

Chapter1 cmake基础搭建使用

遵循以下步骤即可！

1. 进入终端，在合适的位置新建一个空文件夹：**mkdir myCmake**
2. 进入此文件夹：**cd myCmake**
3. 在文件夹中新建两个文档，按下述填充：**CMakeLists.txt** 和 **main.cpp**

CMakeLists.txt

```
cmake
1 | cmake_minimum_required(VERSION 3.5) # 设置CMake最小版本
2 | project (hello_cmake)               # 设置工程名
3 | add_executable(hello_cmake main.cpp) # 生成可执行文件
```

main.cpp

```
#include <iostream>

int main(int argc, char *argv[])
{
    std::cout << "Hello CMake!" << std::endl;
    return 0;
}
```

4. 对3的补充交代

1. CMake构建包含一个**项目名称**，project (**hello_cmake**)命令会自动生成一些变量，在使用多个项目时引用某些变量会更加容易。比如生成了：PROJECT_NAME 这个变量。PROJECT_NAME是变量名，\${PROJECT_NAME}是变量值，值为hello_cmake
2. add_executable()命令指定某些源文件生成**可执行文件**，本例中是**main.cpp**。add_executable () 函数的第一个参数是可执行文件名，第二个参数是要编译的源文件列表。
3. project(hello_cmake)函数执行时会生成一个变量，是PROJECT_NAME，\${PROJECT_NAME}表示PROJECT_NAME变量的值为hello_cmake,所以把\${PROJECT_NAME}用在add_executable()里可以生成可执行文件名字叫hello_cmake

一般格式：

- cmake版本说明
- project(项目名字/可执行文件名)
- add_executable(可执行文件名 待编译的列表中文件)

5. 外部构建

1. 在myCmake文件夹中 mkdir build
2. cd build
3. cmake ..
4. 搭建完毕！

6. 总览全局

- 输入：tree
- 显示如下：

```

.
├── CMakeLists.txt
├── build
│   ├── CMakeCache.txt
│   ├── CMakeFiles
│   │   ├── 3.25.1
│   │   │   ├── CMakeCCompiler.cmake
│   │   │   ├── CMakeCXXCompiler.cmake
│   │   │   ├── CMakeDetermineCompilerABI_C.bin
│   │   │   ├── CMakeDetermineCompilerABI_CXX.bin
│   │   │   ├── CMakeSystem.cmake
│   │   │   ├── CompilerIdC
│   │   │   │   ├── CMakeCCompilerId.c
│   │   │   │   ├── a.out
│   │   │   │   └── tmp
│   │   │   ├── CompilerIdCXX
│   │   │   │   ├── CMakeCXXCompilerId.cpp
│   │   │   │   ├── a.out
│   │   │   │   └── tmp
│   │   ├── CMakeDirectoryInformation.cmake
│   │   ├── CMakeOutput.log
│   │   ├── CMakeScratch
│   │   ├── Makefile.cmake
│   │   ├── Makefile2
│   │   ├── TargetDirectories.txt
│   │   ├── cmake.check_cache
│   │   ├── hello_cmake.dir
│   │   │   ├── DependInfo.cmake
│   │   │   ├── build.make
│   │   │   ├── cmake_clean.cmake
│   │   │   ├── compiler_depend.make
│   │   │   ├── compiler_depend.ts
│   │   │   ├── depend.make
│   │   │   ├── flags.make
│   │   │   ├── link.txt
│   │   │   └── progress.make
│   │   ├── pkgRedirects
│   │   └── progress.marks
│   ├── Makefile
│   └── cmake_install.cmake
└── main.cpp

```

- 可以看到，build文件夹下生成了许多二进制文件，如果要从头开始重新创建cmake环境，只需删除构建目录build，然后重新运行cmake即可
- 事实上还有内部搭建，但是不推荐，从略！