**Занятие 2**

Cmake наиболее эффективный инструмент при работе со сторонними расширениями, пакетами, библиотеками.

В этой работе будет рассмотрено подключение сторонней библиотеки и компиляция проекта.

Для начала разберемся, что из себя представляет библиотека.

Библиотека – это сборник подпрограмм или объектов, используемых для разработки программного обеспечения. Иными словами, библиотеки – это уже готовый код, который программист может использовать в своих проектах.

С точки зрения операционной системы и прикладного ПО, библиотеки разделяются на динамические и статические.

**Статические библиотеки** представляют собой скомпилированный код, который встраивается в наш исполняемый файл в процессе линковки программы.

**Динамические библиотеки** представляют собой отдельные компоненты кода, как правило в виде отдельных файлов, поставляемых вместе с исполняемой программой. Загрузка динамической библиотеки в память происходит в процессе работы программы.

Распространённые форматы статических библиотек: .lib, .a.

Распространённые форматы динамических библиотек: .dll, .so, .dylib.

Когда мы разобрались с принципом работы библиотек и их видами, давайте попробуем подключить реальную библиотеку в наш проект на примере SDL2.

Задание

1. Скачайте библиотеку инструменты разработки SDL2-devel, а именно заголовочные файлы и библиотеку, поищите инструкцию по установке в интернете. Этот ресурс может быть полезен [Simple DirectMedia Layer - SDL version 2.0.14 (stable) (libsdl.org)](https://www.libsdl.org/download-2.0.php#source).
2. Основной код программы выглядит следующим образом, запишите его в файл с расширением .cpp или .c.

#include <SDL2/SDL.h>

const int SCREEN\_WIDTH = 640;

const int SCREEN\_HEIGHT = 480;

int main (int argc, char \*\* args) {

    if( SDL\_Init( SDL\_INIT\_EVERYTHING ) != 0 )

    {

        return 1;

    }

    SDL\_Surface\* screen\_surface = NULL;

    SDL\_Window\* window = NULL;

    window = SDL\_CreateWindow("Hello, SDL 2!",SDL\_WINDOWPOS\_UNDEFINED,

    SDL\_WINDOWPOS\_UNDEFINED, SCREEN\_WIDTH, SCREEN\_HEIGHT,

    SDL\_WINDOW\_SHOWN);

    if (window == NULL) {

        return 1;

    }

    screen\_surface = SDL\_GetWindowSurface(window);

    SDL\_FillRect(screen\_surface, NULL,

SDL\_MapRGB( screen\_surface->format, 0, 255, 0));

    SDL\_UpdateWindowSurface(window);

    SDL\_Delay(2000);

    SDL\_DestroyWindow(window);

    SDL\_Quit();

    return 0;

};

1. Приступим к заполнению CMakeLists.txt.

cmake\_minimum\_required(VERSION 3.17)

project(lesson2)

add\_executable(lesson2 test.cpp)

Для того чтобы указать в нашем проекте используемые библиотеки можно воспользоваться опцией target\_link\_libraries.

Эта опция ищет файлы наших библиотек в определенных директориях и линкует их с программой, если библиотека там есть.

Для MinGW под Windows добавьте строчку в CMakeLists.txt.

target\_link\_libraries(lesson2 mingw32 SDL2main SDL2)

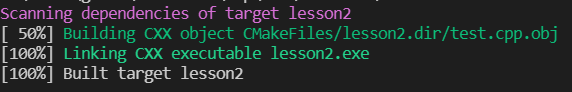
Мы указали библиотеки mingw32 SDL2main и SDL2 в качестве параметров, библиотека mingw32 нужна для правильной компиляции MinGW и может не применяться с другими компиляторами.

1. Произведем сборку последовательно выполнив команды



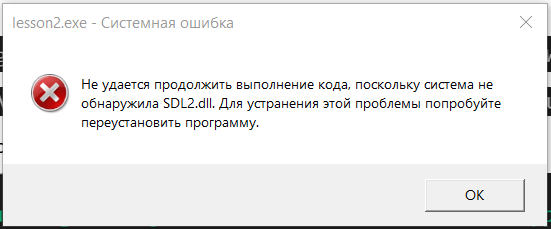


Сборка прошла успешно



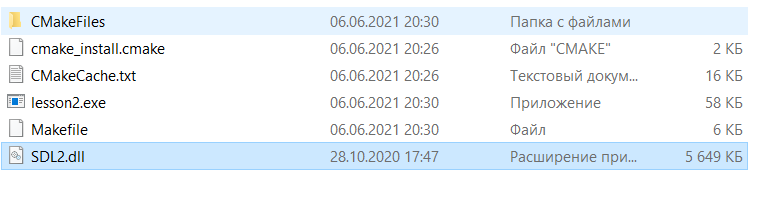
1. Запустите программу

Если вы запустили программу и увидели следующую ошибку, это значит, что при загрузке динамической библиотеки наша программа не смогла её найти.



Есть несколько способов решения данной проблемы: добавить dll файл в системную директорию, куда Windows обращается для поиска сторонних библиотек; добавить путь к dll файлу в system path; положить dll рядом с exe файлом.

Мы воспользуемся последним вариантом (если у вас нет SDL2.dll, его можно найти в интернете) и добавим файл в каталог, где находится lesson2.exe.



Если все было сделано правильно мы получим следующий результат:

