Отчет по лабораторной работе №13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Разанацуа Сара Естэлл

Содержание

# 1 Цель работы

Цель данной лабораторной 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;
* -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.

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. Команд- ный 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 ко- мандный 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

* Создаю файл с разрешением на исполнение (рис.1).

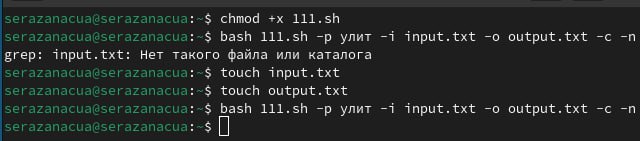


Рис. 1: Создание файла

* а затем ищет в указанном файле нужные строки, определяемые ключом -p (рис.2).

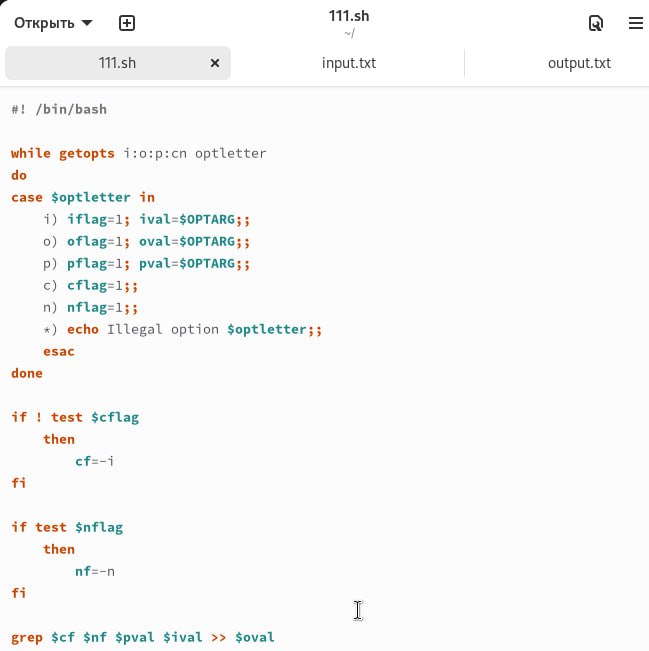


Рис. 2: Код программы

* Результат работы программы в файле output.txt (рис.3).



Рис. 3: Результат работы программы



Рис. 4: Результат работы программы

* Создаю исполняемый файл для второй программы, также создаю файл 12.с для программы на Си. Пишу программу на языке Си, 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 (рис.5).

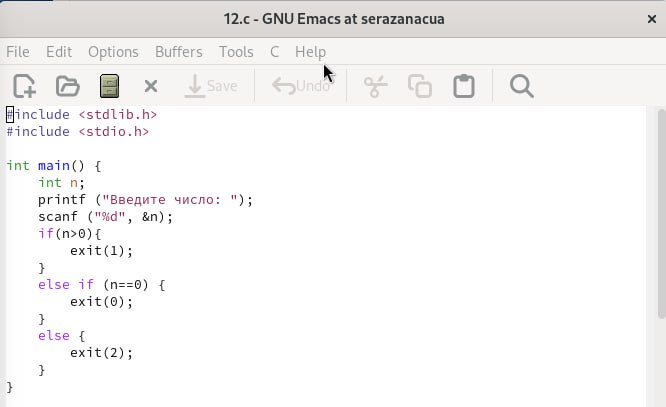


Рис. 5: Код программы на Си

* Командный файл должен вызывать эту программу и, проанализировав с помощью команды $?, выдать сообщение о том, какое число было введено (рис.6).



Рис. 6: Код программы

* Программа работает корректно (рис. fig:007).

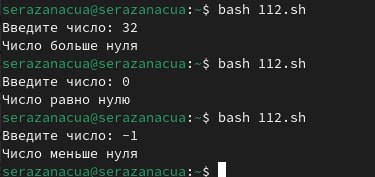


Рис. 7: Результат работы программы

* Создаю исполняемый файл для третьей программы. 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.fig:008).



Рис. 8: Код программы

* Проверяю, что программа создала файлы и удалила их при соответствующих запросах (рис. fig:009).

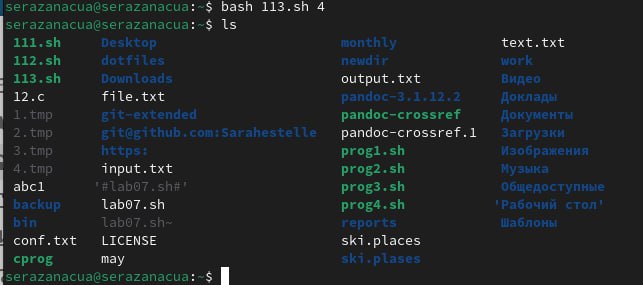


Рис. 9: Результат работы программы

* Создаю исполняемый файл для четвертой программы. Это 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. fig:010).

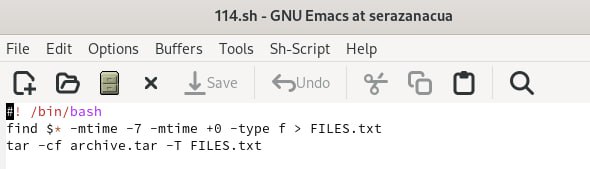


Рис. 10: Код программы

* Проверяю работу программы (рис. fig:011).

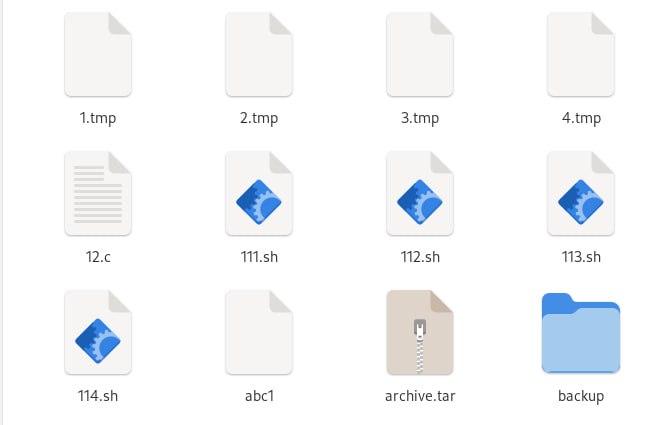


Рис. 11: Результат работы программы

# 4 Выводы

* При выполнении данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов