отчёта по лабораторной работе №12

Программирование в командномпроцессоре ОС UNIX. Ветвления и циклы

Джумаев Бегенч

Содержание

# Цель работы

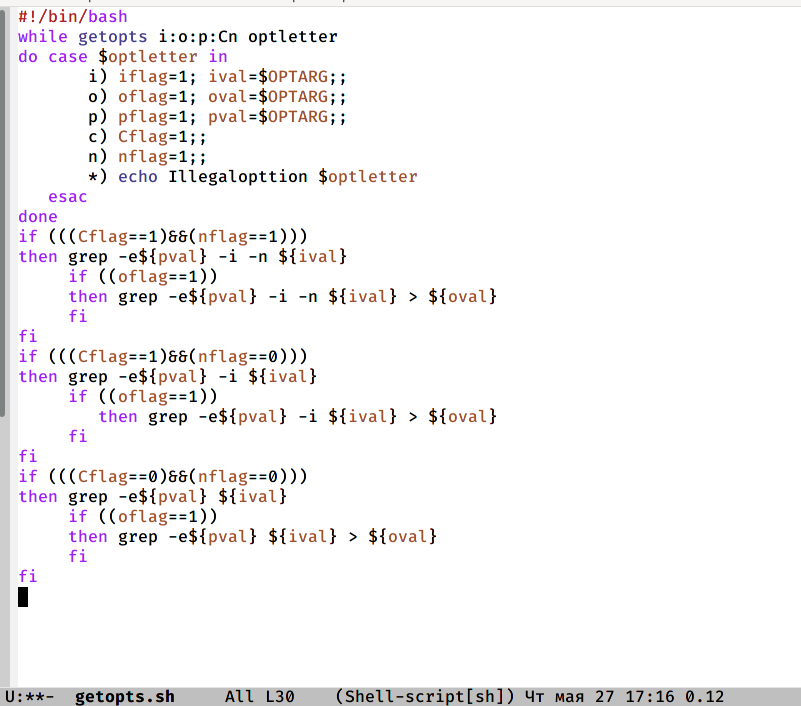
Изучить основы программирования в оболочке ОС UNIX. Научится писать бо-лее сложные командные файлы с использованием логических управляющих кон-струкций и циклов.

# Задание

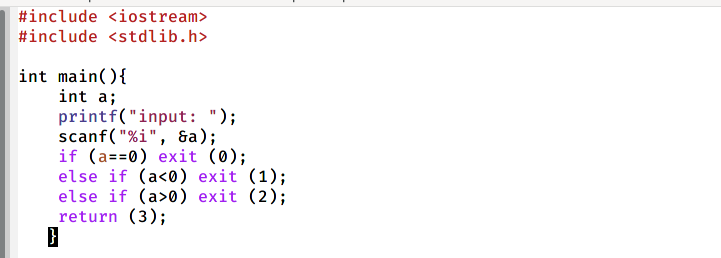
1. Используя команды getopts grep, написать командный файл, который анализирует командную строку с ключами: -iinputfile— прочитать данные из указанного файла; -ooutputfile— вывести данные в указанный файл; -pшаблон— указать шаблон для поиска; -C— различать большие и малые буквы; -n— выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом-p.
2. Написать на языке Си программу, которая вводит число и определяет, являетсяли оно больше нуля, меньше нуля или равно нулю. Затем программа завершаетсяс помощью функцииexit(n), передавая информацию в о коде завершения воболочку. Командный файл должен вызывать эту программу и, проанализировавс помощью команды$?, выдать сообщение о том, какое число было введено.
3. Написать командный файл, создающий указанное число файлов, пронумерован-ных последовательно от1доN(например1.tmp,2.tmp,3.tmp,4.tmpи т.д.).Число файлов, которые необходимо создать, передаётся в аргументы команднойстроки. Этот же командный файл должен уметь удалять все созданные им файлы(если они существуют).
4. Написать командный файл, который с помощью командыtarзапаковывает вархив все файлы в указанной директории. Модифицировать его так, чтобы запа-ковывались только те файлы, которые были изменены менее недели тому назад(использовать командуfind).

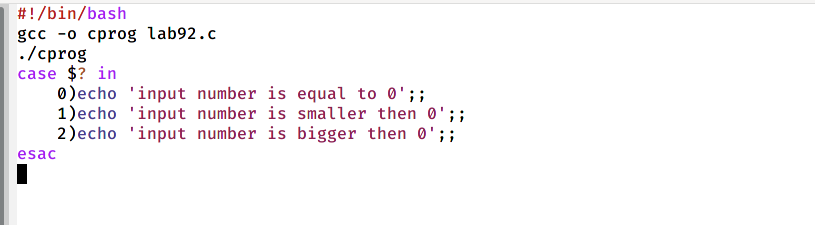
# Выполнение лабораторной работы

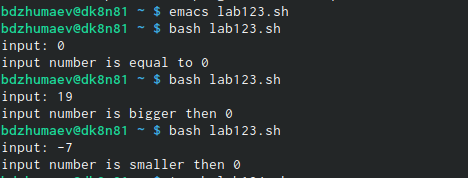
1. Используя команды getopts grep, написал командный файл, который анализирует командную строку с ключами: -iinputfile— прочитать данные из указанного файла; -ooutputfile— вывести данные в указанный файл; -pшаблон— указать шаблон для поиска; -C— различать большие и малые буквы; -n— выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом-p.



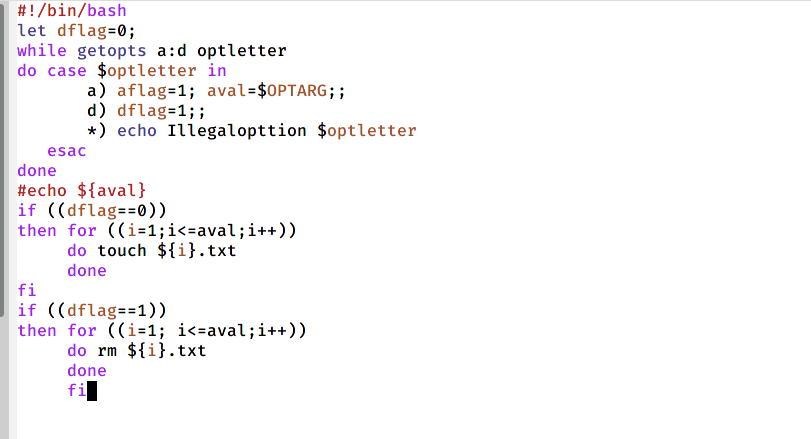
1. Написал на языке Си программу, которая вводит число и определяет, являетсяли оно больше нуля, меньше нуля или равно нулю. Затем программа завершаетсяс помощью функцииexit(n), передавая информацию в о коде завершения воболочку. Командный файл должен вызывать эту программу и, проанализировавс помощью команды $?, выдать сообщение о том, какое число было введено.

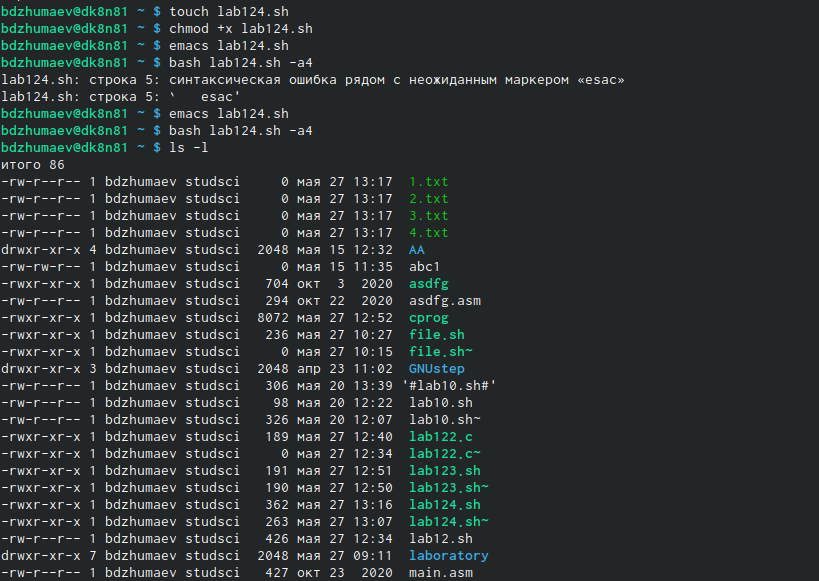


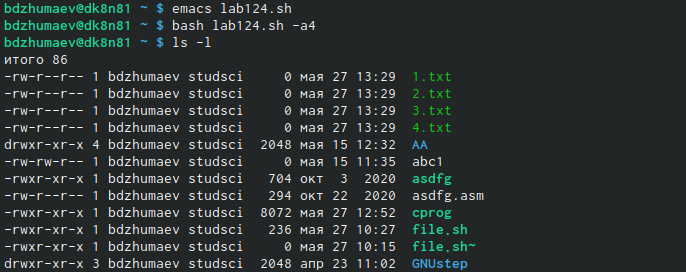




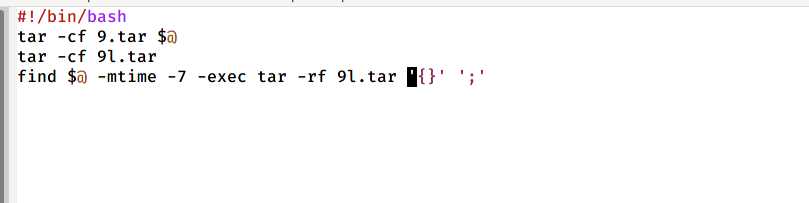
1. Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от1до N (например1.tmp,2.tmp,3.tmp,4.tmpи т.д.).Число файлов, которые необходимо создал, передаётся в аргументы команднойстроки. Этот же командный файл должен уметь удалять все созданные им файлы

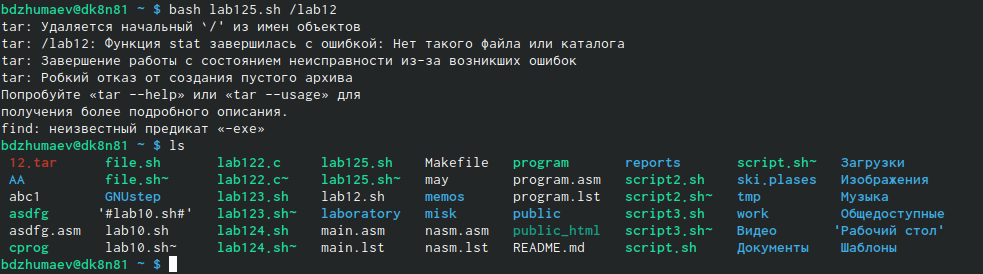






1. Написал командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировал его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовала команду find).





# Выводы

Изучал основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Контрольные вопросы

1. Команда getopts является встроенной командой командной оболочки bash, предназначенной для разбора параметров сценариев. Она обрабатывает исключительно однобуквенные параметры как с аргументами, так и без них и этого вполне достаточно для передачи сценариям любых входных данных.
2. При генерации имен используют метасимволы:

* произвольная (возможно пустая) последовательность символов;

? 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. Операторы && и || являются управляющими операторами. Если в командной строке стоит command1 && command2, то command2 выполняется в том, и только в том случае, если статус выхода из команды command1 равен нулю, что говорит об успешном ее завершении. Аналогично, если командная строка имеет вид command1 || command2, то команда command2 выполняется тогда, и только тогда, когда статус выхода из команды command1 отличен от нуля.
2. Оператор break завершает выполнение ближайшего включающего цикла или условного оператора, в котором он отображается.
3. Команда true всегда возвращает ноль в качестве выходного статуса для индикации успеха. Команда false всегда возвращает не-ноль в качестве выходного статуса для индикации неудачи. Во всех управляющих конструкциях в качестве логического значения используется код возврата из программы, указанной в качестве условия. Код возврата 0 – истина, любое другое значение – ложь. Программа true – всегда завершается с кодом 0, false – всегда завершается с кодом 1.
4. Введенная строка означает условие существования файла mani.$s.
5. Цикл While выполняется до тех пор, пока указанное в нем условие истинно. Когда указанное условие становится ложным - цикл завершается. Цикл Until выполняется до тех пор, пока указанное в нем условие ложно.