

# Cmake用法简介

---

## 一个简单例子

### 文档树

```
./Demo
|
+--- main.cc
|
+--- math/
    |
    +--- MathFunctions.cc
    |
    +--- MathFunctions.h
```

### 各代码段内容

- main.cpp

```
#include <stdio.h>
#include <iostream>
#include <stdlib.h>
#include "config.h"

#ifdef USE_MYMATH
    #include "MathFunctions.h"
#else
    #include <math.h>
#endif

using namespace std;

int main(int argc, char *argv[])
{
    if (argc < 3){
        printf("%s Version %d.%d\n",
            argv[0],
            Demo_VERSION_MAJOR,
            Demo_VERSION_MINOR);
        printf("Usage: %s base exponent \n", argv[0]);
        return 1;
    }
    double base = atof(argv[1]);
    int exponent = atoi(argv[2]);

    #ifdef USE_MYMATH
        printf("Now we use our own Math library. \n");
        double result = power(base, exponent);
    #else
        printf("Now we use the standard library. \n");
        double result = pow(base, exponent);
    #endif

    printf("%g ^ %d is %g\n", base, exponent, result);
    return 0;
}
```

- MathFunctions.cpp

```
#include "MathFunctions.h"
double power(double base, int exponent)
```

```
{
    int result = base;
    int i;

    if (exponent == 0) {
        return 1;
    }

    for(i = 1; i < exponent; ++i){
        result = result * base;
    }
    return result;
}
```

- MathFunction.h

```
double power(double base, int exponent);
```

- 简单说明
  - 以上代码是一个能实现Cmake基础功能的简单版本
  - mian函数中Demo\_VERSION\_MAJOR和Demo\_VERSION\_MINOR表示代码的主版本号 and 副版本号，用户版本控制
  - MathFunction中实现了一个power的函数，用作利用Cmake实现调库功能的示例

## CMake实现（注：CMakeLists.txt用#注释）

- 顶层CMakeLists.txt

```
cmake_minimum_required(VERSION 2.6) #表示使用的Cmake的最低版本

project(Demo7) #设置project名
set(Demo_VERSION_MAJOR 1) #设置主版本号
set(Demo_VERSION_MINOR 0) #设置副版本号

configure_file(
    "${PROJECT_SOURCE_DIR}/config.h.in" #设置配置文件
    "${PROJECT_BINARY_DIR}/config.h" #config.h由config.h.in生成
)

option(USE_MYMATH
    "Use provided math implementation" ON) #设置可选项，若为ON，则表示当前使用的是自己编写的库

if(USE_MYMATH)
    include_directories("${PROJECT_SOURCE_DIR}/math") #若USE_MYMATH为ON，则添加MathFunctions库
    add_subdirectory(math)
    set(EXTRA_LIBS ${EXTRA_LIBS} MathFunctions)
endif(USE_MYMATH)

aux_source_directory(. DIR_SRCS) #搜索当前目录下的所有源文件，把名字赋值给DIR_SRCS

add_executable(Demo ${DIR_SRCS}) #链接源文件，生成可执行文件Demo

target_link_libraries(Demo ${EXTRA_LIBS}) #链接EXTRA_LIBS到Demo里

install (TARGETS Demo DESTINATION bin) #生成makefile之后只用make install时调用
install (FILES "${PROJECT_BINARY_DIR}/config.h" DESTINATION include) #用于安装程序到制定路径下

enable_testing() #打开调试功能，生成makefile之后使用make test调用

# add_test(test_run Demo 5 2)
#add_test(test_usage Demo)
#set_tests_properties(test_usage PROPERTIES PASS_REGULAR_EXPRESSION "Usage: .* base exponent")
#add_test(test_5_2 Demo 5 2)
#set_tests_properties(test_5_2 PROPERTIES PASS_REGULAR_EXPRESSION "is 25")
#add_test(test_10_5 Demo 10 5)
```

```

#set_tests_properties(test_10_5 PROPERTIES PASS_REGULAR_EXPRESSION "is 100000")
#add_test(test_2_10 Demo 2 10)
#set_tests_properties(test_2_10 PROPERTIES PASS_REGULAR_EXPRESSION "is 1024")

macro(do_test arg1 arg2 result)          #使用宏定义, 简化测试代码编写
    add_test(test_${arg1}_${arg2} Demo ${arg1} ${arg2})
    set_tests_properties(test_${arg1}_${arg2} PROPERTIES PASS_REGULAR_EXPRESSION ${result})
endmacro(do_test)
do_test(5 2 "is 25")
do_test(10 5 "is 100000")
do_test(2 10 "is 1024")

set(CMAKE_BUILD_TYPE "Debug")           #使用Debug模式, 添加gdb调试功能
set(CMAKE_CXX_FLAGS_DEBUG "$ENV{CXXFLAGS} -O0 -Wall -g -ggdb")
set(CMAKE_CXX_FLAGS_RELEASE "$ENV{CXXFLAGS} -O3 -Wall")

include (InstallRequiredSystemLibraries) #用于生成安装包
set (CPACK_RESOURCE_FILE_LICENSE         #使用cpack -C CPackConfig.cmake命令生成二进制安装包
    "${CMAKE_CURRENT_SOURCE_DIR}/License.txt") #使用cpack -C CPackSourceConfig.cmake生成源码安装包
set (CPACK_PACKAGE_VERSION_MAJOR "${Demo_VERSION_MAJOR}") #会在该目录下创建三个不同格式的安装包
set (CPACK_PACKAGE_VERSION_MINOR "${Demo_VERSION_MINOR}") #安装示例: sh Demo8-1.0.1-Linux.sh
include (CPack) #安装完成后会生成对应子目录, 即可正常使用

```

- math子目录下CMakeLists.txt

```

aux_source_directory(. DIR_LIB_SRCS)

add_library(MathFunctions ${DIR_LIB_SRCS})

install (TARGETS MathFunctions DESTINATION bin) #make install使用
install (FILES MathFunctions.h DESTINATION include) #make install使用

```

- config.h.in

```

#cmakedefine USE_MYMATH           #是否调用MathFunctions库
#define Demo_VERSION_MAJOR @Demo_VERSION_MAJOR@ #获取版本号
#define Demo_VERSION_MINOR @Demo_VERSION_MINOR@

```

- Liccese.txt

```

The MIT License (MIT)           #安装时需要用到的协议= =

Copyright (c) 2013 Joseph Pan(http://hahack.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```