Программирование в командномпроцессоре ОС UNIX. Командные файлы

Отчет по лабораторной работе №12

Башкирова Я.Д

Содержание

# Теоретическое введение

Циклы позволяют выполнять один и тот же участок кода необходимое количество раз. В большинстве языков программирования существует несколько типов циклов. Большинство из них поддерживаются оболочкой Bash. Циклы бывают:

* for - позволяет перебрать все элементы из массива или использует переменную-счетчик для определения количества повторений;
* while - цикл выполняется пока условие истинно;
* until - цикл выполняется пока условие ложно.

Циклы Bash, это очень полезная вещь и разобраться с ними будет несложно. Bash позволяет использовать циклы как в скриптах, так и непосредственно в командной оболочке.

# Цель работы

Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

# 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, 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.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (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 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).

# Ход работы

1. Я, используя команды getopts grep, написала командный файл, который анализирует командную строку с ключами:

– -iinputfile — прочитать данные из указанного файла;

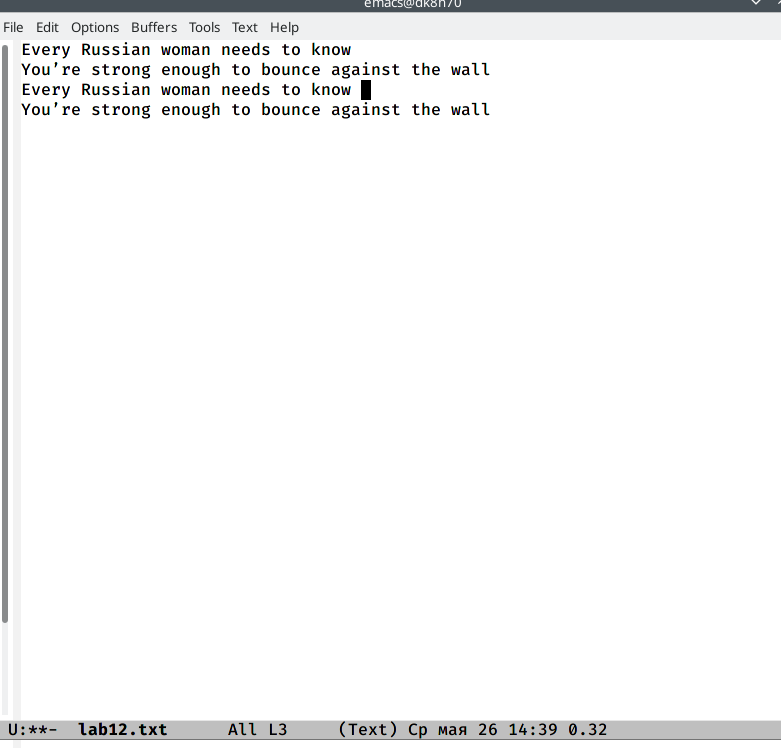
– -ooutputfile — вывести данные в указанный файл;

– -pшаблон — указать шаблон для поиска;

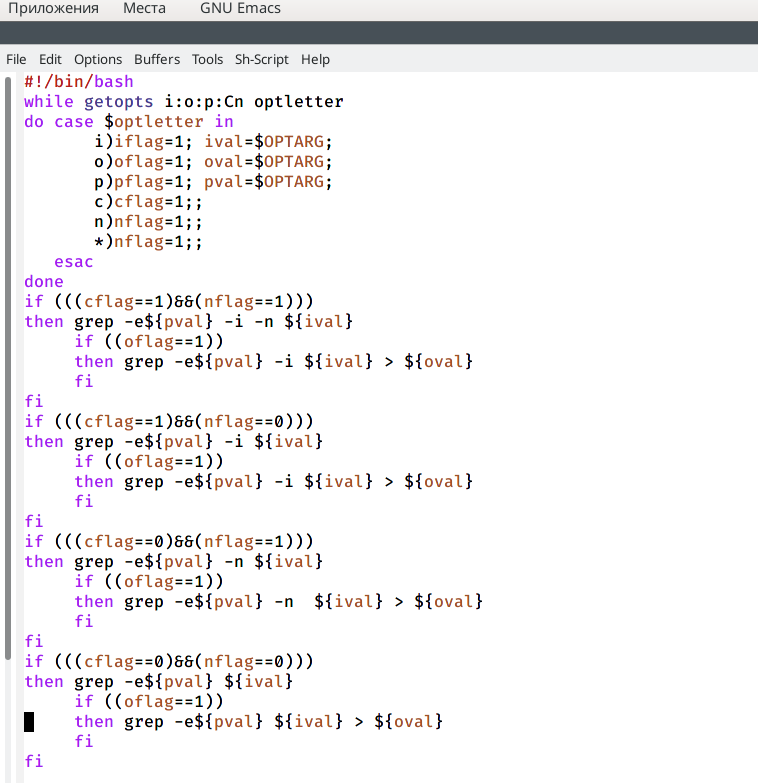
– -C — различать большие и малые буквы;

– -n — выдавать номера строк.

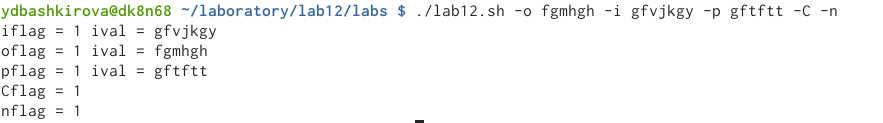
а затем ищет в указанном файле нужные строки, определяемые ключом -p.



Текст

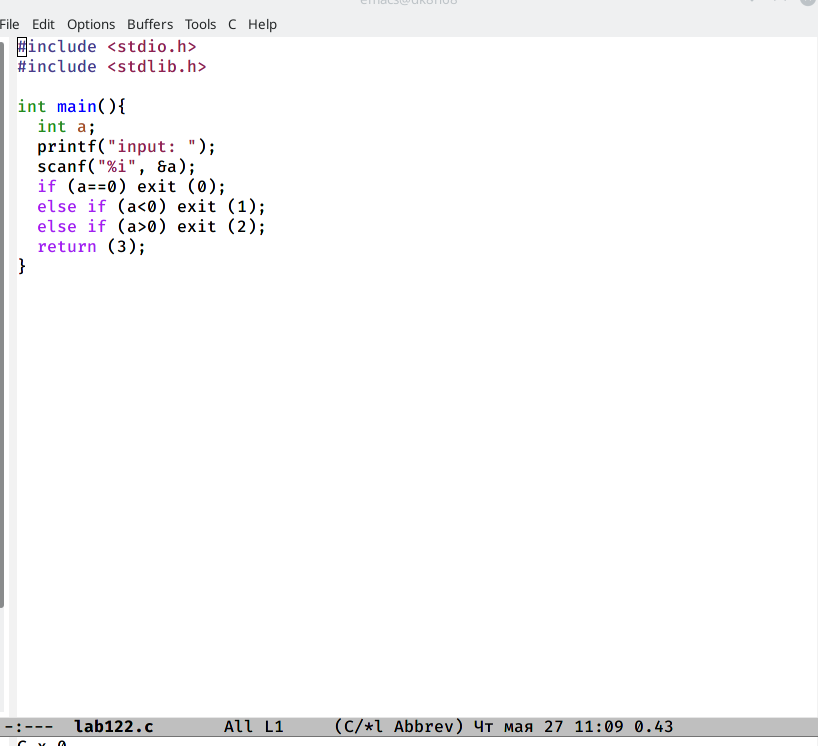


Командный файл,который анализирует командную строку

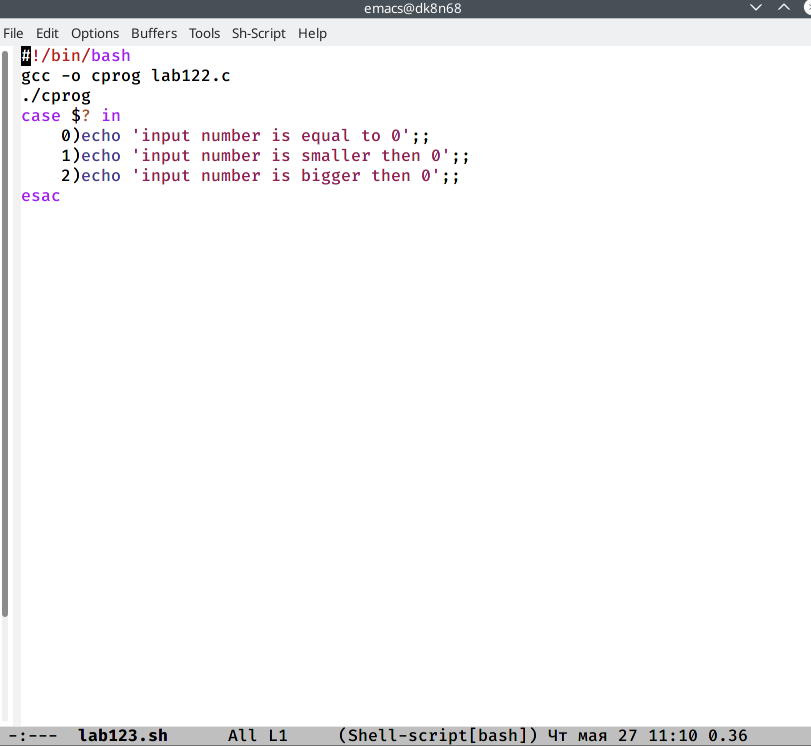


Анализ командной строки с ключами

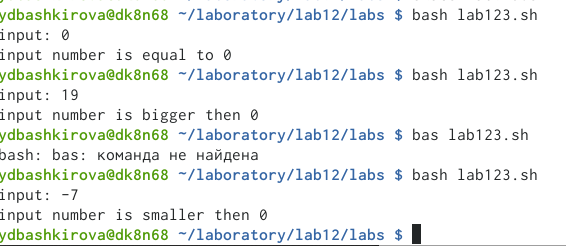
1. Я написала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции exit(n), передавая информацию о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды $?, выдать сообщение о том, какое число было введено.



Программа на языке Си

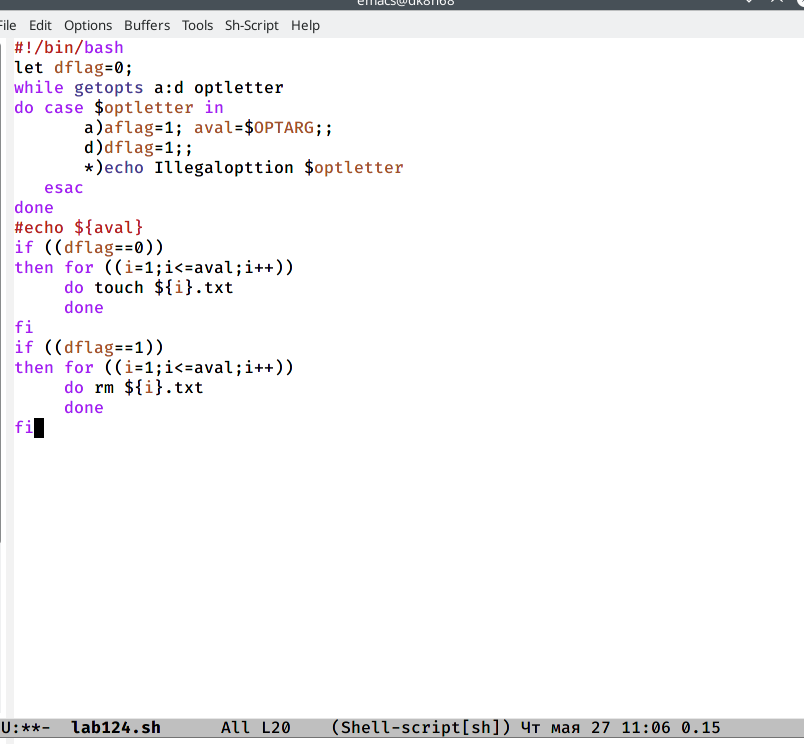


Командный файл

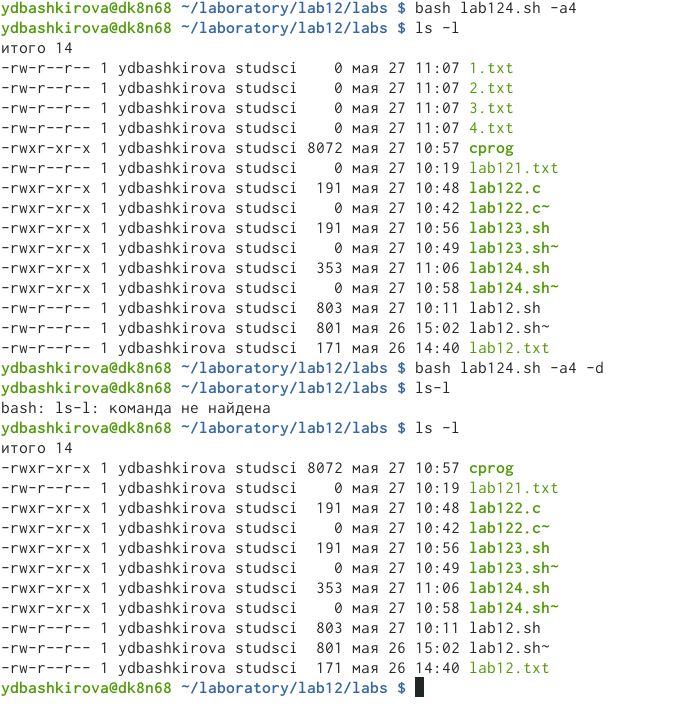


Ввод чисел

1. Я написала 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).

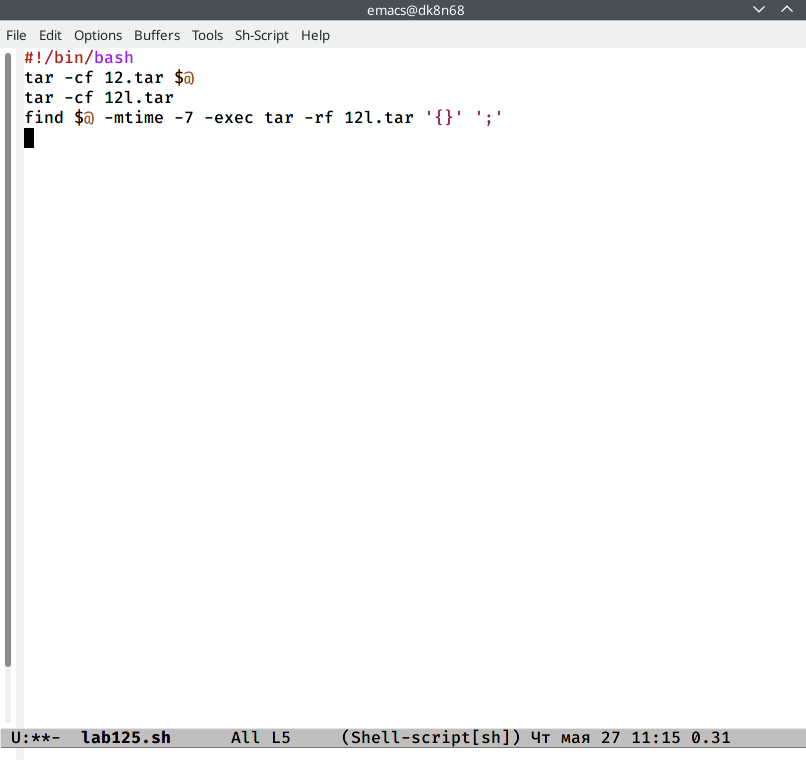


Командный файл,создающий указанное число файлов

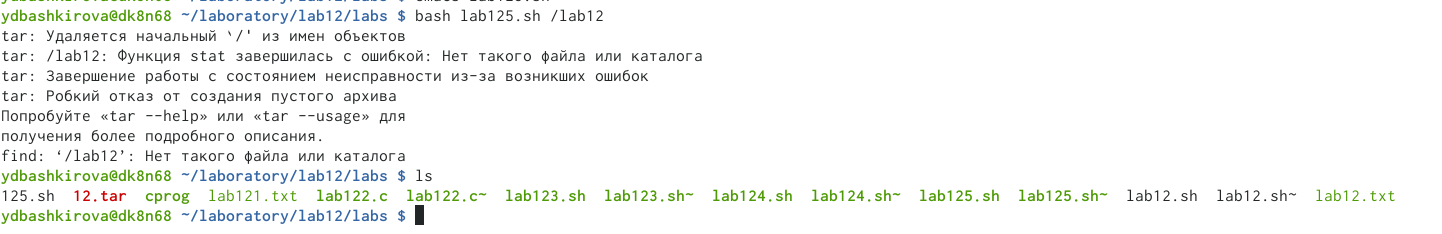


Число файлов от 1 до N и их удаление

1. Я написала 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).



Командный файл,запаковывающий в архив все файлы



Архив всех файлов в указанной директории

# Выводы

Изучила основы программирования в оболочке ОС UNIX/Linux. Научилась писать небольшие командные файлы.

# Библиография

https://esystem.rudn.ru/pluginfile.php/1142520/mod\_resource/content/3/009-lab\_shell\_prog\_2.pdf

https://losst.ru/tsikly-bash

# Контрольные вопросы

1. Команда getopts является встроенной командой командной оболочки bash, предназначенной для разбора параметров сценариев. Она обрабатываетисключительно 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 Untilвыполняется до тех пор, пока указанное в нем условие ложно.