Лабораторная работа No 12

Программирование в командномпроцессоре ОС UNIX. Ветвления и циклы

Кеан Путхеаро НПИбд-01-20

Содержание

# Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать бо-лее сложные командные файлы с использованием логических управляющих кон-струкций и циклов.

# Задание

1. Используя командыgetopts grep, написать командный файл, который анали-зирует 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, являетсяли оно больше нуля, меньше нуля или равно нулю. Затем программа завершаетсяс помощью функцииexit(n), передавая информацию в о коде завершения воболочку. Командный файл должен вызывать эту программу и, проанализировавс помощью команды$?, выдать сообщение о том, какое число было введено.
2. Написать командный файл, создающий указанное число файлов, пронумерован-ных последовательно от1доN(например1.tmp,2.tmp,3.tmp,4.tmpи т.д.).Число файлов, которые необходимо создать, передаётся в аргументы команднойстроки. Этот же командный файл должен уметь удалять все созданные им файлы(если они существуют).
3. Написать командный файл, который с помощью командыtarзапаковывает вархив все файлы в указанной директории. Модифицировать его так, чтобы запа-ковывались только те файлы, которые были изменены менее недели тому назад(использовать командуfind).

# Выполнение лабораторной работы

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.

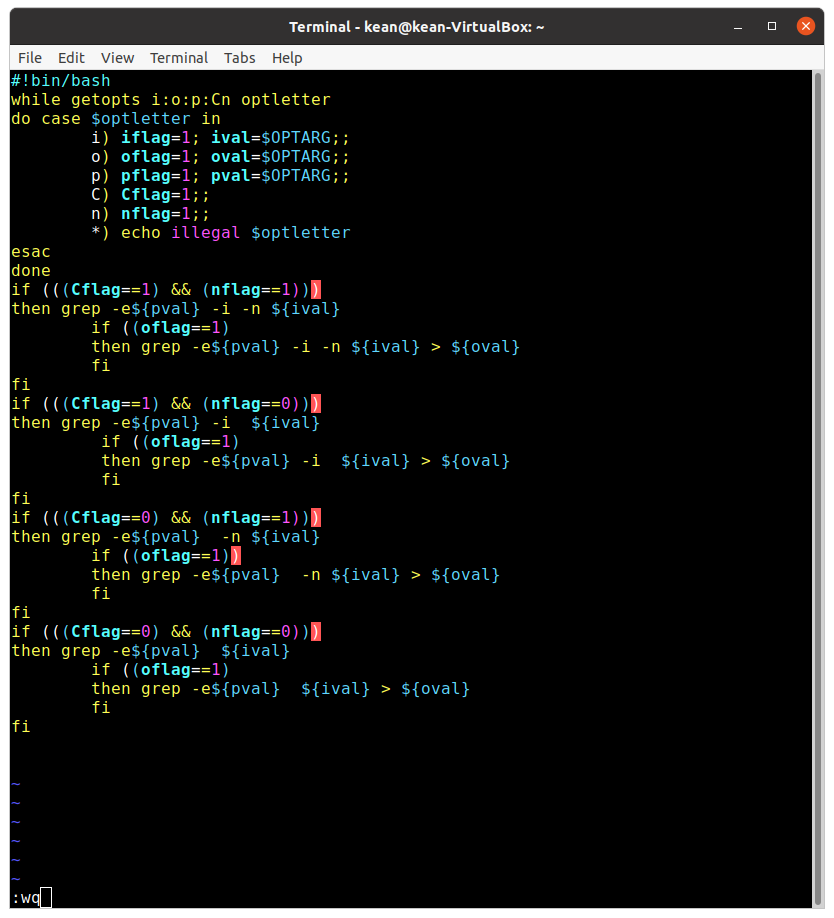


Figure 1: Первый скрипт

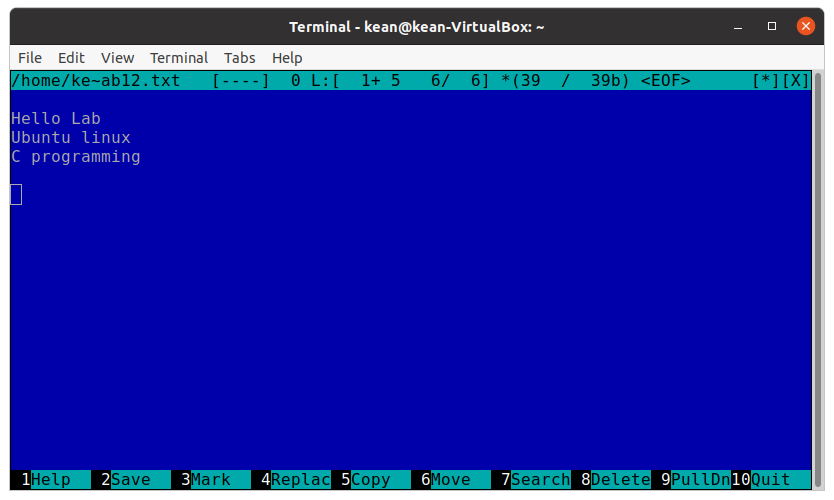


Figure 2: Проверка работы скрипта

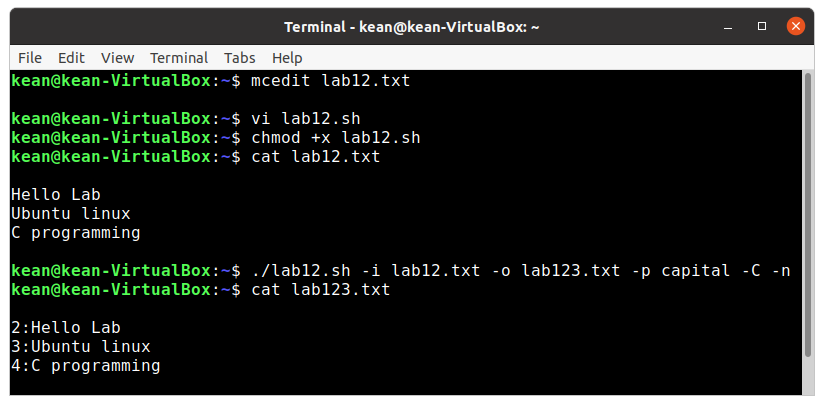


Figure 3: Проверка работы скрипта

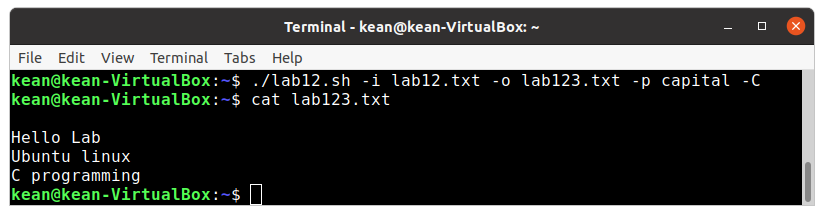


Figure 4: Проверка работы скрипта

1. 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), 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.

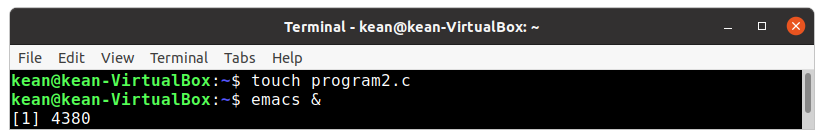


Figure 5: создание файла program2.c

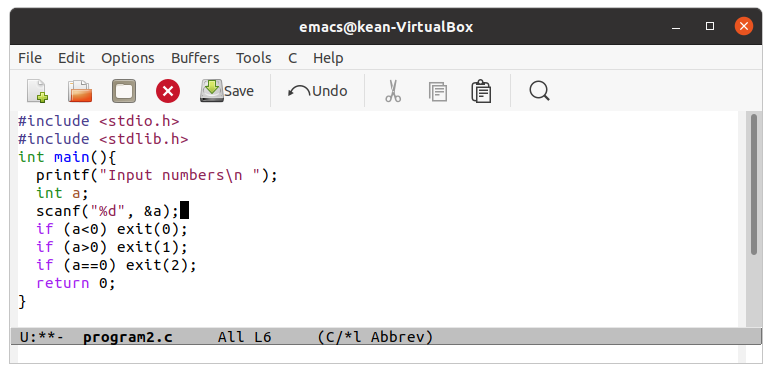


Figure 6: Файл program2.c

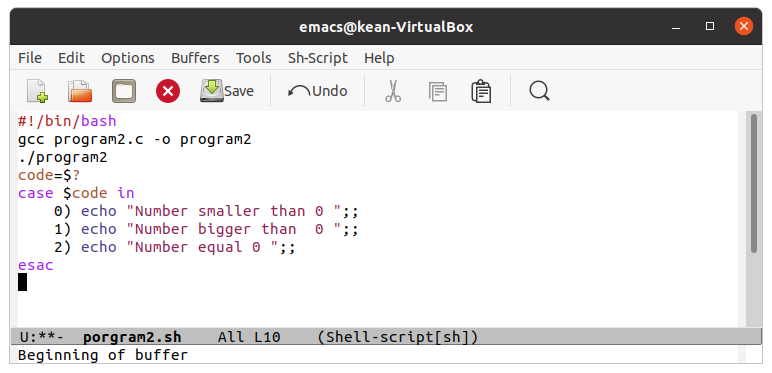


Figure 7: Файл program2.sh

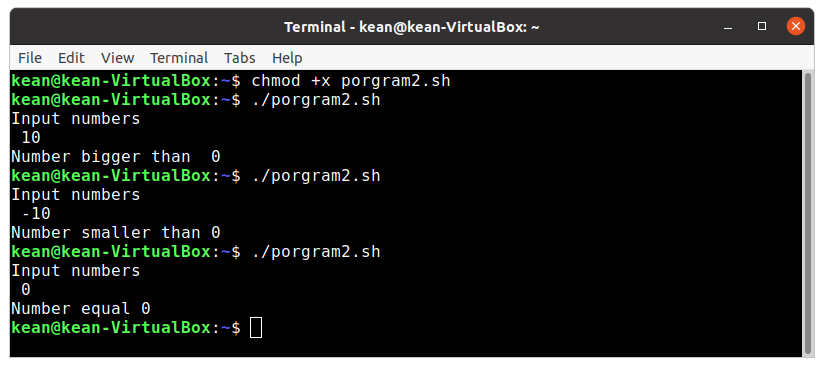


Figure 8: pезультат работы командного файла

1. Написал 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).

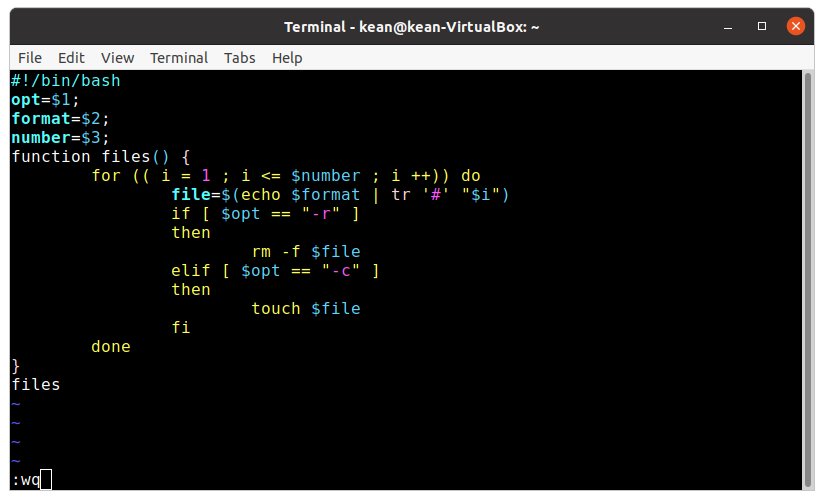


Figure 9: создание файла Program3.sh

файлы b1.txt b2.txt b3.txt созданы.

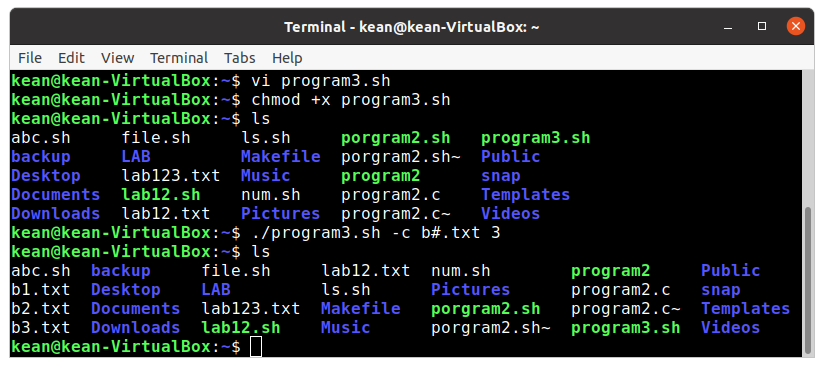


Figure 10: результат работы командного файла

файлы b1.txt b2.txt b3.txt удалены.

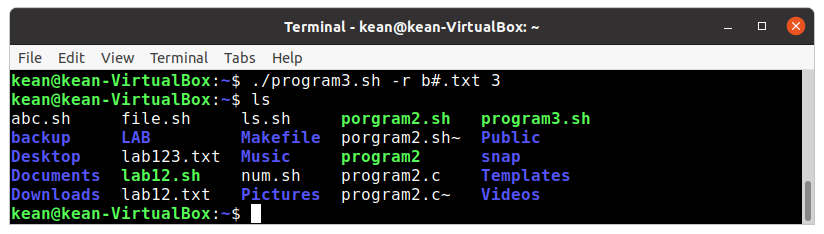


Figure 11: результат работы командного файла

1. Написал 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).

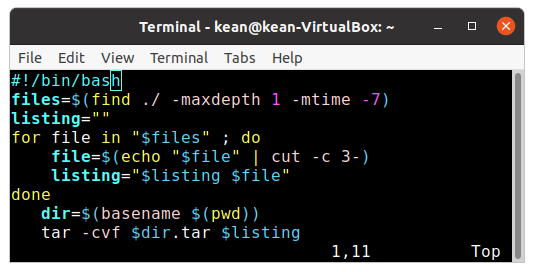


Figure 12: создание файла Program4.sh

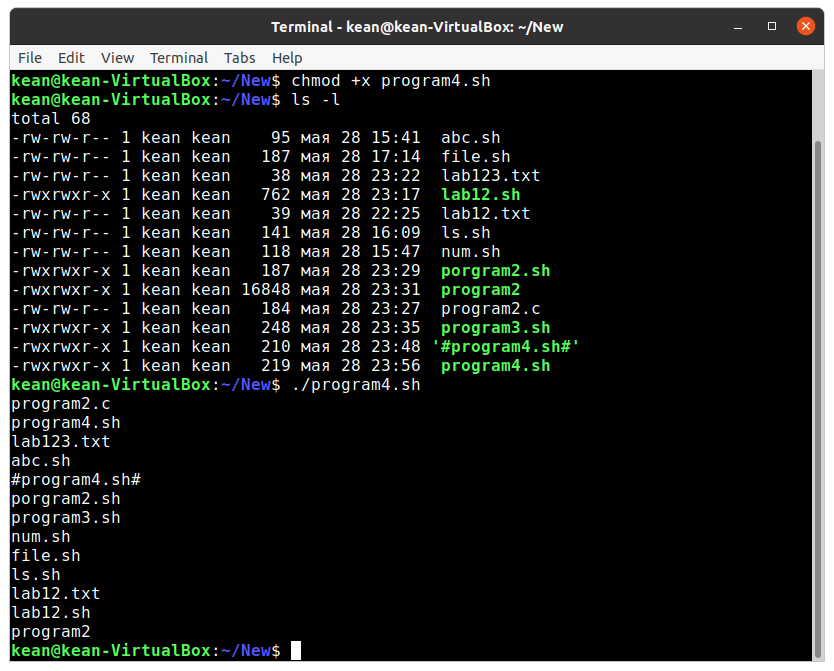


Figure 13: результат работы

Итак, New.tar создан.

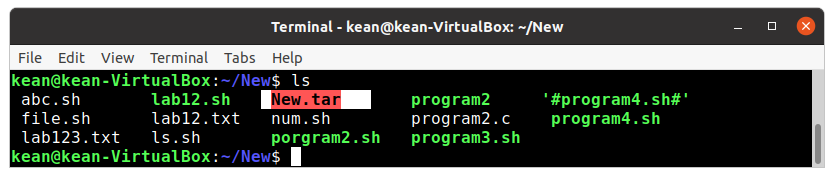


Figure 14: результат работы

# Вывод

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

# Библиография

1. (Лабораторная работа №12) https://esystem.rudn.ru/pluginfile.php/1142380/mod\_resource/content/3/009-lab\_shell\_prog\_2.pdf
2. (stackexchange) https://vi.stackexchange.com/questions/10209/execute-current-buffer-as-bash-script-from-vim
3. (BASH: функция getopts — используем опции в скриптах) https://esystem.rudn.ru/pluginfile.php/1142380/mod\_resource/content/3/009-lab\_shell\_prog\_2.pdf
4. (stackoverflow) https://stackoverflow.com/questions/16483119/an-example-of-how-to-use-getopts-in-bash