Отчёт по лабораторной работе №10:

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Касакьянц Владислав Сергеевич

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:

* -i - inputfile — прочитать данные из указанного файла;
* -o - outputfile — вывести данные в указанный файл;
* -p - шаблон — 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 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, напишем командный файл, который анализирует командную строку с ключами (-i, -o, -p,-c, -n), а затем ищет в указанном файле нужные строки, определяемые ключом -p (рис. 1):

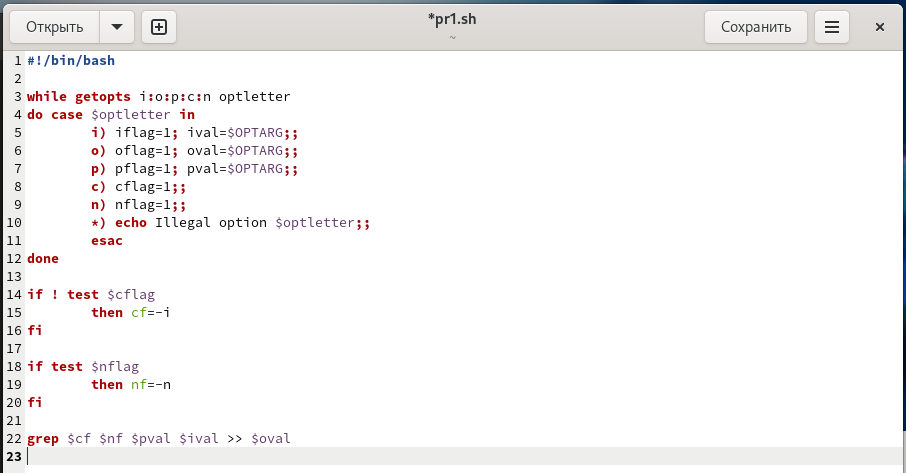


Рис. 1: Командный файл №1

Создадим один текстовый файл со стихотворением “input.txt” и файл, в который будет записываться результат “output.txt” (рис. 2), (рис. 3).

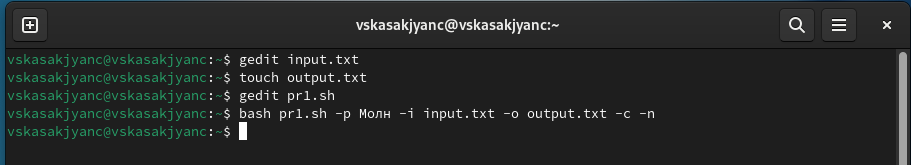


Рис. 2: Создание нужных файлов

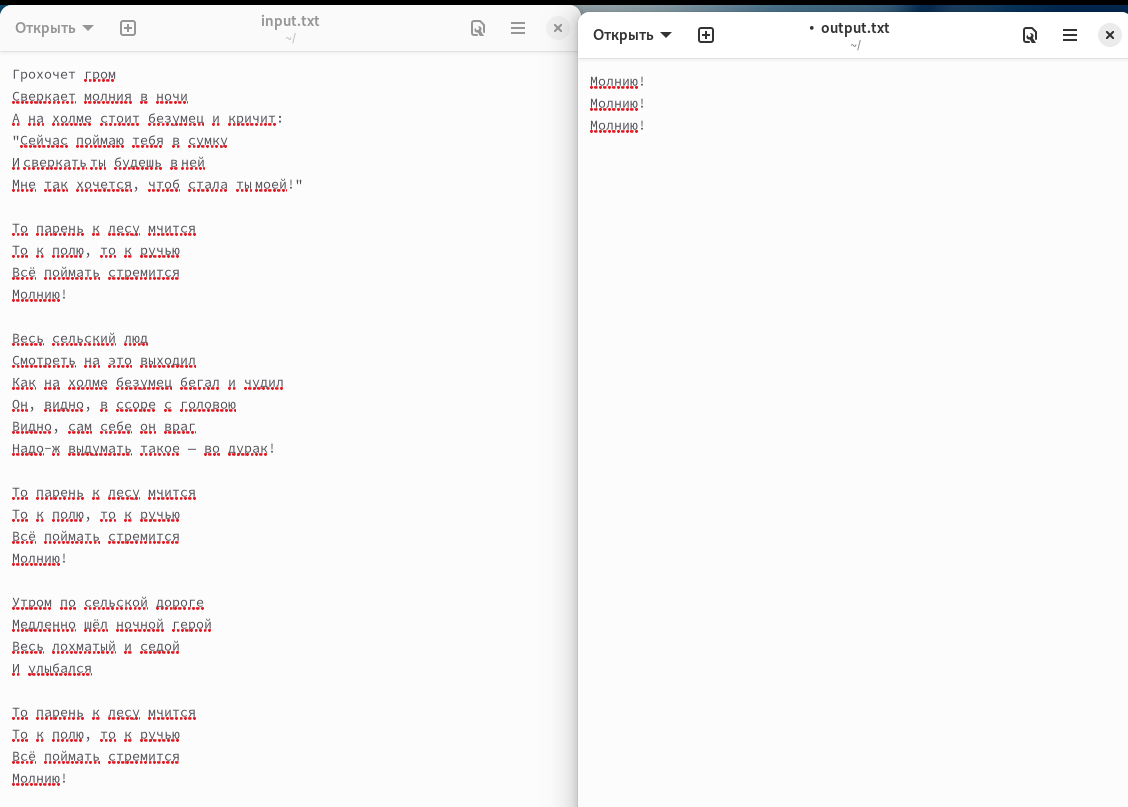


Рис. 3: Результат выполнения командного файла №1

1. Напишем на языке Си программу (рис. 4), 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 о коде завершения в оболочку. Командный файл (рис. 5) должен вызывать эту программу и, проанализировав с помощью команды $?, выдать сообщение о том, какое число было введено (рис. 6):

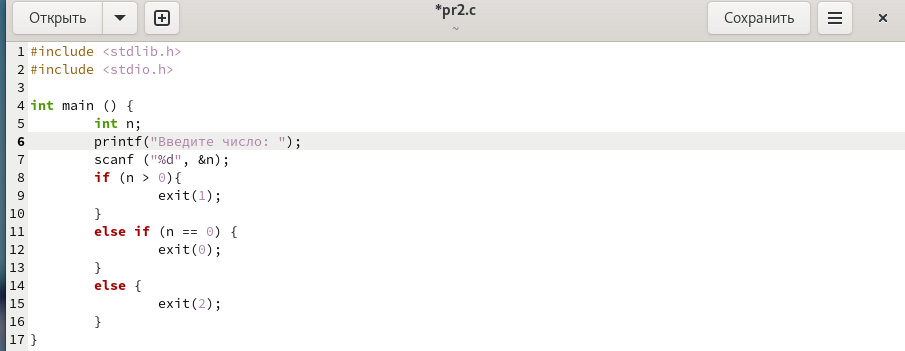


Рис. 4: Код на СИ

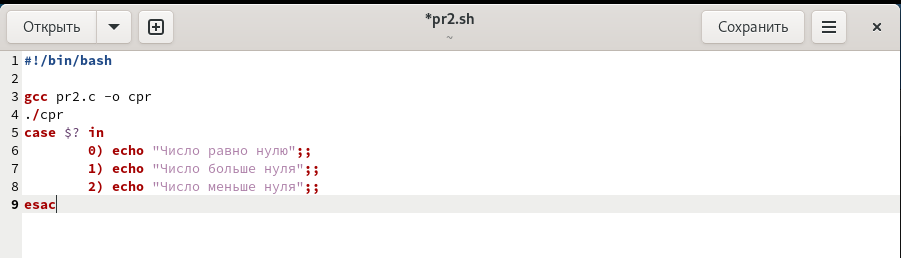


Рис. 5: Командный файл

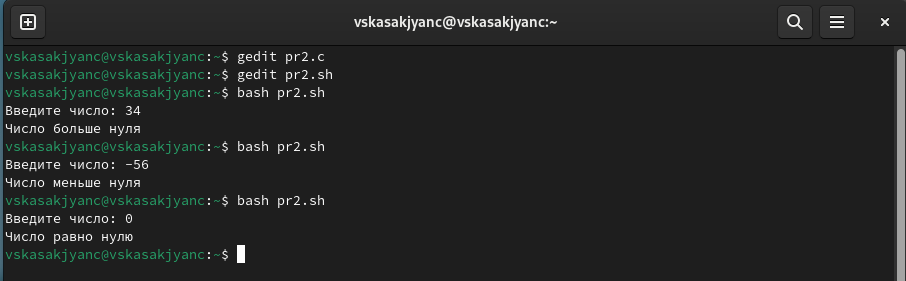


Рис. 6: Результат выполнения командного файла №2

1. Напишем командный файл (рис. 7), 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 (рис. 8):

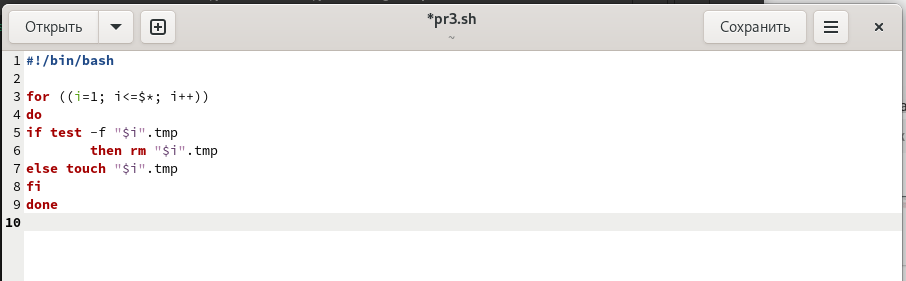


Рис. 7: Командный файл №3

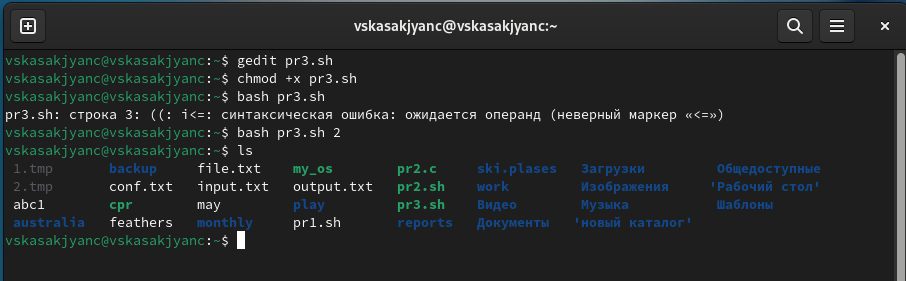


Рис. 8: Результат выполнения командного файла №3

1. Напишем командный файл (рис. 9), 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, использовав команду find (рис. 10):

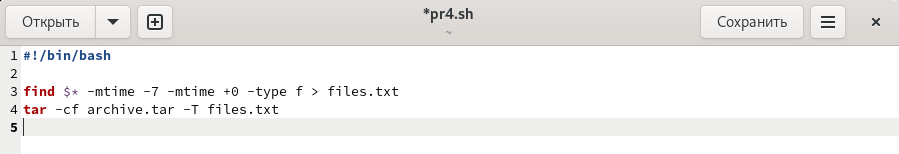


Рис. 9: Командный файл №4

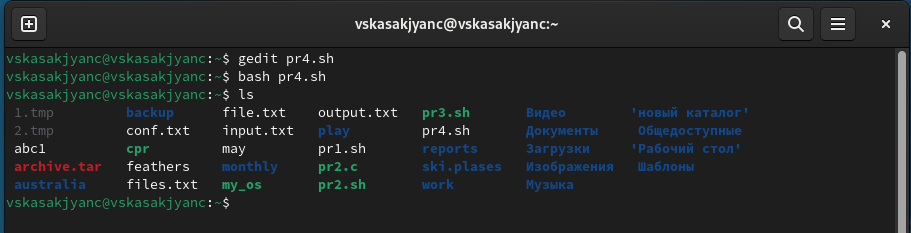


Рис. 10: Результат выполнения командного файла №4

# 4 Контрольные вопросы

1. **Каково предназначение команды getopts?**

Команда getopts используется для обработки аргументов командной строки. Она позволяет извлекать опции и их значения из списка аргументов.

1. **Какое отношение метасимволы имеют к генерации имён файлов?**

Метасимволы используются в генерации имён файлов для сопоставления шаблонов. Например, звездочка (\*) сопоставляет любое количество символов, а знак вопроса (?) сопоставляет любой один символ.

1. **Какие операторы управления действиями вы знаете?**

Операторы управления действиями используются для изменения потока выполнения скрипта. Вот некоторые из наиболее распространенных операторов управления действиями:

* **if…then…else:** Выполняет блок кода, если условие истинно. Если условие ложно, выполняется блок кода else (необязательно).
* **case…esac:** Выполняет блок кода в зависимости от значения переменной.
* **for…do…done:** Выполняет блок кода для каждого элемента в списке.
* **while…do…done:** Выполняет блок кода, пока условие истинно.
* **until…do…done:** Выполняет блок кода, пока условие ложно.

1. **Какие операторы используются для прерывания цикла?**

* **break:** Немедленно выходит из цикла.
* **continue:** Переходит к следующей итерации цикла, пропуская оставшиеся операторы в текущей итерации.

1. **Для чего нужны команды false и true?**

Команды false и true используются для возврата кода выхода, указывающего на успех (true) или неудачу (false).

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

Эта строка проверяет, существует ли файл с именем mani.$s. Если файл существует, выполняется оператор then.

1. **Объясните различия между конструкциями while и until.**

* **while:** Выполняет блок кода, пока условие истинно.
* **until:** Выполняет блок кода, пока условие ложно.

# 5 Выводы

В данной лабораторной работе мы изучили основы программирования в оболочке ОС UNIX, а также научилимь писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Список литературы