Лабораторная работа №11

Ханина Людмила Константиновна

Table of Contents

# Цель работы

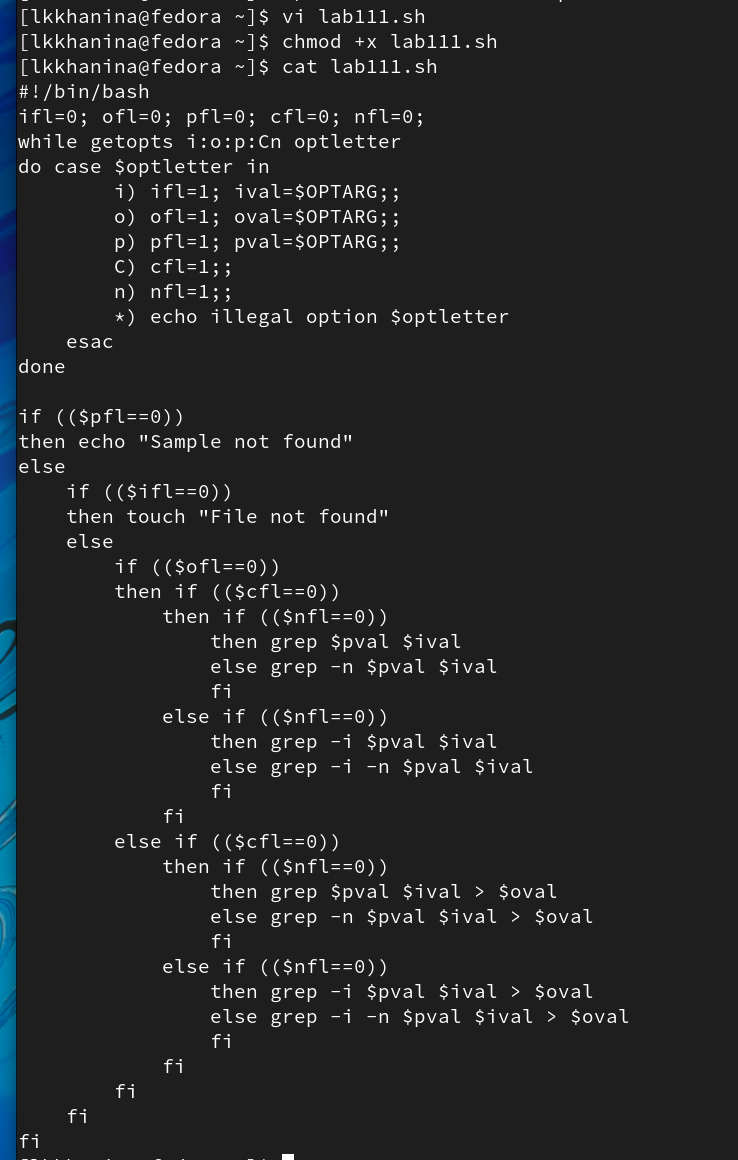
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Задание

* Используя команды getoptsgrep, 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.
* 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. Команд- ный файл должен вызывать эту программу и, проанализировав с помощью команды $?, выдать сообщение о том, какое число было введено.
* 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 ко- мандный файл должен уметь удалять все созданные им файлы (если они существуют).
* 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).

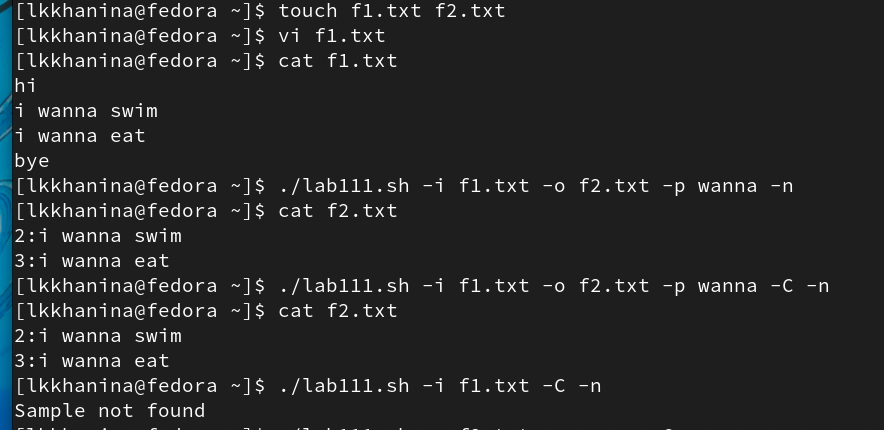
# Выполнение лабораторной работы

1. Cоздадим файл lab111.sh и запишем в него скрипт, который будет анализировать командную строку с ключами. Далее изменим доступ к файлу, чтобы можно было его запускать.



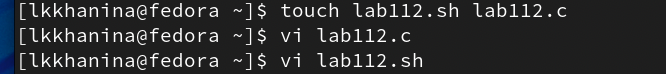
Первый скрипт

1. Проверим, что он работает. Для этого создадим файлы f1.txt и f2.txt. В первый напишем текст. Теперь запустим файл и увидим, что программа отработала корректно.

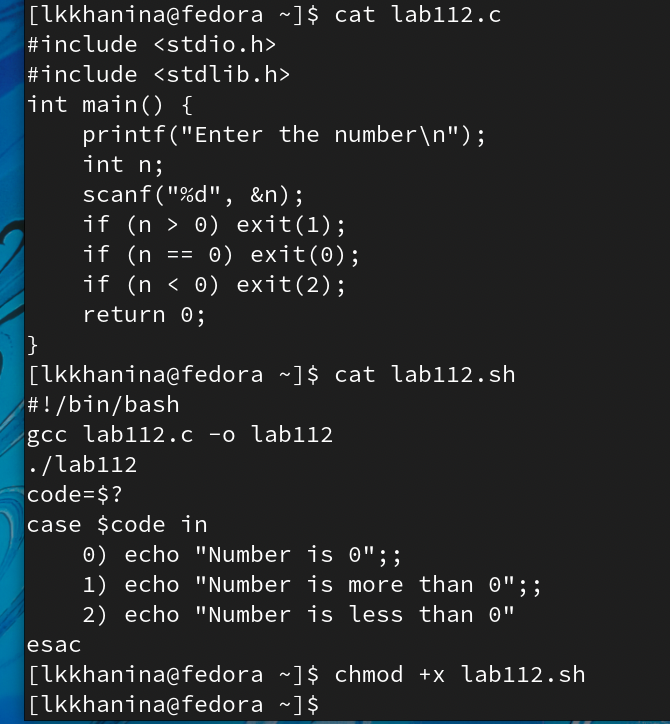


Запускаем первый скрипт

1. Далее создадим файлы lab112.c и lab112.sh. В первый запишем скрипт, который бдут определять, какое число ввел пользователь. Во второй скрипт, который будет анализировать исполнение первого. Изменим доступ к файлу lab112.sh, чтобы можно было его запускать.

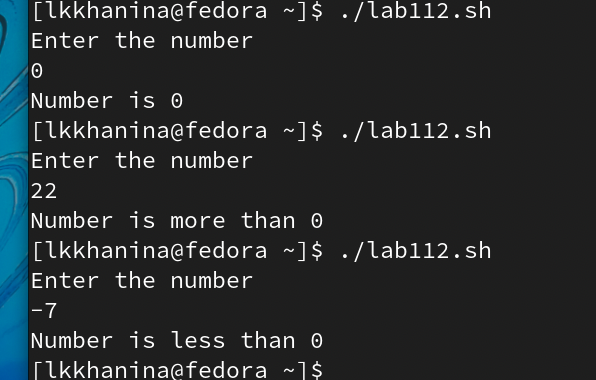


Второй скрипт часть 1



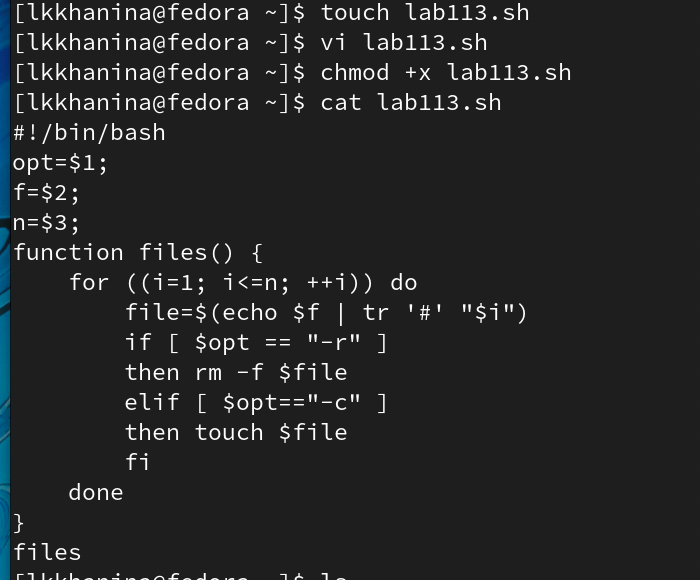
Второй скрипт часть 2

1. Теперь запустим файл и увидим, что программа отработала корректно.



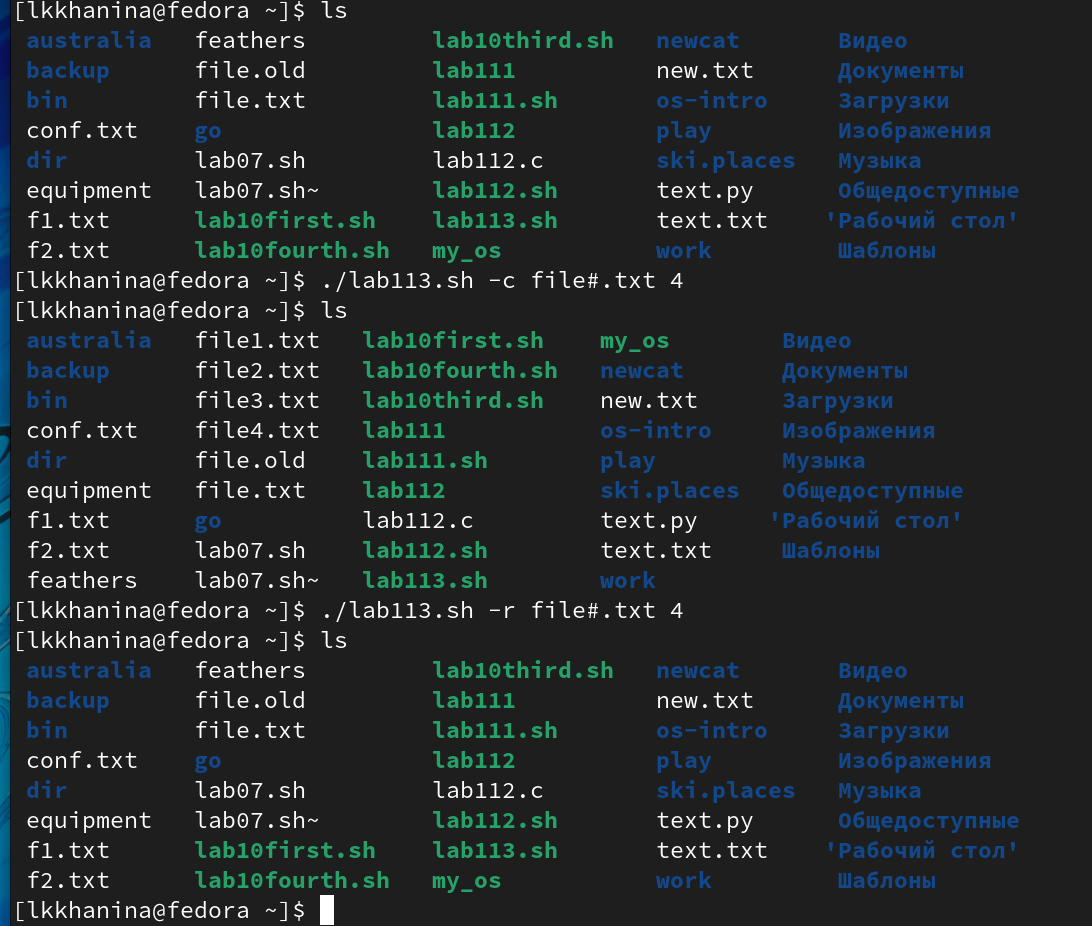
Запускаем второй скрипт

1. Cоздадим файл lab113.sh и запишем в него скрипт, который будет с опцией -c создавать заданное количество файлов, а с опцией -r — их удалять. Далее изменим доступ к файлу, чтобы можно было его запускать.



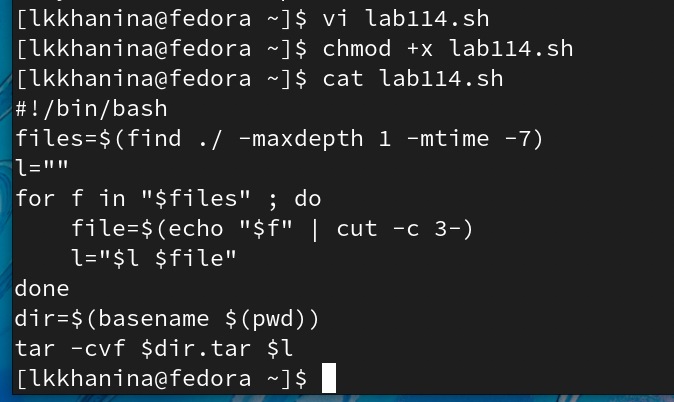
Третий скрипт

1. Теперь запустим файл и увидим, что программа отработала корректно.



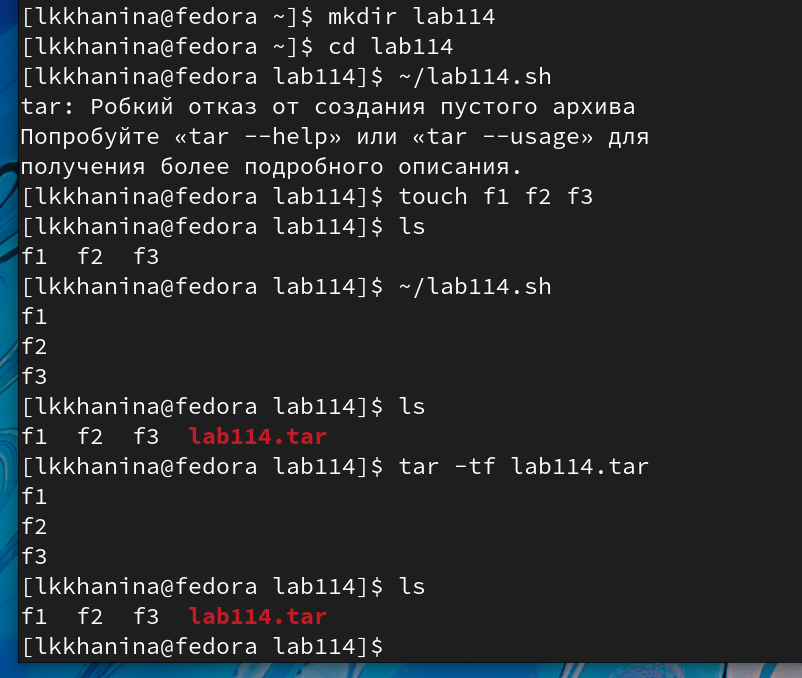
Запускаем третий скрипт

1. Cоздадим файл lab114.sh и запишем в него скрипт, который будет с помощью команды tar запаковывает в архив все файлы, которые были изменены менее недели тому назад, в указанной директории. Далее изменим доступ к файлу, чтобы можно было его запускать.



Четвертый скрипт

1. Чтобы проверить корректность скрипта, создаем директорию и добавляем в нее файлы. Теперь запускаем файл и видим, что программа отработала корректно.



Запускаем четвертый скрипт

# 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”.

1. Операторы управления действиями: if, for, while, case.
2. Чтобы прервать цикл, можно использовать оператор break.
3. Операция сравнения возвращает: значение true («истина»), если высказывание с оператором правдивое (условие выполняется), и false («ложь») — если высказывание с оператором ложное (условие не выполняется).
4. Строка if test -f mani.s/s и является ли этот файл обычным файлом. Если данный файл является каталогом, то команда вернет false.
5. Цикл while выполняет тело цикла пока условие истинно. Цикл until выполняет тело цикла пока условие ложно.

# Выводы

Я научилась писать более сложные командный файлы с использованием логических управляющих конструкций и циклов.