Лабораторная работа №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Перевощиков Данил Алексеевич

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 Ход работы

**1.** Используя команды getopts grep, написали командный файл, который анализирует командную строку с ключами: - -i inputfile — прочитать данные из указанного файла; - -o outputfile — 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.(рис. 1)



Рис. 1: Первый скрипт.

Мы использовали файл conf.txt из одной из прошлых работ.(рис. 2)

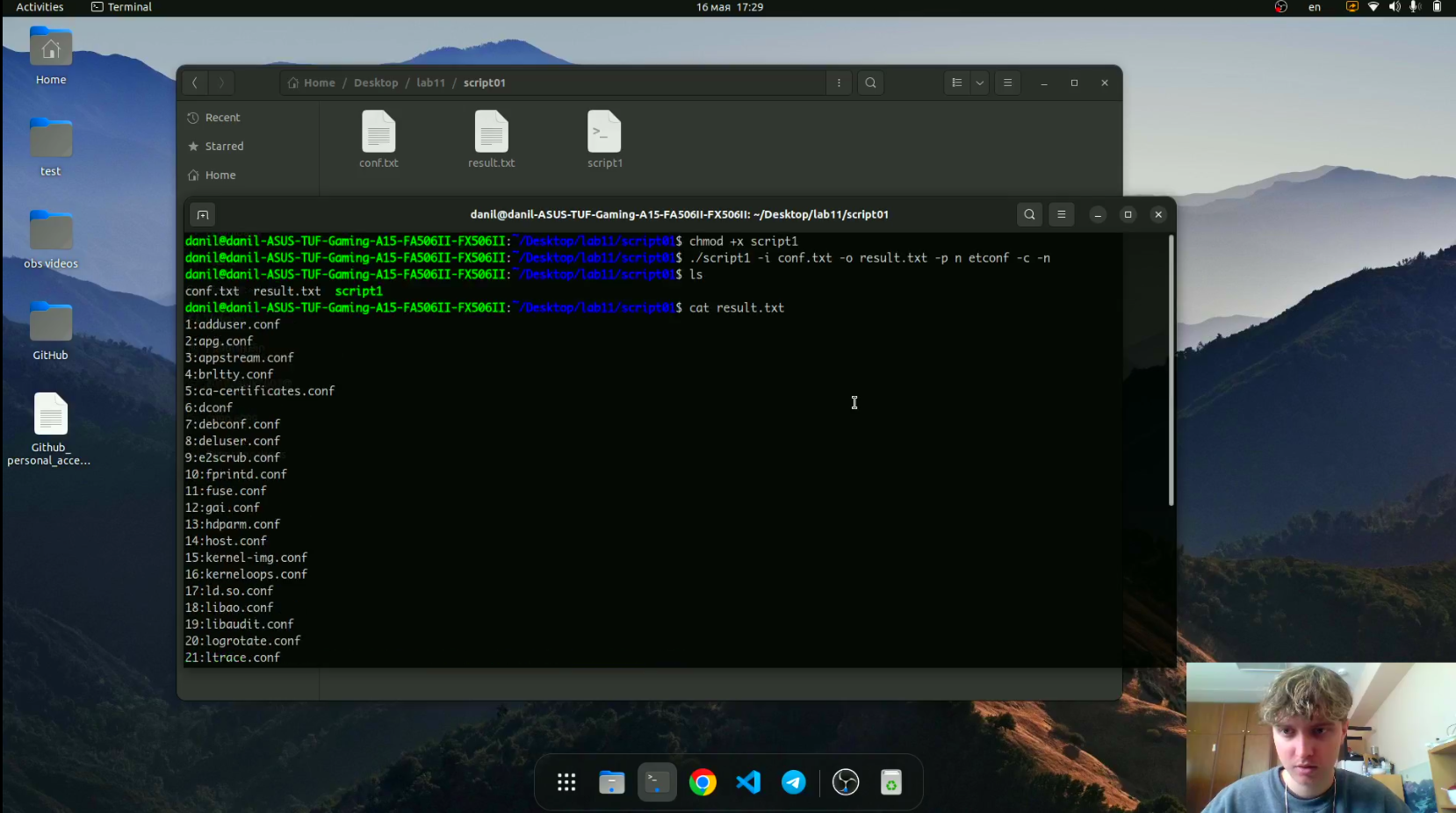


Рис. 2: Выполнение первого скрипта.

**2.** Написали на языке с++ 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 о коде завершения в оболочку. Командный файл вызывает эту программу и, проанализировав с помощью команды $?, выдает сообщение о том, какое число было введено.(рис. 3-5)

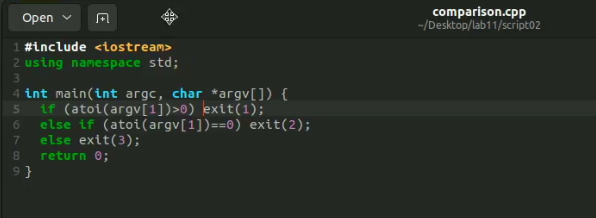


Рис. 3: Программа на языке с++.

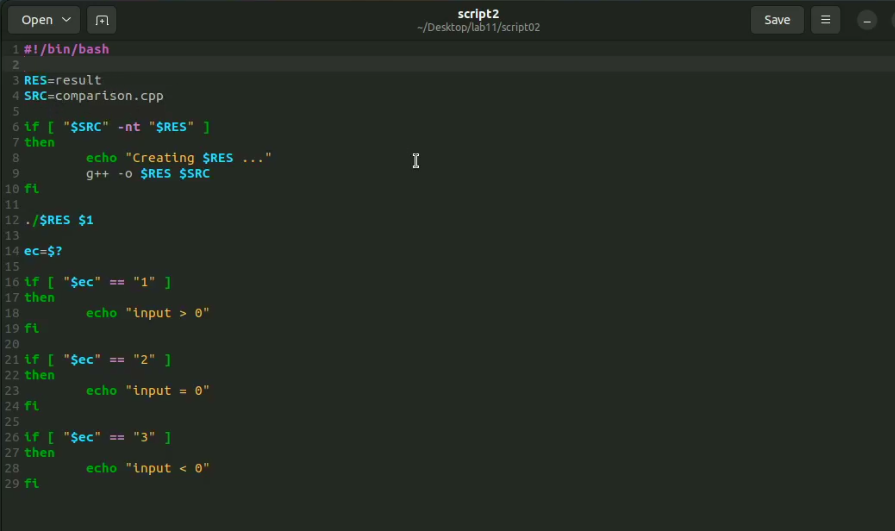


Рис. 4: Второй скрипт.

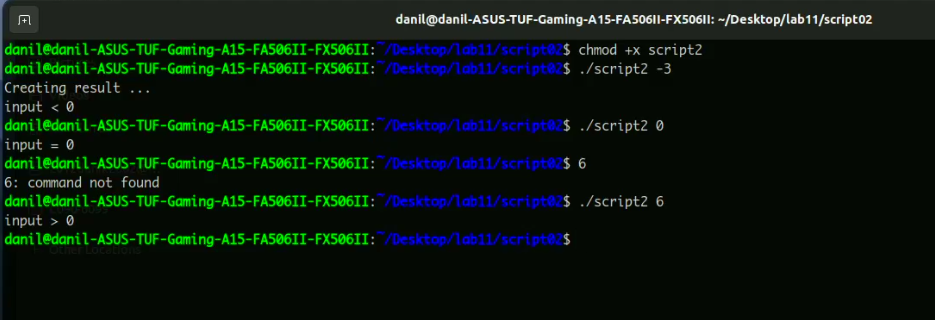


Рис. 5: Выполнение второго скрипта.

**3.** Написали командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл умеет удалять все созданные им файлы (если они существуют).(рис. 6, 7)

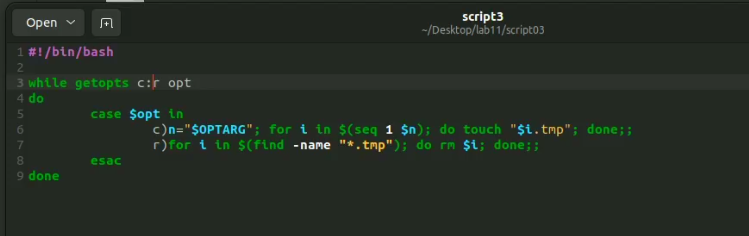


Рис. 6: Третий скрипт.

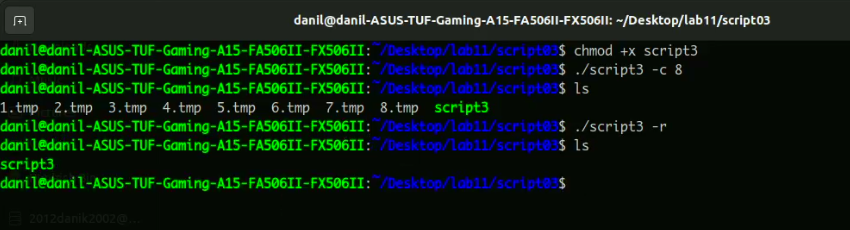


Рис. 7: Выполнение третьего скрипта.

**4.** Написали командный файл, который с помощью команды tar запаковывает в архив файлы, которые были изменены менее недели тому назад.(рис. 8, 9)

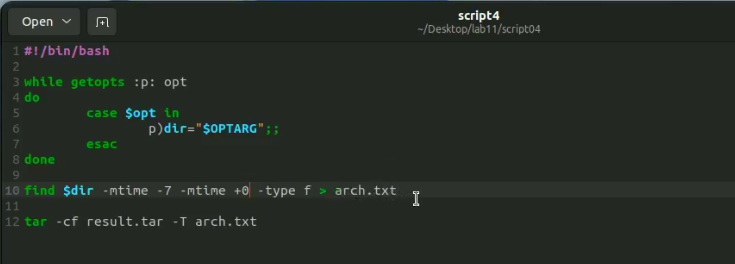


Рис. 8: Третий скрипт.

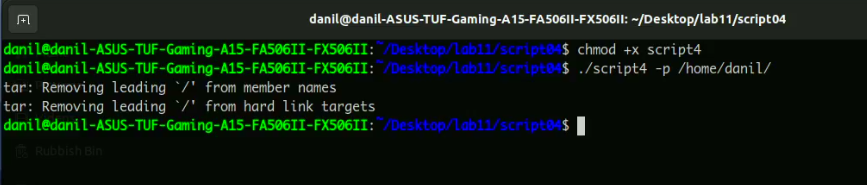


Рис. 9: Выполнение четвертого скрипта.

После этого создался архив result.tar.(рис. 10)

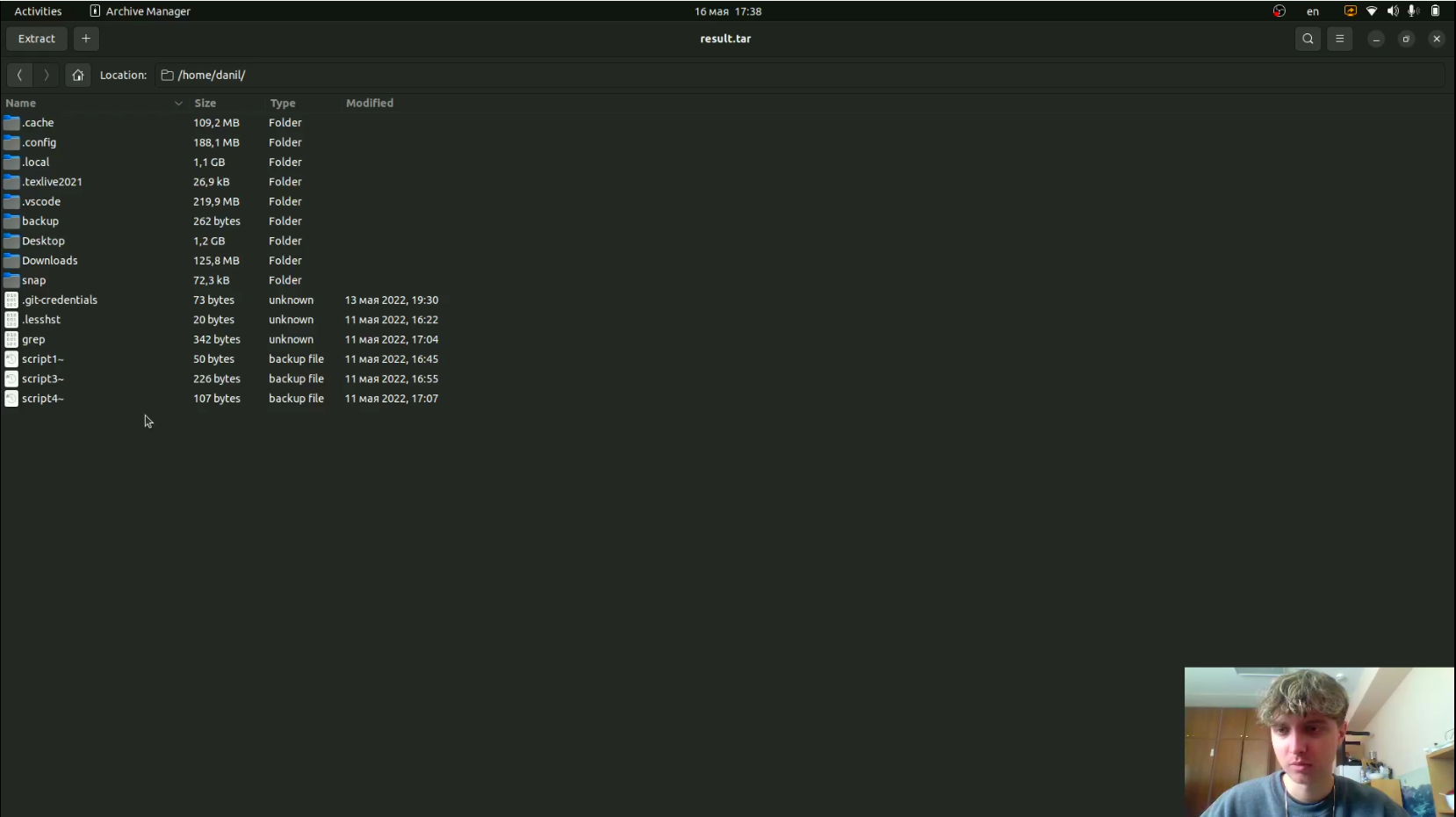


Рис. 10: Результат выполнения четвертого скрипта.

# 3 Вывод

Изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# 4 Контрольные вопросы

1. *Каково предназначение команды getopts?*

Осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.

1. *Какое отношение метасимволы имеют к генерации имён файлов?*

При перечислении имен файлов текущего каталога можно использовать следующие символы: - \* — соответствует произвольной, в том числе и пустой строке; - ? — соответствует любому одному символу; - [c1-c1] — соответствует любому символу, лексикографически на ходящемуся между символами c1 и с2. - echo \* — выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды ls; - ls \*.c — выведет все файлы с последними двумя символами, равными .c. - echo prog.? — выдаст все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются prog. . - [a-z]\* — соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита.

1. *Какие операторы управления действиями вы знаете?*

Часто бывает необходимо обеспечить проведение каких-либо действий циклически и управление дальнейшими действиями в зависимости от результатов проверки некоторого условия. Для решения подобных задач язык программирования bash предоставляет Вам возможность использовать такие управляющие конструкции, как for, case, if и while. С точки зрения командного процессора эти управляющие конструкции являются обычными командами и могут использоваться как при создании командных файлов, так и при работе в интерактивном режиме. Команды, реализующие подобные конструкции, по сути дела являются операторами языка программирования bash. Поэтому при описании языка программирования bash термин оператор будет использоваться наравне с термином команда.

1. *Какие операторы используются для прерывания цикла?*

Два несложных способа позволяют вам прерывать циклы в оболочке bash. Команда break завершает выполнение цикла, а команда continue завершает данную итерацию блока операторов. Команда break полезна для завершения цикла while в ситуациях, когда условие перестает быть правильным. Пример бесконечного цикла while, с прерыванием в момент, когда файл перестает существовать:

* while true do
* if [! -f $file] then break
* fi
* sleep 10 done

1. *Для чего нужны команды false и true?*

Команды ОС UNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда test, например, создана специально для использования в командных файлах. Единственная функция этой команды заключается в выработке кода завершения.

1. *Что означает строка if test -f mani.$s, встреченная в командном файле?*

Введенная строка означает условие существования файла mani.$s

1. *Объясните различия между конструкциями while и until.*

Если речь идет о 2-х параллельных действиях, то это while. когда мы показываем, что сначала делается 1-е действие. потом оно заканчивается при наступлении 2-го действия, применяем until.