Отчёт по лабораторной работе №12

дисциплина: Операционные системы

Латаева Гюзелия Андреевна

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 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. Командный 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 командный 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

**Задание 1.**

1. Создала файл getopts.sh и устанавливаю ему права на выполнение.
2. Написала 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. ??), (рис. ??)

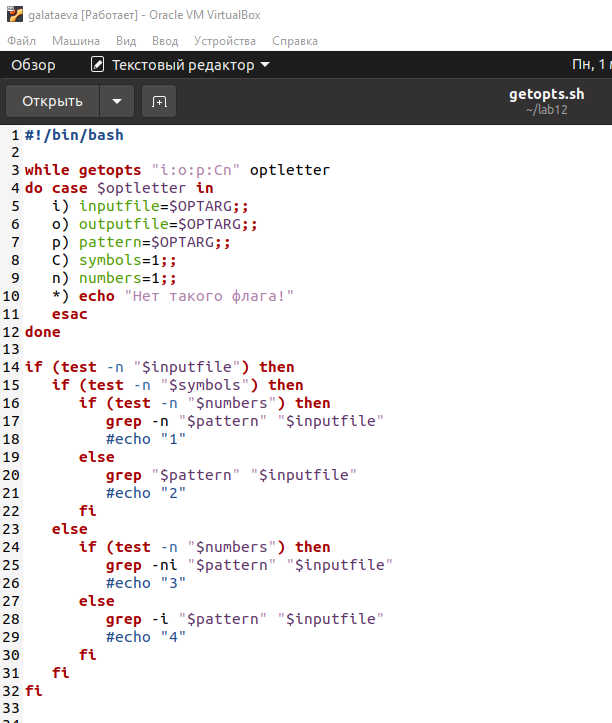


Рисунок 1

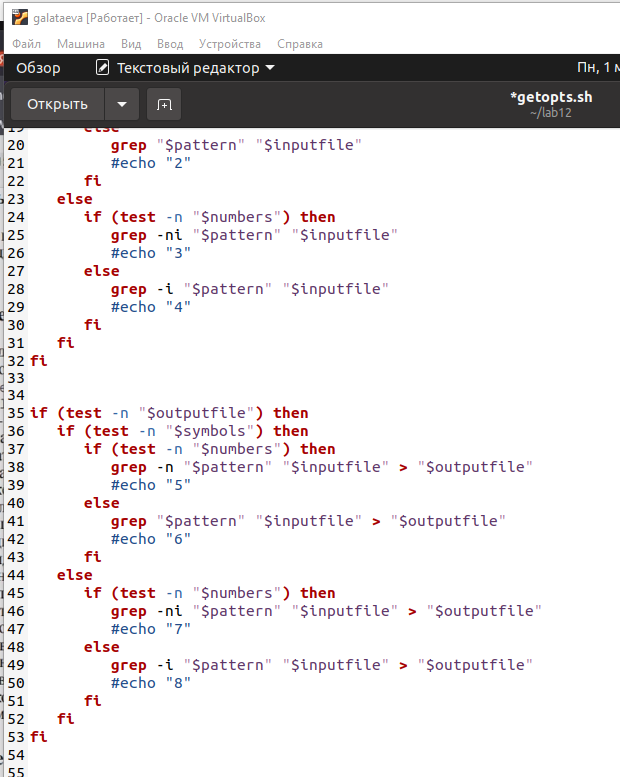


Рисунок 2

1. Результат после выполнения: (рис. ??)

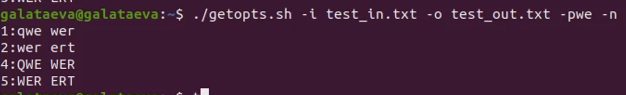


Рисунок 3

**Задание 2.**

1. Создала файл exit\_prog.sh и устанавливаю ему права на выполнение.
2. Написала 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. (рис. ??). Командный файл должен вызывать эту программу и, проанализировав с помощью команды $?, выдать сообщение о том, какое число было введено (рис. ??):

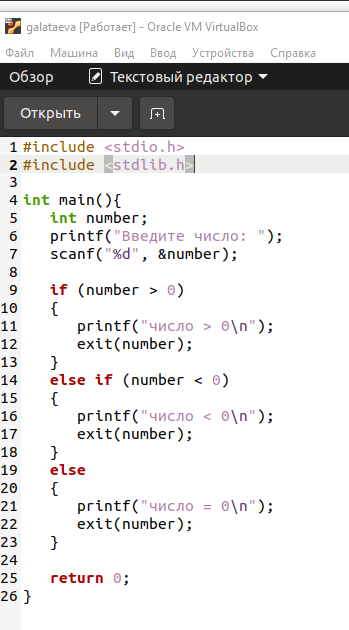


Рисунок 4

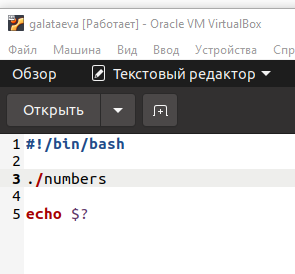


Рисунок 5

1. Выполнила скрипт и получила результат: (рис. ??)

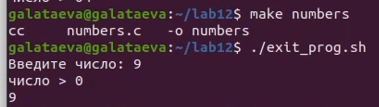


Рисунок 6

**Задание 3.**

1. Создала файл create\_file.sh и устанавливаю ему права на выполнение.
2. Написала 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. ??)

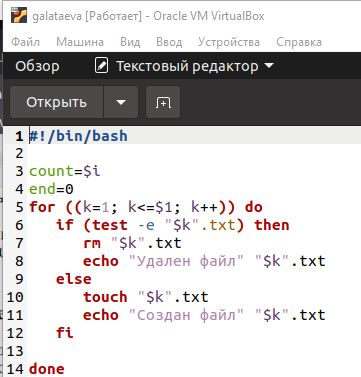


Рисунок 7

1. Выполнила скрипт и получила результат: (рис. ??):

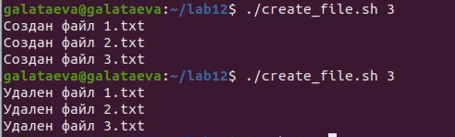


Рисунок 8

**Задание 4.**

1. Создала файл tar.sh и устанавливаю ему права на выполнение.
2. Написала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории и модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (рис. ??):

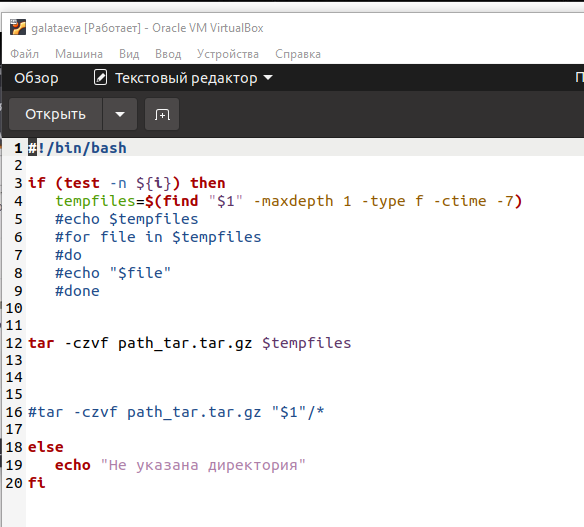


Рисунок 9

1. Выполнила скрипт и получила результат: (рис. ??):

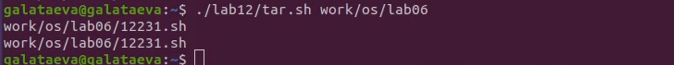


Рисунок 10

1. Результат: (рис. ??):

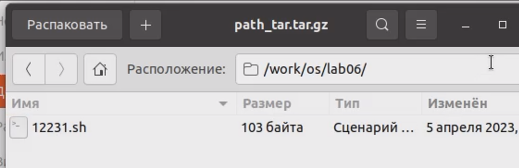


Рисунок 11

# 4 Выводы

Я изучила основы программирования в оболочке ОС UNIX и научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# 5 Контрольные вопросы

1. Каково предназначение команды getopts?

Команда getopts в скриптах оболочки Unix/Linux предназначена для обработки опций командной строки. Она позволяет скрипту получать аргументы командной строки в определенном формате, что делает его более удобным для использования конечными пользователями. Команда getopts используется в цикле while, который обрабатывает каждый аргумент командной строки по очереди. Кроме того, getopts позволяет обрабатывать несколько опций в одном аргументе командной строки, а также определить опцию по умолчанию, которая будет использоваться, если пользователь не указал никаких опций.

1. Какое отношение метасимволы имеют к генерации имён файлов?

Метасимволы в UNIX-подобных операционных системах используются для генерации имен файлов и путей к ним. Например, символ звездочки (\*) соответствует любой последовательности символов в имени файла, а символ вопросительного знака (?) соответствует одному любому символу. Также метасимволы используются для выполнения поиска и замены текста в файле при использовании утилит, таких как sed и awk.

1. Какие операторы управления действиями вы знаете?

В языке программирования bash есть много операторов управления действиями, вот некоторые из них:

if-then-else: оператор условия, позволяет выполнять определенные действия в зависимости от условия.

for: оператор цикла, позволяет выполнить набор инструкций несколько раз, итерируясь по списку значений.

while: оператор цикла, позволяет выполнять набор инструкций, пока определенное условие остается истинным.

case: оператор выбора, позволяет выбрать действие, которое должно быть выполнено, исходя из значения переменной.

until: оператор цикла, выполняет набор инструкций до тех пор, пока определенное условие не станет истинным.

select: оператор выбора, позволяет создать меню выбора для пользователя.

break: оператор, который прерывает выполнение цикла или выбора.

continue: оператор, который пропускает текущую итерацию цикла или выбора и переходит к следующей.

trap: оператор, который позволяет устанавливать обработчики сигналов и выполнять определенные действия при получении сигнала.

shift: оператор, который позволяет сдвигать значения параметров командной строки на одну позицию.

Это только некоторые из операторов управления действиями в языке bash.

1. Какие операторы используются для прерывания цикла?

break: используется для немедленного выхода из цикла. continue: используется для пропуска текущей итерации цикла и перехода к следующей.

1. Для чего нужны команды false и true?

Команды false и true являются базовыми утилитами в UNIX-подобных операционных системах, которые возвращают код возврата 1 и 0 соответственно. Обычно эти команды используются в комбинации с другими командами и операторами для контроля над процессами, запускаемыми в скриптах.

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

Данная строка используется для проверки наличия файла в определенном месте в файловой системе. Она проверяет, существует ли файл с именем mani.$s, где $s, $i - переменные, заданные ранее в скрипте.

1. Объясните различия между конструкциями while и until.

Конструкция while используется для выполнения цикла, пока условие истинно. Цикл выполняется, пока результат выражения внутри скобок while истинен.

Конструкция until похожа на while, но выполняет цикл, пока условие ложно. Цикл будет выполняться, пока результат выражения внутри скобок until ложен.

# Список литературы