



这是一个要填好久的坑了，背景主要在于将C/C++的ide从Dev-C++的过渡到了CLion，然后发现了巨大的不同，特别是CLion中有关.c, .cpp, .h等各类文件的链接极度依赖于CMakeList，因此记录一下相关的知识点。

CMakeList 学习

背景

CMakeList.txt是CMake的指导说明书吧，CMake是跨平台编译工具，比make更高级一些。其编译的主要工作是生成CMakeLists.txt文件，然后根据该文件生成Makefile，最后调用make来生成可执行程序或者动态库。所以基本步骤就只有两步：（1）cmake生成CMakeLists.txt文件；（2）make执行编译工作。

而CLion中我们关注于如何编写CMakeList即可

基础配置

设置项目版本

可以通过project配置项目信息

```
project(PROJECT_NAME VERSION 1.0.0 LANGUAGES C CXX)
```

VERSION对应的版本号应该实际为 `main.minor.patch.tweak`，方便在代码运行的时候了解当前版本号，并结合configure命令可以得到版本头文件，但感觉CLion中对此方面没有做要求，可以省略跳过。

制定语言版本

可以用set进行定制，如下

```
set(CMAKE_C_STANDARD 99)
set(CMAKE_CXX_STANDARD 11)
```

就是说，在编译C代码的时候，使用的是C99标准，编译C++的时候使用C++11标准。

这里设置的变量都是CMAKE_开头，这类变量都是CMake的内置变量，正是通过修改这些变量的值来配置CMake构建的行为。

CMake的内置变量一般有三种

- `_CMAKE`
- `_CMAKE`
- 下划线开头加上CMake命令的名称的变量名