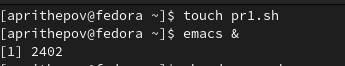
### 11 Лабораторная работа

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

## Выполнение лабораторной работы

1. Создадим командный файл pr1.sh, далее откроем его в emacs (рис.1) рис 1:



изображение

* Используя команды getopts grep, запишем в файл программу (рис.2), которая 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.

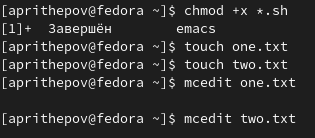
рис 2:



изображение

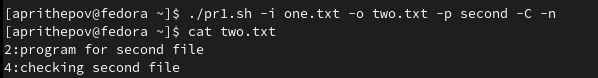
* Проверим работу файла, предварительно дав ему права на выполнение (chmod +x \*.sh), и создадим два файла для проверки работы (touch one.txt two.txt) (рис.3,4). Запускаем файл

рис 3:



изображение

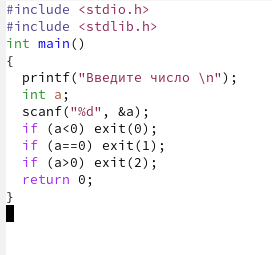
рис 4:



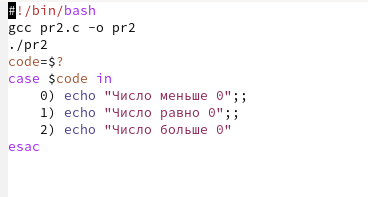
изображение

## 2. Создадим два файла для третьего задания (команда touch pr2.c pr2.sh) и откроем в emacs.

Затем 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 должен вызывать эту программу и, проанализировав с помощью команды $?, выдать сообщение о том, какое число было введено

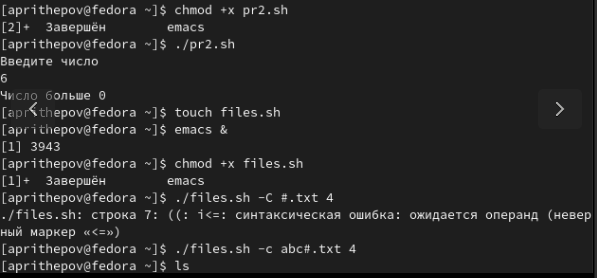


изображение

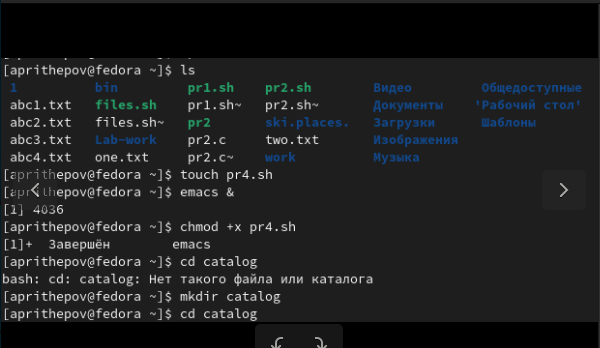


изображение

* Проверим работу командного файла, передав ему права на выполнения и запустив его

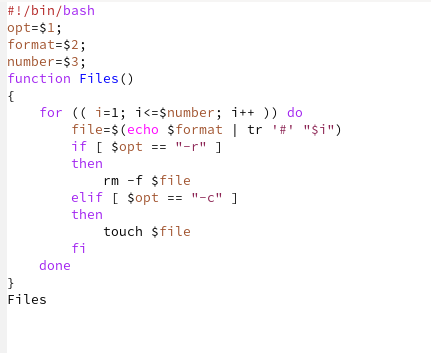


изображение



изображение

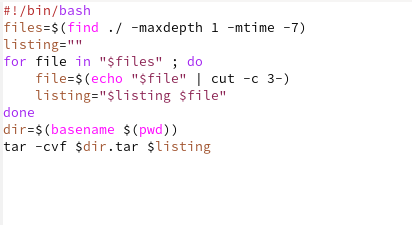
## 3. Создадим командный файл files.sh и откроем его в emacs.Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до 𝑁 (например 1.tmp, 2.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).



изображение

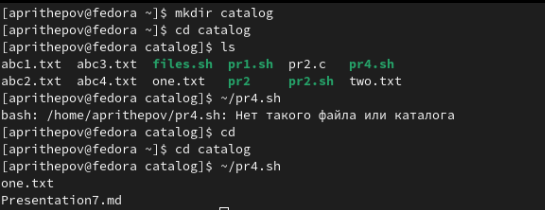
* Проверим его работу, передав ему права на выполнения и запустив его (команда ./files.sh)

## 4. Создадим командный файл pr4.sh и откроем его в emacs. 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 (использовать команду find).



изображение

* Создадим в домашнем каталоге каталог catalog и перенесем туда некоторые файлы, измененные в разное время. Дадим командному файлу право на выполнение (chmod +x pr4.sh) и запустим его в этом каталоге. Файл работает исправно.



изображение

## Выводы

Я научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.