Отчёт по лабораторной работе №13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Сергей Витальевич Павлюченков

Содержание

Список иллюстраций

Список таблиц

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; – -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.
2. 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.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до ⊠ (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).
4. 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 директорию для работы и файл для первого задания

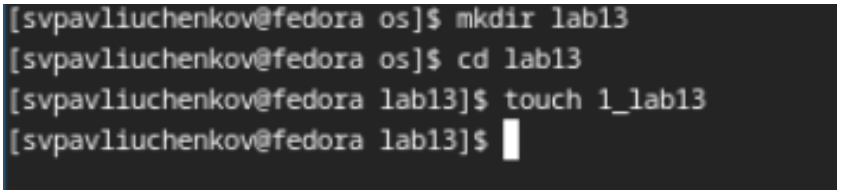

Рис. 3.1: Создание файла

Используя команду getopts grep, написал командный файл, который анализирует командную строку с ключами: – -iinputfile — прочитать данные из указанного файла; – -ooutputfile — вывести данные в указанный файл; – -ршаблон — 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. и Записывает все в файл заданный -o.

Рис. 3.2: Код программы

Для вида сначала вывожу содержимое файла, после чего запускаю программу и утверждаюсь в правильности ее работы. Код нашел все строки в которых встретилось слово apsh.

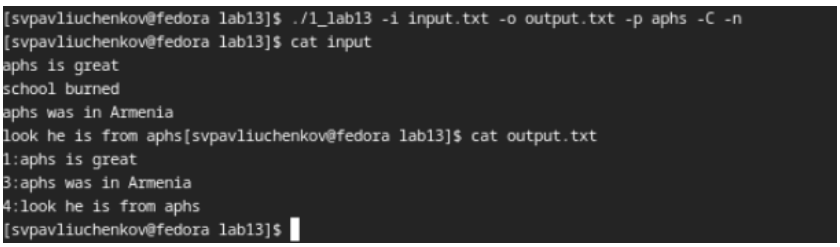

Рис. 3.3: Работа программы

Приступаю к выполнению 2-го задания. Написал 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), пе-

редавая информацию в о коде завершения в оболочку.

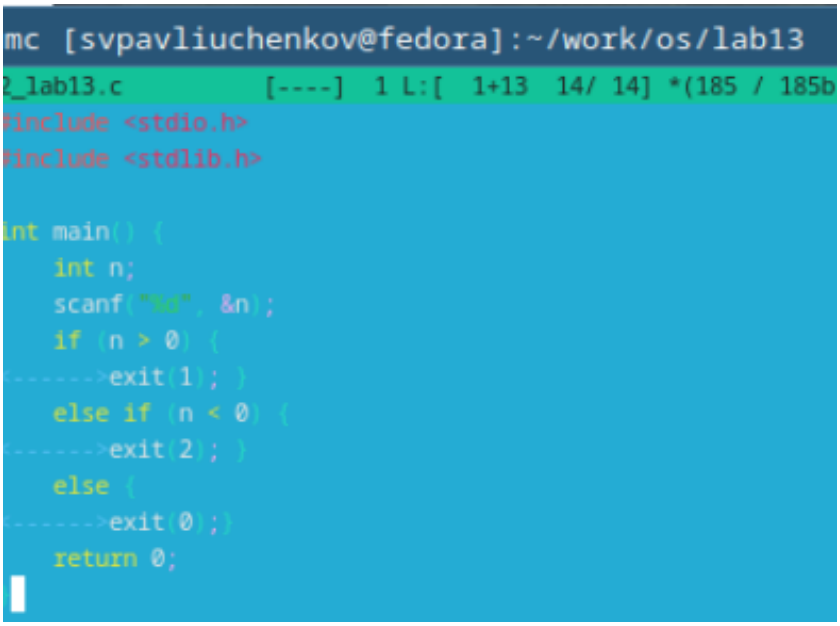

Рис. 3.4: Код программы на C

Написал командный файл который должен вызывать эту программу и, проанализировав с помощью команды $?, выдавать сообщение о том, какое число было введено.

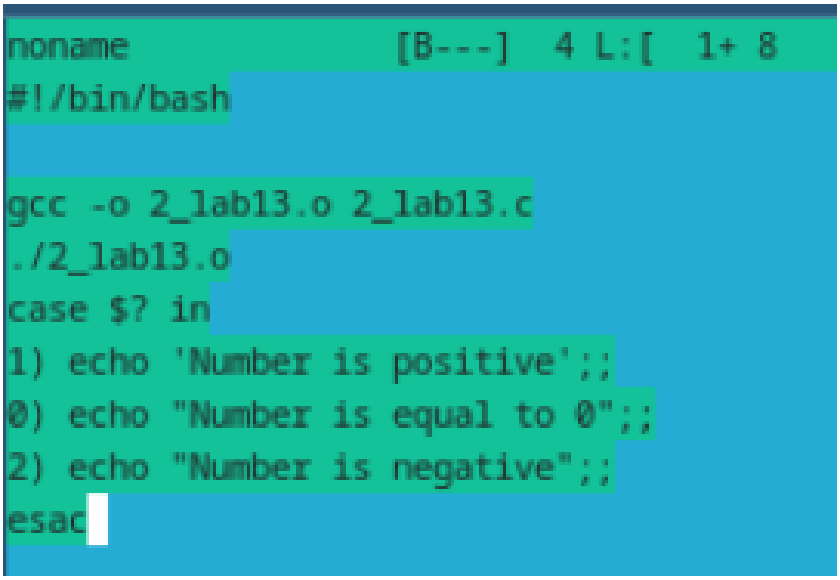

Рис. 3.5: Код командного файла

Запускаю код пару раз, все сработало правильно. тк 6>0 и 0=0.

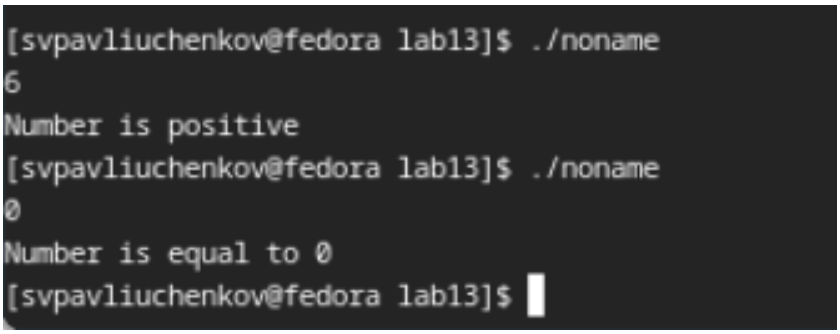

Рис. 3.6: Работа программы

Приступаю к 3 заданию. Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до ⊠ (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 умеет удалять все созданные им файлы (если они существуют).

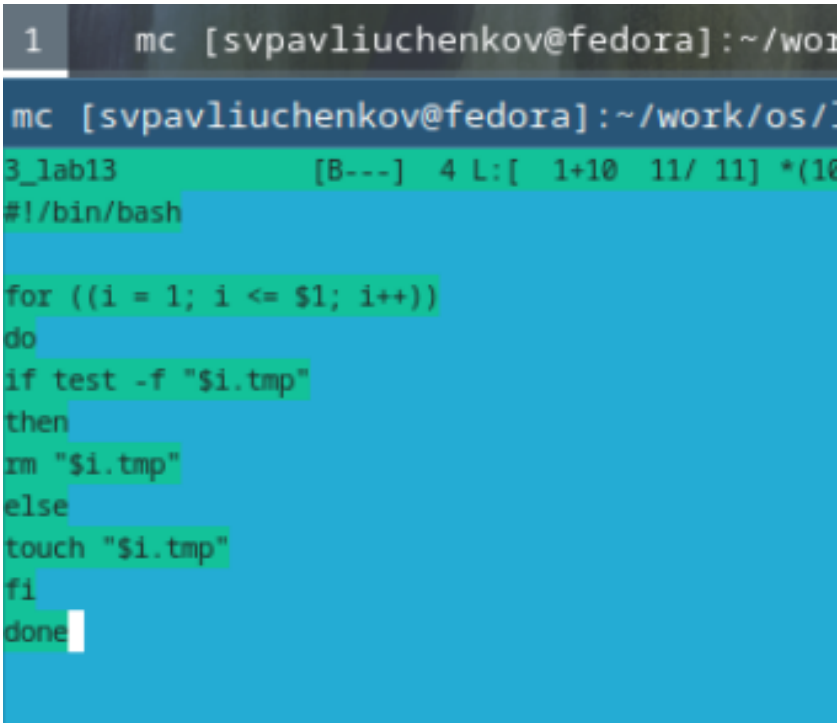

Рис. 3.7: Код программы

Демострирую правильную работу программы. Сначала код создал файлы от 1

до 3, после чего удалил 1-3 и создал 4-5.

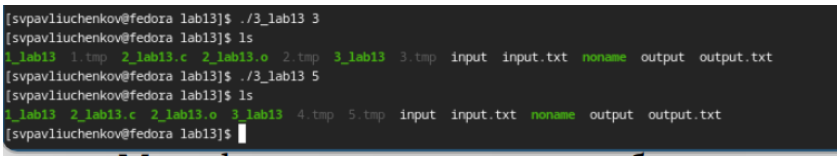

Рис. 3.8: Запуск программы

Приступаю к последнему заданию. Написал командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировал его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовал команду find).

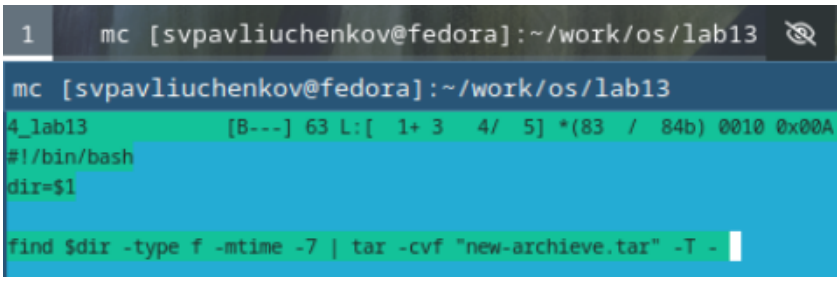

Рис. 3.9: Код программы

Проверяю, что программа запаковала только файлы которые были изменены менее недели назад.

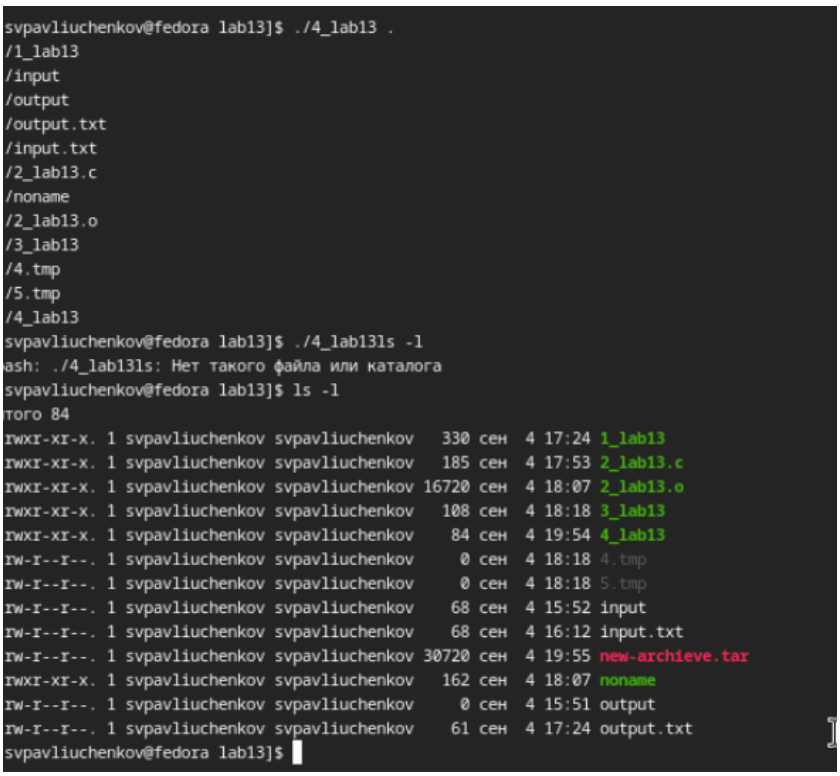

Рис. 3.10: Запуск программы

4 Выводы

Я попрактиковался в создании разных алгоритмов на bash. Поработал в getopts, grep и find.

5 Контрольные вопросы

1. Каково предназначение команды getopts? getopts осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.
2. Какое отношение метасимволы имеют к генерации имён файлов? имя файла содержится в $0 .
3. Какие операторы управления действиями вы знаете? for, case, if и while
4. Какие операторы используются для прерывания цикла? break и continue
5. Для чего нужны команды false и true? Для циклов и условных конструкций.
6. Что означает строка if test -f man$s$/i.$s, встреченная в командном файле? Данная строка проверяет является ли man$1/1.s файлом.
7. Объясните различия между конструкциями while и until. При замене в операторе цикла while служебного слова while на until условие, при выполнении которого осуществляется выход из цикла, меняется на противоположное