лабораторная работа 13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Неустроева Ирина Николаевна

Содержание

# 1 Цель работы

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

# 2 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.

1. 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.
2. 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).
3. 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).

# 3 Выполнение лабораторной работы

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.

Создадим файл для скрипта, введём в него код (предварительно присвоив ему право на исполнение с помощью команды chmod) (рис. 1).

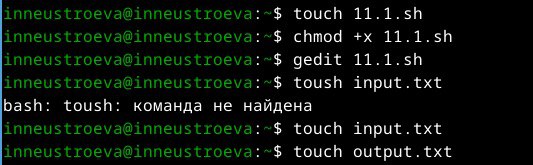


Рис. 1: Создание файла

Вызовем файл на исполнение (рис. 2).

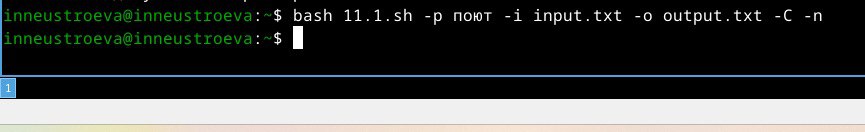


Рис. 2: Исполнение

Прилагаю скрипт (рис. 3).

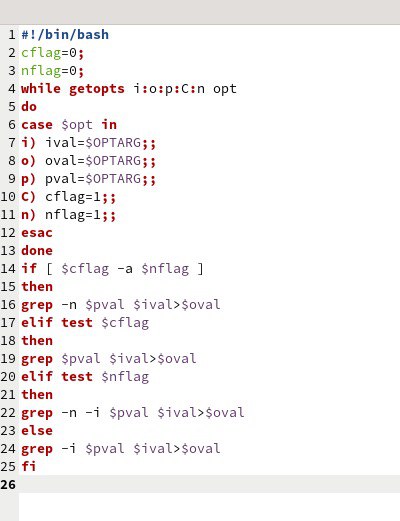


Рис. 3: Скрипт

Проверим работу данного файла (рис. 4). (рис. 5).

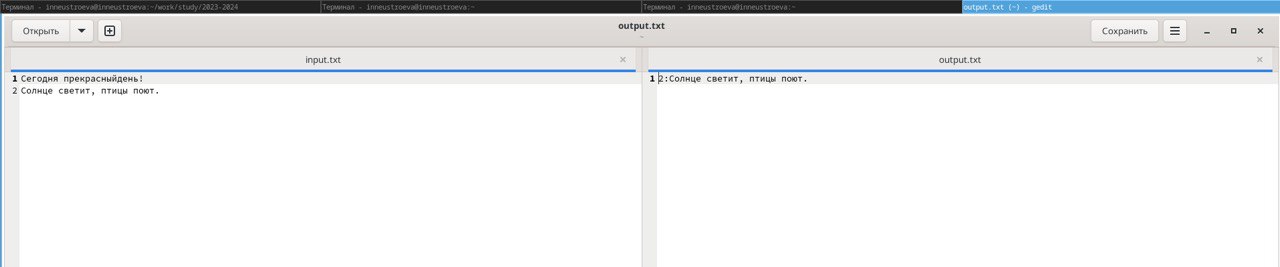


Рис. 4: Проверка работы

1. 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. (рис. 5).

Создаём файл для скрипта и файл для программы на языке Си, присваиваем командному файлу право на исполнение, вызываем его.Далее пишем программу на языке Си

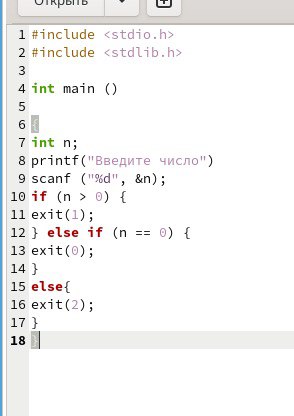


Рис. 5: Скрипт Си

Теперь пишем скрипт в командном файле. (рис. 6).

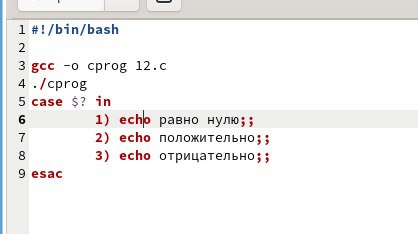


Рис. 6: Скрипт

* Проверим работу данного файла (рис. 7).

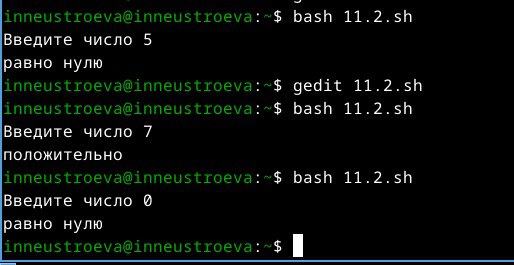


Рис. 7: Проверка работы

1. 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).

Для начала создаём командный файл, присваиваем ему право на исполнение. Проверяем, корректно ли отработал код. В домашнеё папке должны создаться 7 файла при вызове командного файла на исполнение и передаче ему 7 в качестве аргумента (рис. 8).

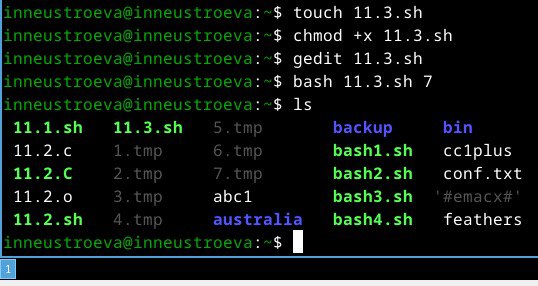


Рис. 8: Создание 7 файлов

Пишем скрипт (рис. 9).



Рис. 9: Скрипт

Теперь проверим, удалит ли файлы наш скрипт (при вызове командного файла на исполнение и передаче ему 2 в качестве аргумента созданные только что файлы должны быть удалены)(рис. 10).

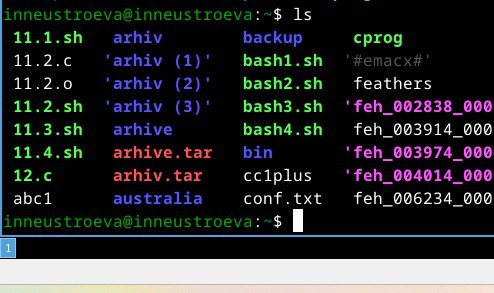


Рис. 10: Проверка

1. 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).

Пишем скрипт (рис. 11).

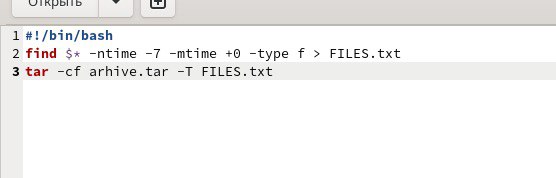


Рис. 11: Скрипт

Проверяем его работу (рис. 12).

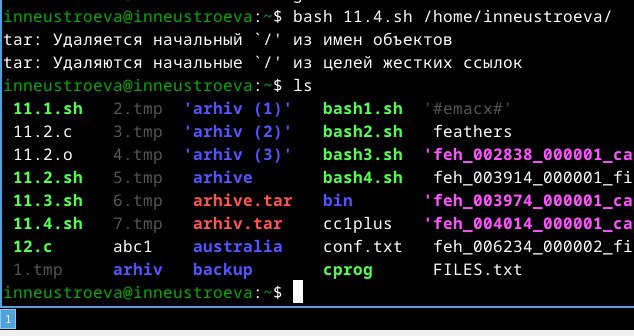


Рис. 12: Проверка работы

# 4 Вывод

В данной работе мы изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.