Лабораторная работа №11

Операционные системы

Савурская Полина

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 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;

– -p шаблон—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. 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.
2. 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 командный файл должен уметь удалять все созданные им файлы (если они существуют).
3. 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).

# 3 Выполнение лабораторной работы

1. Создаем файлы file1.sh и file1.txt. Задаем им необходимые разрешения. Проверяем, появилисьли у нас эти файлы.

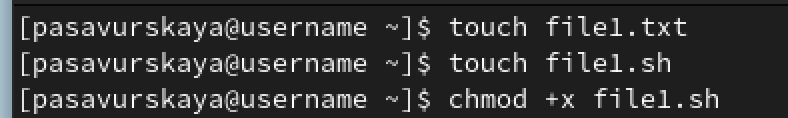


Figure 1: создание файлов

1. В файле file1.txt пишем текст, который будет выводится на экран. В файле file1.sh пишем код.

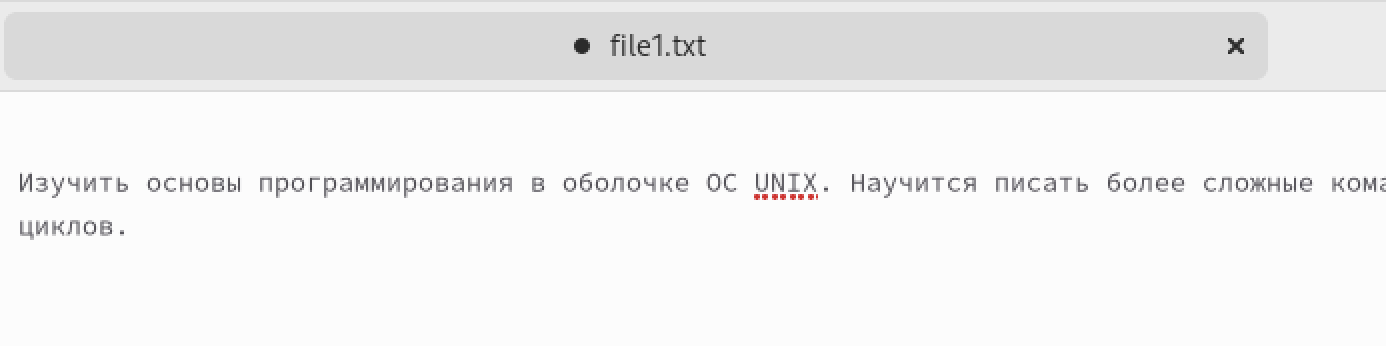


Figure 2: пишем текст



Figure 3: пишем код

1. Задаем файлам необходимые конфигурации и запускаем.

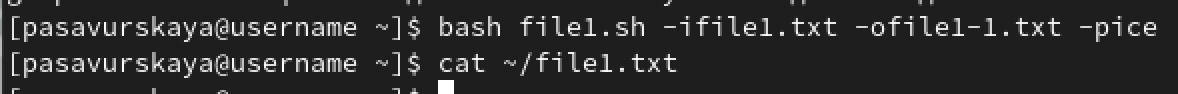


Figure 4: запускаем файл

1. Создаем файлы file2.sh и file2.с. Задаем им необходимые разрешения.

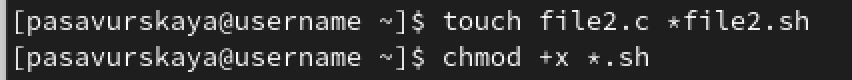


Figure 5: создание файлов

1. Открываем эти файлы и пишем там нужные коды.



Figure 6: пишем код



Figure 7: пишем код

1. Запускаем файл file2.sh и вводим числа. Они выводятся с пояснением относительно нуля. Все сделано правильно.

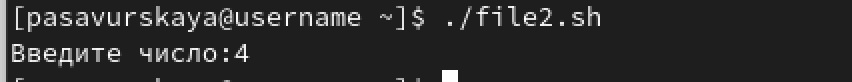


Figure 8: запуск файла file2.sh

1. Создаем файл file3.sh. Задаем ему необходимые разрешения. Проверяем, появился ли у нас этот файл.

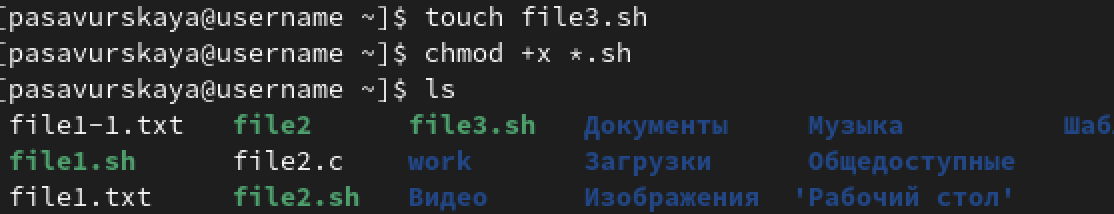


Figure 9: создание файла file3.sh

1. Открываем этот файл и пишем там нужный код.



Figure 10: пишем код

1. Запускаем файл file3.sh. Он создает 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 файл удаляет все созданные им файлы.

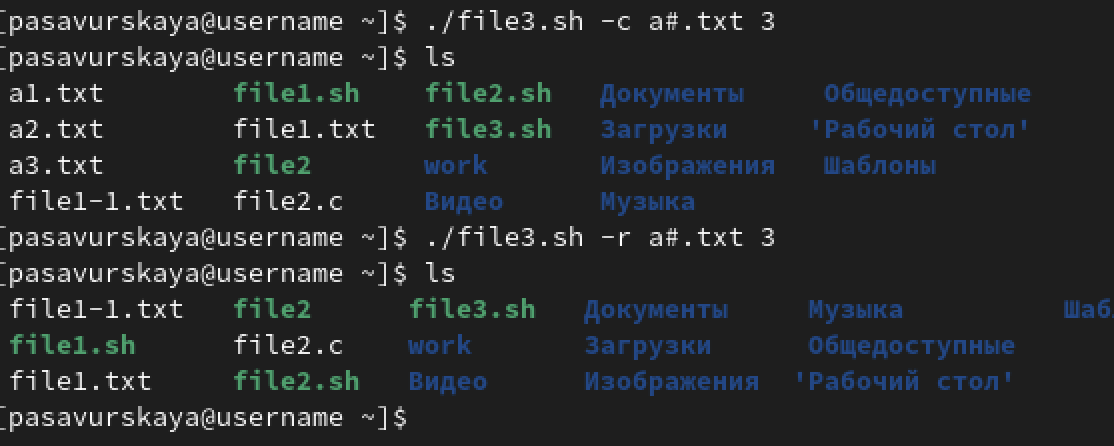


Figure 11: запуск файла file3.sh

1. Создаем файл file4.sh.

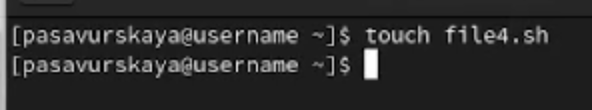


Figure 12: создание файла file4.sh

1. Открываем этот файл и пишем там нужный код.



Figure 13: пишем код

1. Запускаем файл file4.sh.

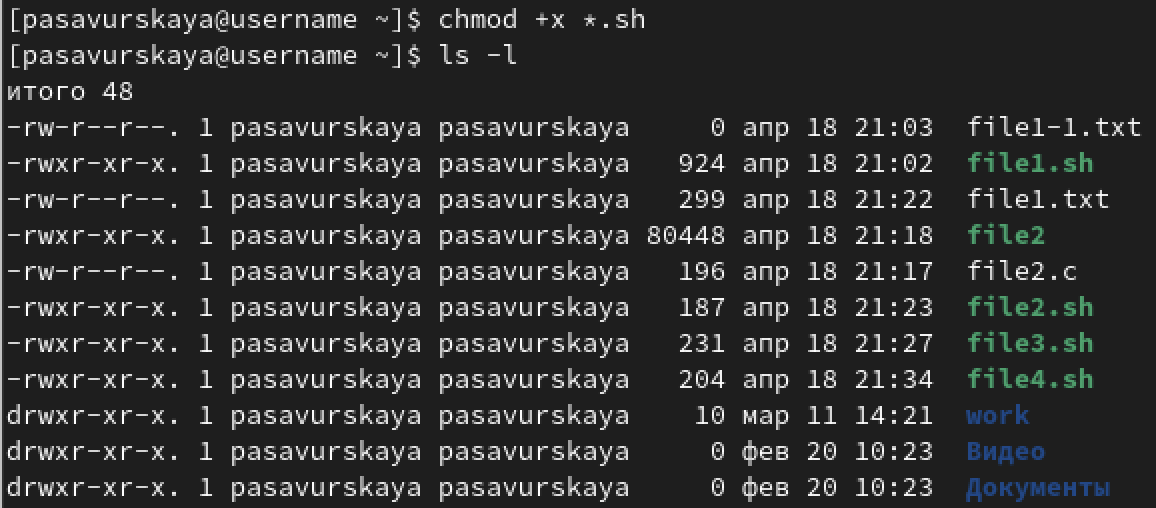


Figure 14: запуск файла file4.sh

# 4 Выводы

Я изучить основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.