Операционные системы

Лабораторная работа №11

Матюшкин Денис Владимирович (НПИбд-02-21)

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 Ход работы

1. Используя команды getopts grep, напишем 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 (рис. 1). Проверим написанный командный файл (рис. 2).

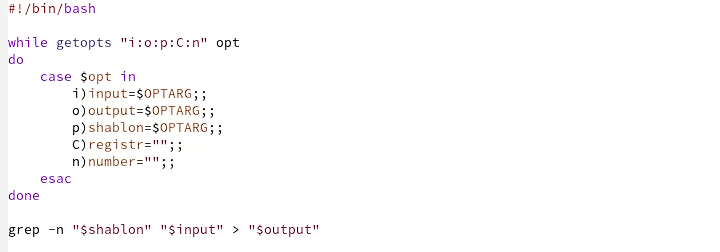


Рис. 1: Командный файл поиска файлов

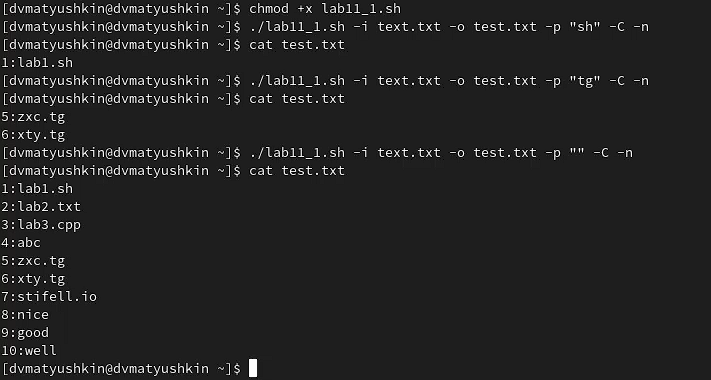


Рис. 2: Проверка файла

1. Напишем 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 вызывает эту программу и, проанализировав с помощью команды $?, выдает сообщение о том, какое число было введено (рис. 3 и рис. 4). Проверим написанный командный файл (рис. 5).

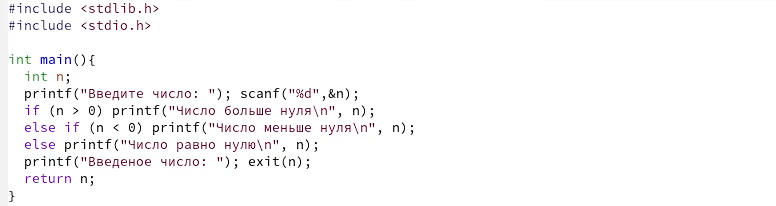


Рис. 3: Программа на Си



Рис. 4: Командный файл

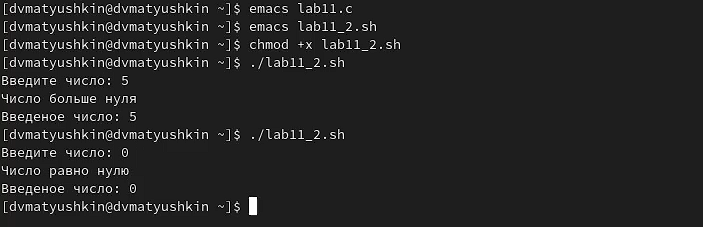


Рис. 5: Проверка файла

1. Напишем 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 создается, передаётся в аргументы командной строки. Этот же командный файл удаляет все созданные им файлы (если они существуют) (рис. 6). Проверим написанный командный файл (рис. 7).

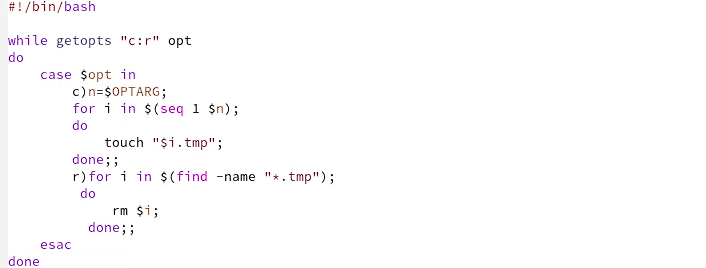


Рис. 6: Командный файл для создание файлов

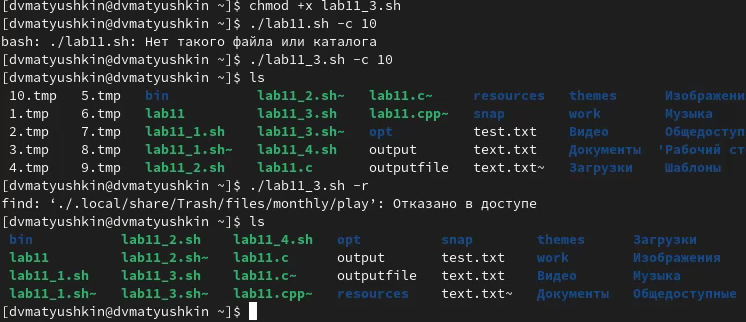


Рис. 7: Проверка файла

1. Напишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (используем команду find) (рис. 8). Проверим написанный командный файл (рис. 9 и рис. 10).

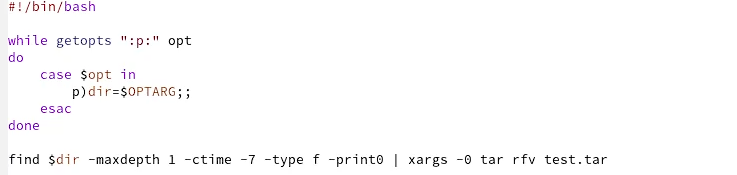


Рис. 8: Командный файл архиватор

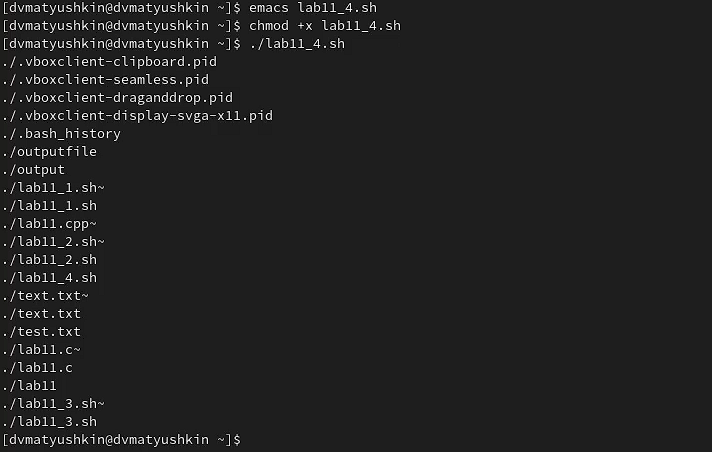


Рис. 9: Проверка файла

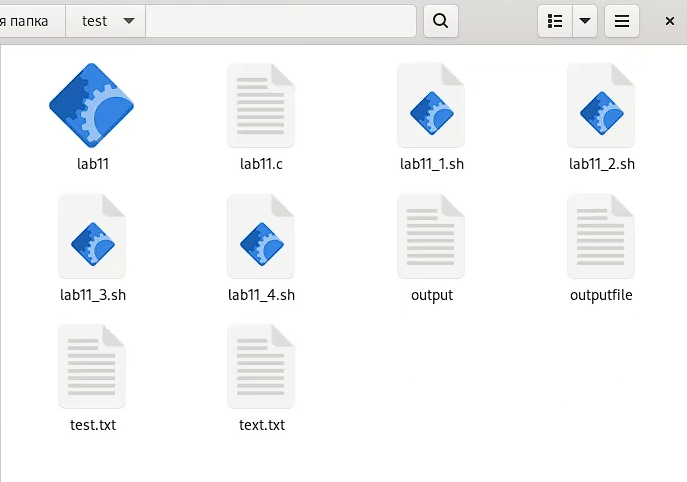


Рис. 10: Проверка файла

# 3 Контрольные вопросы

**1. Каково предназначение команды getopts?**  
Команда getopts используется для разбора параметров и проверки опций на допустимость. Осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.

**2. Какое отношение метасимволы имеют к генерации имён файлов?**  
Метасимволы отвечают за параметры выдачи файлов, а следовательно могут вносить иной смысл, нежели прямое значение как символа. Например:  
• \* — соответствует произвольной, в том числе и пустой строке;  
• ? — соответствует любому одному символу;  
• [c1-c1] — соответствует любому символу, лексикографически на ходящемуся между символами c1 и с2.  
• echo \* — выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды ls;  
• ls *.c — выведет все файлы с последними двумя символами, равными .c.*  
*• echo prog.? — выдаст все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются prog. .*  
*• [a-z]* — соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита.

**3. Какие операторы управления действиями вы знаете?**  
Ответ: for, break , while, until, case ,continue, if, else.

**4. Какие операторы используются для прерывания цикла?**  
Ответ: break

**5. Для чего нужны команды false и true?**  
Это логические значения (0 и 1 соответственно). Можно использовать как более понятный вариант.

1. Ответ: условие существование файла mani.$s

**7. Объясните различия между конструкциями while и until.**  
while - проверка условия затем выполнение тела цикла.  
until - выполнение тела цикла затем проверка условия.

# 4 Вывод

* В ходе этой лабораторной работы мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.