[**Makefile条件编译debug版和release版**](https://www.cnblogs.com/caosiyang/archive/2012/06/13/2548051.html)

一般，在开发测试阶段用debug版本，而上线发布用release版本。

使用Makefile定制编译不同版本，避免修改程序和Makefile文件，将会十分方便。

读了一些资料，找到一个解决方法，Makefile预定义宏与条件判断，结合make预定义变量，进行条件编译。

比如，有一个test.cpp，包含这段代码

#ifdef debug

//your code

#endif

你希望在debug版本要执行它，在release版本不执行。

我们可以写这样的一个Makefile：

[复制代码](javascript:void(0);)

1 ver = debug

2

3 ifeq ($(ver), debug)

4 ALL: test\_d

5 CXXFLAGS = -c -g -Ddebug

6 else

7 ALL: test\_r

8 CXXFLAGS = -c -O3

9 endif

10

11 test\_d: test.do

12 g++ -o $@ $^

13

14 test\_r: test.ro

15 g++ -o $@ $^

16

17 %.do: %.cpp

18 g++ $(CXXFLAGS) $< -o $@

19

20 %.ro: %.cpp

21 g++ $(CXXFLAGS) $< -o $@

[复制代码](javascript:void(0);)

简单说一下，Makefile根据ver的不同定义了不同的编译选项CXXFLAGS与输出程序ALL，

debug版本输出程序是test\_d，release版本输出程序是test\_r

debug版本编译选项是"-c -g -Ddebug"，release版本编译选项是"-c -O3"

debug版本object文件后缀是".do"，release版本object文件后缀是".ro"

debug版本编译选项使用"-D"定义宏debug，使得your code能够执行。

不同版本的编译选项、object文件、输出程序均不同，所以可以同时编译两个版本的程序，互不影响。

Makefile执行时，首先判断ver变量，如果ver的值是debug，编译debug版，否则编译release版。当然，默认情况下是编译debug版的。

如果想编译release版，要怎么做？

只要在执行make时，对ver变量赋值，使得ver的值不为debug，比如

# make ver=release

# 如何判断是linux/windows库，module或程序debug还是release

2013年08月29日 09:45:24 [legendmaker](https://me.csdn.net/legendmaker) 阅读数：1087

**linux：**

readelf -S xxx.a |grep "\.debug\*"

如果有输出, 是debug, 没有是release   
   
 

**windows：**

dumpbin /DEPENDENTS xx.dll, 结果输出中有MSVCRTD.dll的是debug版本. 为了脚本自动判断, 可用powergrep等工具自动分拣. .lib和我们的编译选项有关, 通过dumpbin /HEADERS 也可以判断出来, 具体要看编译选项了.