Собираем по-новому. CMake — конспект темы

Системы сборки и зачем они нужны

C++ — это международный стандарт, который поддерживается различными компиляторами на разных операционных системах. Вы знакомы как минимум с двумя основными:

«Большая тройка» компиляторов: GCC, Clang и Microsoft Visual C++ (доступен бесплатно в составе Microsoft Visual Studio Community).

IDE для работы с кодом на C++:

- Eclipse,
- Visual Studio,
- Visual Studio Code,
- CLion,
- QT creator,
- Code Blocks.

Преобразование из общей конфигурации в проекты разных IDE выполняют системы автоматизированной сборки. Одна из них — CMake.

Собираем CMake'ом

CMake сам собирать не может, нужен компилятор. Подойдёт любой современный. Для Windows используйте консоль MinGW с компилятором GCC.

Основы синтаксиса CMake

Переменная `CMAKE_CXX_STANDARD` определяет стандарт C++, влияет на параметры компиляции.

Переменная `CMAKE_BUILD_TYPE` определяет конфигурацию сборки.

Эти переменные заданы разными способами: одна через команду `set`, вторая через параметры командной строки. При задании через `set` переменную нужно указать первым параметром, а всё остальное содержимое скобок станет её значением:

```
# создаём переменную HELLO_WORLD. Её значением
# будет список из слов “Hello” и “world”
set(HELLO_WORLD Hello world)
```

По стайлгайду курса переменные CMake называются заглавными буквами.

Операция `${}` позволяет читать значения.

Команда `message` выводит значение в консоль:

```
set(HELLO_WORLD Hello world)

# Первый параметр команды message задаёт тип сообщения.
# STATUS указывает, что это простое информационное сообщение.
message(STATUS "Value of HELLO_WORLD: " ${HELLO_WORLD})
```

Команды из файла CMakeLists.txt выполняются на этапе генерации.

Команда `message` выводит свои аргументы без пробелов. Все типы сообщений — в документации по CMake.

Особенность CMake: если в команду подставляется список, его элементы становятся отдельными аргументами.

В CMake функция принимает название переменной и записывает в неё значение. Для работы со списками в CMake есть встроенная функция `list`.

Функция `string` поддерживает разнообразные операции со строками, поиск, замену, регулярные выражения и даже JSON.

Переменные кэша — это опции сборки вашей программы, которые должны быть заданы при генерации. Их поддерживают графические утилиты для работы с CMake, например cmake-gui. Описание для переменных кэша задавать обязательно.

Полный список типов смотрите в документации.

Операция `${}` позволяет получить значения переменных кэша.

В CMake `if` напоминает `if` из C++. Отличия: нет фигурных скобок после `if` и `else`, есть круглые скобки после `else`. Конец определяется командой `endif`, после неё ставят пустые круглые скобки:

```
if(MY_BEAUTIFUL_VARIABLE)
    message(STATUS "MY_BEAUTIFUL_VARIABLE exists and equals ${MY_BEAUTIFUL_VARIABLE}")
else()
```

```
    message(STATUS "MY_BEAUTIFUL_VARIABLE does not exist or equals OFF. Miss it.")
endif()
```

Это условие будет не выполнено, если переменная с таким именем существует, но равна `OFF` — аналогу `false` из C++. Подробнее о синтаксисе `if` читайте в документации. В CMake также есть циклы foreach и while.

Библиотеки: какие они бывают

Библиотека C++ — это пакет с бинарными и h-файлами. В типичной библиотеке такие каталоги или категории файлов:

- include — содержит h-файлы, подключаемые вашей программой через `#include`;
- lib — бинарные файлы, компонуемые с вашей программой;
- dll (под Windows) — динамические библиотеки;
- framework (под macOS) — пакет, содержащий в себе бинарные файлы и ресурсы.

Археологические раскопки: подключаем LibJPEG

Библиотека LibJPEG позволяет читать PPM и JPEG. Под Linux-системами можно установить уже готовый пакет с библиотекой, а под Windows — скачать готовый бинарный файл DLL или Lib.