Отчёт по лабораторной работе №12

Операционные системы

Ильина Любовь Александровна

Содержание

# Цель работы

Изучение основ программирования в оболочке ОС UNIX. Научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Выполнение лабораторной работы

1. Используя команды getopts grep, напишем командный файл, который анализирует командную строку с ключами (рис. [1](#fig:001) - [2](#fig:002)): – -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.

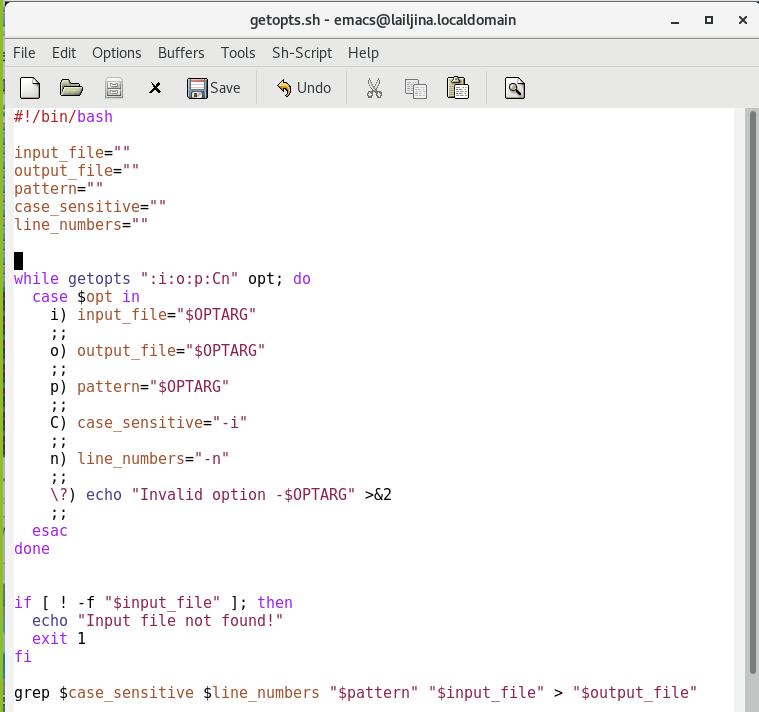


Figure 1: Командный файл

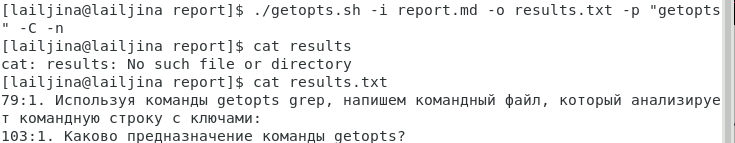


Figure 2: Проверка работы файла

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. [3](#fig:003) - [5](#fig:005))

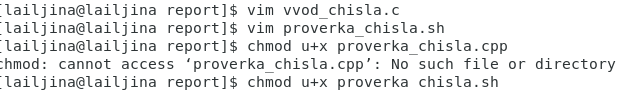


Figure 3: Создание файлов

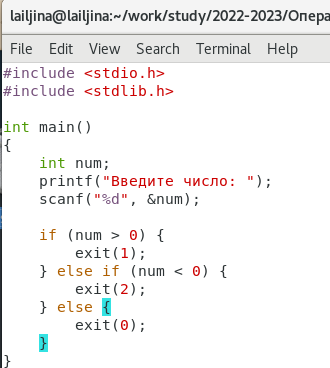


Figure 4: Код программы

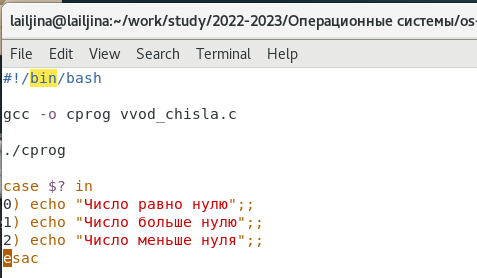


Figure 5: Командный файл

Проверяю работу файлов (рис. [6](#fig:006))

Figure 6: Проверка работы файла

Figure 6: Проверка работы файла

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). (рис. [7](#fig:007) - [8](#fig:008))

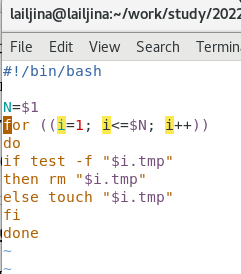


Figure 7: Командный файл

Проверяю работу файлов (рис. [8](#fig:008))

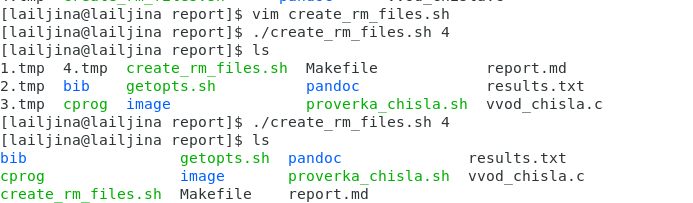


Figure 8: Проверка работы файла

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). (рис. [9](#fig:009) - [10](#fig:010))

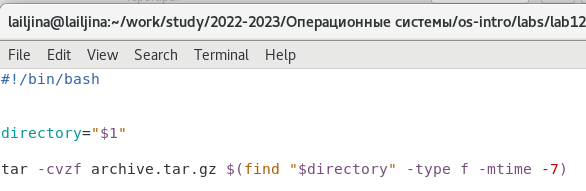


Figure 9: Командный файл

Проверяю работу файлов (рис. [10](#fig:010))

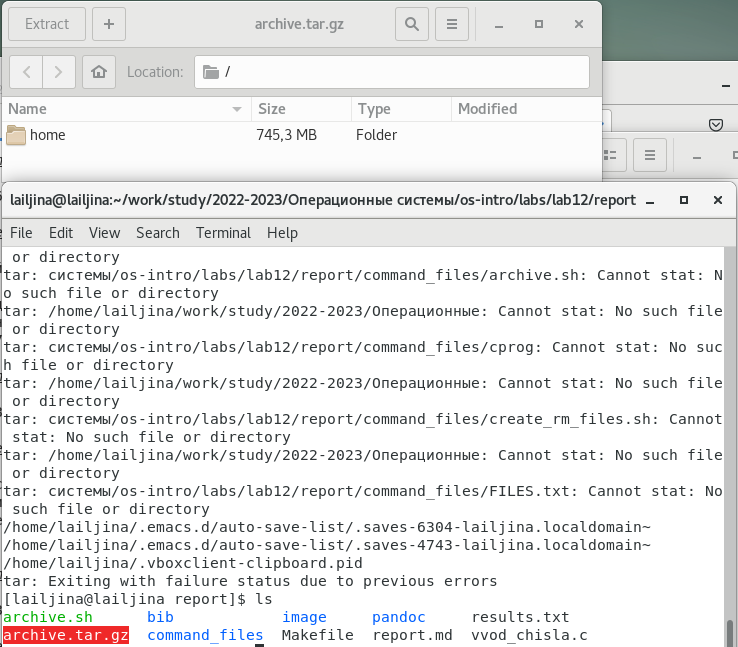


Figure 10: Проверка работы файла

Контрольные вопросы

1. Каково предназначение команды getopts? getopts осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных. Функция getopts также понимает переменные типа массив, следовательно, можно использовать её в функции не только для синтаксического анализа аргументов функций, но и для анализа введённых пользователем данных.
2. Какое отношение метасимволы имеют к генерации имён файлов? Метасимволы позволяют генерировать имена файлов на основе заданного шаблона.
3. Какие операторы управления действиями вы знаете?

* test -f file - возвращает нулевой код завершения (истина), если файл file существует, и ненулевой код завершения (ложь) в противном случае: – test s –— истина, если аргумент s имеет значение истина; – test -f file — истина, если файл file существует; – test -i file — истина, если файл file доступен по чтению; – test -w file — истина, если файл file доступен по записи; – test -e file — истина, если файл file — исполняемая программа; – test -d file — истина, если файл file является каталогом.

1. Какие операторы используются для прерывания цикла? Команда break завершает выполнение цикла, а команда continue завершает данную итерацию блока операторов.
2. Для чего нужны команды false и true? команда true, которая всегда возвращает код завершения, равный нулю (т.е. истина), и команда false, которая всегда возвращает код завершения, не равный нулю (т. е. ложь).
3. Что означает строка if test -f mani.$s, встреченная в командном файле? проверяет, существует ли файл с именем man<значение переменной $s>/<значение переменной $i>.<значение переменной $s> в текущем каталоге
4. Объясните различия между конструкциями while и until. При замене в операторе цикла while служебного слова while на until условие, при выполнении которого осуществляется выход из цикла, меняется на противоположное. В остальном оператор цикла while и оператор цикла until идентичны.

# Выводы

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