Первым делом установим текстовый редактор VIM и набор компиляторов gcc:





Также мы можем посмотреть версии:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Для того чтобы открыть текстовый редактор пропишите «vim» и откроется окно, в дальнейшем так мы будем писать код;

Чтобы выйти из текстового редактора необходимо нажать «esc» и написать «:wq», чтобы изменения сохранились.

Теперь приступим к созданию файла, для этого пропишем «vim» + название вашего файла.

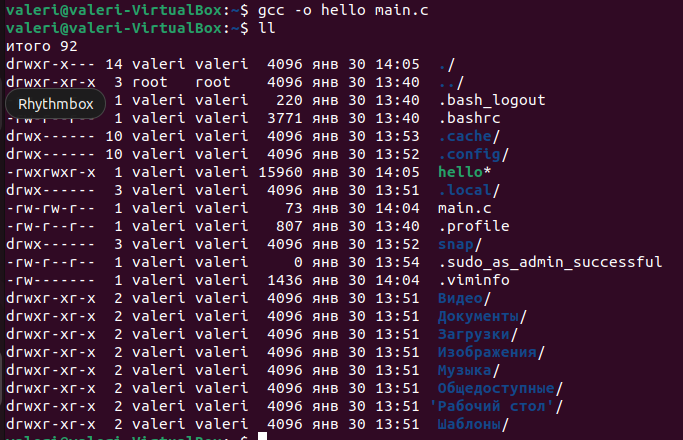


Напишем код программы:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Скомпилируем с помощью gcc в простой исполняемый файл:



Запустим и увидим, что все работает:



Создадим файл, содержащий функцию печати – hello.c

И создадим файл, содержащий прототип функции (интерфейсную часть) – hello.h :

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

hello.h :

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

hello.c :

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

И изменим содержание файла main.c :

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Скомпилируем файлы в объектный код с помощью команд «gcc –o hello.o –c hello.c ». Повторим те же действия с файлом main.c командой «gcc –o main.o –c main.c»

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Для того чтобы собрать исполняемый модуль – нужно написать следующую команду:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Проверим работу:



Все работает.

Научимся компилировать разделяемые библиотеки. С помощью компилятора выполним такую опцию, которая соберет не исполняемый файл, а динамическую библиотеку.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Так как мы работаем на 64-разрядной системе – нам необходимо указать флажок «-fPIC».

Теперь научимся использовать библиотеку.

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Но если мы начнем запускать программу, то у нас появится сообщение об ошибке. Поменяем переменную окружения LD\_LIBRARY\_PATH. Запустим программу снова и увидим, что все работает.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Разберем утилиту make, это утилита для управления зависимостями.

Сперва создадим Makefile:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Не забываем, чтобы выйти нажимаем «esc» и пишем «:wq».

Напишем просто «make» и увидим, как будет выполнена первая цель, которая приводит к компиляции программы.

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Если прописать «make clean» – сгенерированные файлы будут удалены:

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание

Прописав «make lib», мы увидим, что соберется только библиотека:

Изображение выглядит как текст, снимок экрана, меню

Автоматически созданное описание