Лабораторная работа №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Лебедева Ольга Андреевна

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 Теоретическое введение

Цикл for - это оператор языка программирования bash, который позволяет многократно выполнять код. Цикл for классифицируется как оператор итерации, то есть это повторение процесса в сценарии bash. Например, вы можете запустить команду или задачу UNIX 5 раз или прочитать и обработать список файлов с помощью цикла for.

Цикл while используется для выполняет заданный набор команд неизвестное число раз до тех пор , как данное условие принимает значение истинно.

1. Используя команды getopts grep, написали командный файл, который анализирует командную строку с ключами: (рис. 1)

– -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.

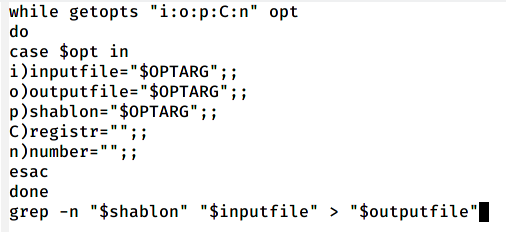


Рис. 1: Код скрипта1

Сделаем файл исполняемым. (рис. 2)

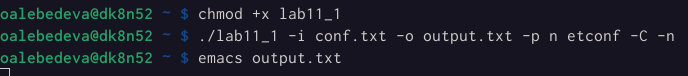


Рис. 2: Исполняем файл

Проверим результат работы программы. (рис. 3)

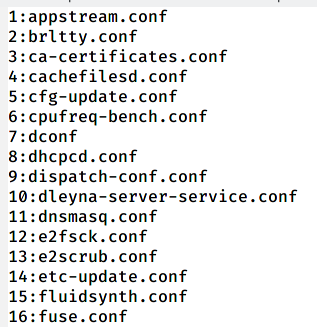


Рис. 3: Результат работы

1. Написали на языке 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. (рис. 4) (рис. 5)

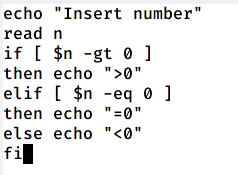


Рис. 4: Код скрипта2

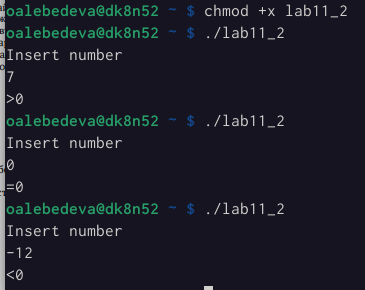


Рис. 5: Результат работы

1. Написали 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. 6)

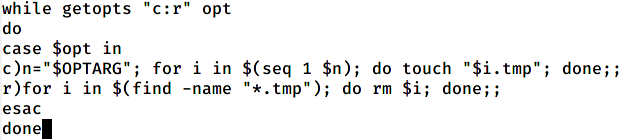


Рис. 6: Код скрипта3

Задали программе создать 4 файла tmp. (рис. 7)

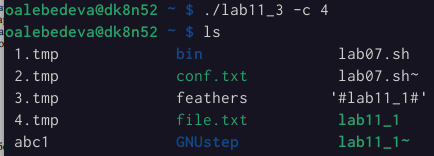


Рис. 7: Результат работы

Теперь одной командой удалим созданные файлы. (рис. 8)

Рис. 8: Команда удаления

Рис. 8: Команда удаления

Проверяем содержание домашнего каталога, файлы были удалены. (рис. 9)

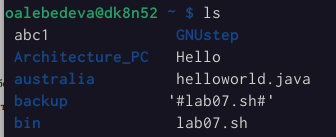


Рис. 9: Результат работы

1. Написали 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. 10)

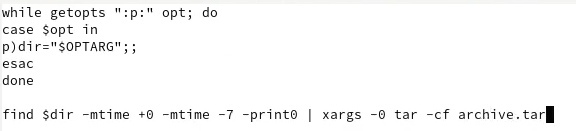


Рис. 10: Код скрипта4

Файлы, которые были изменены менее недели тому назад: (рис. 11)



Рис. 11: Результат работы

# 3 Вывод

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

# 4 Контрольные вопросы

1. Каково предназначение команды getopts?

Команда getopts выводит сообщение об ошибке в стандартный протокол, когда встречается опция, не включенная в цепочку\_опций.

1. Какое отношение метасимволы имеют к генерации имён файлов?

При генерации имен используют метасимволы:

? один произвольный символ;

[…] любой из символов, указанных в скобках перечислением и/или с указанием диапазона;

cat f\* выдаст все файлы каталога, начинающиеся с “f”;

cat *f* выдаст все файлы, содержащие “f”;

cat program.? выдаст файлы данного каталога с однобуквенными расширениями, скажем “program.c” и “program.o”, но не выдаст “program.com”;

cat [a-d]\* выдаст файлы, которые начинаются с “a”, “b”, “c”, “d”. Аналогичный эффект дадут и команды “cat [abcd]*” и ”cat [bdac]*”.

1. Какие операторы управления действиями вы знаете?

Часто бывает необходимо обеспечить проведение каких-либо действий циклически и управление дальнейшими действиями в зависимости от результатов проверки некоторого условия. Для решения подобных задач язык программирования bash предоставляет Вам возможность использовать такие управляющие конструкции, как for, case, if и while. С точки зрения командного процессора эти управляющие конструкции являются обычными командами и могут использоваться как при создании командных файлов, так и при работе в интерактивном режиме.

1. Какие операторы используются для прерывания цикла?

Оператор break используется для выхода из текущего цикла. Оператор continue используется для выхода из текущей итерации цикла и начала следующей итерации.

1. Для чего нужны команды false и true?

true: - всегда возвращает 0 в качестве кода выхода. false - всегда возвращает 1 в качестве кода выхода.

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

Введенная строка означает условие существования файла mani.$s

1. Объясните различия между конструкциями while и until.

Цикл while выполняет тело цикла пока условие истинно.

Цикл until выполняет тело цикла пока условие ложно. Другими словами цикл until выполняется до тех пор пока условие не станет истинным.