# Cmake简介

cmake 是一个跨平台、开源的构建系统。它是一个集软件构建、测试、打包于一身的软件。它使用与平台和编译器独立的配置文件来对软件编译过程进行控制。

# 常用指令

## 指定cmake最小版本

* 命令

cmake\_minimum\_required(VERSION 3.0)；

* 使用详解

可选;最好加上（CMakeLists.txt 文件中使用了一些高版本cmake特有的一些命令的时候，就需要加上这样一行，提醒用户升级到该版本之后再执行 cmake。）

## 设置项目名称

* 命令

project(demo)；

* 使用详解

可选;最好加上（它会引入两个变量：demo\_BINARY\_DIR和demo\_SOURCE\_DIR ; 同时Cmake自动定义了两个等价的变量： PROJECT\_BINARY\_DIR 和 PROJECT\_SOURCE\_DIR ）

## 设置编译类型

* 命令

add\_executable(demo demo.cpp) # 生成可执行文件

add\_library(common STATIC util.cpp) # 生成静态库

add\_library(common SHARED util.cpp) # 生成动态库或共享库

* 使用详解

add\_library 默认生成是静态库，通过以上命令生成文件名字:

在 Linux 下是：

demo

libcommon.a

libcommon.so

在 Windows 下是：

demo.exe

common.lib

common.dll

## 指定编译包含的源文件

明确指定包含哪些源文件

add\_library(demo demo.cpp test.cpp util.cpp)

搜索所有的cpp文件

* 命令

aux\_source\_directory( . SRC\_LIST ) # 搜索当前目录下的所有.cpp文件

add\_library(demo ${SRC\_LIST})

* 使用详解

aux\_source\_directory(dir VAR) ： 发现一个目录dir下所有的源代码文件，并将列表存储在一个变量VAR中。

自定义搜索规则

* 命令

file(GLOB SRC\_LIST "\*.cpp" "protocol/\*.cpp")

add\_library(demo ${SRC\_LIST})

# 或者

file(GLOB SRC\_LIST "\*.cpp")

file(GLOB SRC\_PROTOCOL\_LIST "protocol/\*.cpp")

add\_library(demo ${SRC\_LIST} ${SRC\_PROTOCOL\_LIST})

# 或者

aux\_source\_directory(. SRC\_LIST)

aux\_source\_directory(protocol SRC\_PROTOCOL\_LIST)

add\_library(demo ${SRC\_LIST} ${SRC\_PROTOCOL\_LIST})

* 使用详解

file(GLOB files LIST\_DIRECTORIES false \*) //收集文件指令

GLOB命令将所有匹配的文件挑选出来，默认以字典顺序排序；

## 查找指定的库文件

* 命令

find\_library( log-lib log )

* 使用详解

find\_library(var name 预编译库名) 查找到指定的预编译库，并将它的路径存储在变量中；默认的搜索路径为cmake包含的系统库，因此如果是NDK的公共库只需要指定库的name即可；

类似的还有：find\_file()/find\_path()/find\_program()/find\_package() ；

## 设置包含目录

* 命令

include\_directories

(

${CMAKE\_CURRENT\_SOURCE\_DIR}

${CMAKE\_CURRENT\_BINARY\_DIR}

${CMAKE\_CURRENT\_SOURCE\_DIR}/include

)

* 使用详解

Linux 下还可以通过如下方式设置包含的目录:

set( CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} -I${CMAKE\_CURRENT\_SOURCE\_DIR}" )

## 设置链接库搜索目录

* 命令

link\_directories(${CMAKE\_CURRENT\_SOURCE\_DIR}/libs)

* 使用详解

Linux 下还可以通过如下方式设置链接库搜索的目录:

set(CMAKE\_CXX\_FLAGS "${CMAKE\_CXX\_FLAGS} -L${CMAKE\_CURRENT\_SOURCE\_DIR}/libs")

# 常用变量