1. 通过project命令配置项目信息

project(CMakeTemplate VERSION 1.0.0 LANGUAGES C CXX)

项目名：CmakeTemplate

版本号

语言

1. 设定编程语言版本

set(CMAKE\_C\_STANDARD 99)

set(CMAKE\_CXX\_STANDARD 11)

1. 配置编译类型

set(CMAKE\_BUILD\_TYPE Debug)

1. 添加include目录

include\_directories(src/c)

1. 编译静态库

file(GLOB\_RECURSE MATH\_LIB\_SRC

src/c/math/\*.c

)

add\_library(math STATIC ${MATH\_LIB\_SRC})

使用file命令获取src/c/math目录下所有的\*.c文件，然后通过add\_library命令编译名为math的静态库，库的类型是第二个参数STATIC指定的。

如果指定为SHARED则编译的就是动态链接库。

1. 安装

install(TARGETS math demo

RUNTIME DESTINATION bin

LIBRARY DESTINATION lib

ARCHIVE DESTINATION lib)

这里通过TARGETS参数指定需要安装的目标列表；参数RUNTIME DESTINATION、LIBRARY DESTINATION、ARCHIVE DESTINATION分别指定可执行文件、库文件、归档文件分别应该安装到安装目录下个哪个子目录。

如果指定CMAKE\_INSTALL\_PREFIX为/usr/local，那么math库将会被安装到路径/usr/local/lib/目录下；而demo可执行文件则在/usr/local/bin目录下。

同时，还可以使用install命令安装头文件：

file(GLOB\_RECURSE MATH\_LIB\_HEADERS src/c/math/\*.h)

install(FILES ${MATH\_LIB\_HEADERS} DESTINATION include/math)

**编译知识：**

1. 编译后生成的文件类型：

.obj：目标文件，obj文件与cpp文件名字一一对应；

1. 编译的主要过程：

* 编译：把文本形式的源代码翻译成机器语言，并形成目标文件。编译可细分为三个阶段：预处理/预编译、编译优化、汇编；
* 预处理/预编译：展开头文件，替换宏定义，处理条件编译指令#ifdef等，输出.i文件；
* 编译优化：预编译输出的.i文件中，只有常量：数字、字符串、变量的定义，以及关键字：main、if、else、for、while等。这阶段要做的工作主要是，通过语法分析和词法分析，确定所有指令是否符合规则，之后翻译成汇编代码。这个过程将.i文件转化位.s文件；
* 汇编：汇编过程就是把汇编语言翻译成目标机器指令的过程，生成目标文件（.obj .o等）。目标文件中存放的也就是与源程序等效的目标的机器语言代码。目标文件由段组成，通常至少有两个段：代码段：包换主要程序的指令。该段是可读和可执行的，一般不可写；数据段：存放程序用到的全局变量或静态数据。可读、可写、可执行。这个过程将.s文件转化成.o文件。
* 链接：把目标文件、操作系统的启动代码和库文件组织起来形成可执行程序；由汇编程序生成的目标文件并不能立即就执行，还要通过链接过程。原因：
* 某个源文件调用了另一个源文件中的函数或常量
* 在程序中调用了某个库文件中的函数

链接程序的主要工作就是将有关的目标文件连接起来。这个过程将.o文件转化成可执行的文件。

**库相关知识：**

1. 什么是库：

库是写好的，现有的，成熟的，可以复用的代码。本质上来说，库是一种可执行代码的二进制形式，可以被操作系统载入内存执行。库有两种：静态库（.a、.lib）和动态库（.so、.dll）。所谓静态、动态是指链接。库文件是事先编译好的方法的合集。

1. 二者的区别：

静态库在编译时会直接整合到目标程序中，编译成功的可执行文件可独立运行；动态库在编译时不会放到链接的目标程序中，即可执行文件无法单独运行。静态库和动态库最本质的区别就是：该库是否被编译进目标（程序）内部。

参考链接：

https://blog.csdn.net/weixin\_51483516/article/details/120837316