Лабораторная работа №13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Кижваткина Анна Юрьевна

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 Выполнение лабораторной работы

Создаем директорию lab13 и перемещаемся в неё. (рис. 1)

Создание директории

Рис. 1: Создание директории

Создаем файл 1.sh и text. (рис. 2)

Создание файла

Рис. 2: Создание файла

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. (рис. 3)

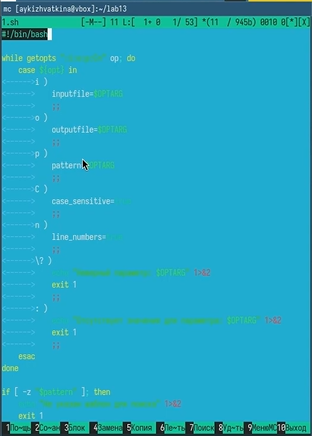


Рис. 3: Программа

Устанавливаем право на выполнение. (рис. 4)

Предоставление доступа

Рис. 4: Предоставление доступа

Проверяем выполнение программы. (рис. 5)

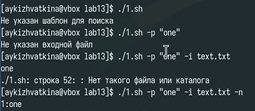


Рис. 5: Запуск программы

Создаем файл 2.sh и 2.с и устанавливаем право на выполнение. (рис. 6)

Создание файла

Рис. 6: Создание файла

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. (рис. 7 и рис. 8)

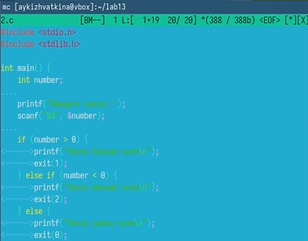


Рис. 7: Программа

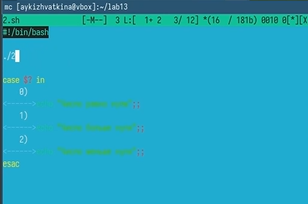


Рис. 8: Программа

Проверяем выполнение программы. (рис. 9)

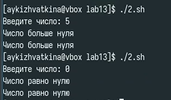


Рис. 9: Запуск программы

Создаем файл 3.sh. (рис. 10)

Создание файла

Рис. 10: Создание файла

Устанавливаем право на выполнение. (рис. 11)

Предоставление доступа

Рис. 11: Предоставление доступа

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). (рис. 12)

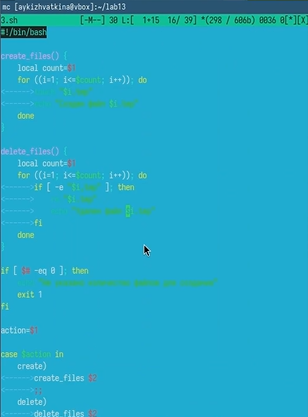


Рис. 12: Программа

Проверяем выполнение программы. (рис. 13)

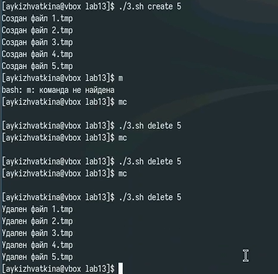


Рис. 13: Запуск программы

Создаем файл 4.sh. (рис. 14)

Создание файла

Рис. 14: Создание файла

Устанавливаем право на выполнение. (рис. 15)

Предоставление доступа

Рис. 15: Предоставление доступа

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). (рис. 16)

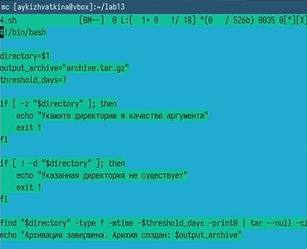


Рис. 16: Программа

Проверяем выполнение программы. (рис. 17 и рис. 18)

Запуск программы

Рис. 17: Запуск программы

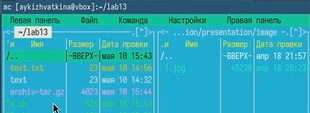


Рис. 18: Запуск программы

# 3 Выводы

С помощью данной лабораторной работы мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.