

CMake技术报告

由于VScode那个神奇又奇怪的配置设置，我决定使用Cmake来构建并编译我的文件

关于Cmake的学习，我一般是看这个网站

[CMake-Cookbook](#)

这个网站讲的很详细但是我觉得这个网站的内容存在一些小问题或者我觉得还有一些东西用来补充

或者可以看官方教程

[CMake Tutorial](#)

[CMake Tutorial中文?](#)

也可以看b站的一个视频

[Cmake](#)

关于单文件的编译

单文件编译用以下代码就可以了

```
cmake_minimum_required(VERSION 3.15) # cmake最低支持的版本号，目前我的
Cmake版本为3.28，所以不要像我一样写个3.5啊！

set(CMAKE_C_STANDARD 99) # 设置C语言标准，默认为c11，而c++是c++11，可设置
可不设置
set(EXECUTABLE_OUTPUT_PATH ..) # 设置可执行文件输出路径，..表示输出到上一级
文件夹中，默认为输出到当前文件夹

project(Main LANGUAGES C) # 工程文件名，你想取什么名字呢？LANGUAGES C表示
支持编译C语言

add_executable(Main main.c) # 加入可执行文件，就是你要编译的文件，括号里的是
(文件名 可执行文件名)
```

要先在终端输入 `mkdir build` 创建文件夹build,再 `cd build` 将终端移动到文件夹，再执行 `cmake -G "Ninja" ..` 和 `cmake ..` 还有 `cmake --build .`

简单来说，就是：

```
mkdir build          # 创建文件夹build
cd build             # 将终端移动到文件夹
cmake -G "Ninja" ..  # 切换生成器为Ninja, ..的意思是CMakeLists.txt在上一级文件夹
cmake ..             # 加载cmake配置, ..的意思是CMakeLists.txt在上一级文件夹
cmake --build .      # 构建项目
```

切换生成器

这个没什么好说的，能用就行，我目前用的是ninja，因为我下了python，直接pip install ninja就能安装（主要还是这个名字好记，也可以用MinGW Makefiles）

创建静态库

比单编译多了两个指令

```
cmake_minimum_required(VERSION 3.15)

project(Main LANGUAGES C)

add_library(lwrb          # 静态库名
    STATIC               #表示为静态库
    #库中的文件，这里要注意，这里写的是什么，那么main中也要#include 什么，不要缺路径，否则会提示找不到文件
    src/include/lwrb/lwrb.h
    src/lwrb/lwrb.c
)

add_executable(Main main.c)

target_link_libraries(Main lwrb)    # 链接库，括号里为(可执行文件名 静态库名)
```

将STATIC改为SHARED就是创建动态库

用其他方式构建多文件编译

能够不通过构建库的方式编译
通过以下两行代码

```
cmake_minimum_required(VERSION 3.25)

project(Main LANGUAGES C)

set(C_STANDARD 11) # 设置C语言标准，默认为c11，而c++是c++11，可设置可不设置
set(EXECUTABLE_OUTPUT_PATH ..) # 设置可执行文件输出路径，..表示输出到上一级
文件夹中，默认为输出到当前文件夹

# ${CMAKE_CURRENT_SOURCE_DIR}表示CMakeLists.txt所在的绝对路径
file(GLOB SRC ${CMAKE_CURRENT_SOURCE_DIR}/src/*.c) # GLOB表示不递归搜
索，SRC表示要保存的变量，自己设，后面就是.c文件的路径，后面一定要加/*.c
include_directories(${CMAKE_CURRENT_SOURCE_DIR}/include) # 表示头文件在
哪个路径

add_executable(Main ${SRC} main.c ) # ${SRC}表示变量的值，这是c文件，同时
编译弄出可执行文件
```

这样的好处是 `#include " " 中""`只需要放头文件就行，不用放相对路径

暂时知道这么多