**在Win7 x64中使用GNUstep和MinGW编写Objective-C的代码**

**本文意义：帮助那些买不起MAC却又想学Objective-C语言的穷屌丝迅速搭建免费学习环境，努力把手中的windows剩余价值榨取尽量干^\_^\***

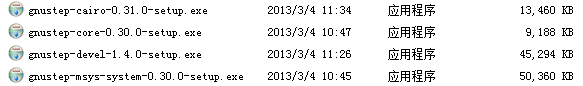
1、下载并安装所需文件：

http://www.gnustep.org/experience/Windows.html

下载以下所有文件：

| **Package** | **Required?** | **Stable** | **Unstable** | **Notes** |
| --- | --- | --- | --- | --- |
| **GNUstep MSYS System** | *Required* | [**0.30.0**](http://ftpmain.gnustep.org/pub/gnustep/binaries/windows/gnustep-msys-system-0.30.0-setup.exe) | **-** | MSYS/MinGW System |
| **GNUstep Core** | *Required* | [**0.30.0**](http://ftpmain.gnustep.org/pub/gnustep/binaries/windows/gnustep-core-0.30.0-setup.exe) | [**0.31.0**](http://ftpmain.gnustep.org/pub/gnustep/binaries/windows/gnustep-core-0.31.0-setup.exe) | GNUstep Core |
| **GNUstep Devel** | *Optional* | [**1.4.0**](http://ftpmain.gnustep.org/pub/gnustep/binaries/windows/gnustep-devel-1.4.0-setup.exe) | **-** | Developer Tools |
| **GNUstep Cairo** | *Optional* | **-** | [**0.31.0**](http://ftpmain.gnustep.org/pub/gnustep/binaries/windows/gnustep-cairo-0.31.0-setup.exe) | Cairo Backend |

下载后在文件夹中的安装包：



2、安装文件到一个统一目录：

比如我把所有安装在：

D:\program\GNUstep

安装好后目录里应该有

D:\program\GNUstep\msys\1.0\msys.bat

创建桌面快捷方式

3、添加源代码和makefile文件：

D:\program\GNUstep\msys\1.0\home\Administrator\

创建hello.m文件：

**[cpp]** [view plaincopy](http://blog.csdn.net/sonikk/article/details/8633465)

1. #import <Foundation/Foundation.h>
3. **int** main (**int** argc, **const** **char** \* argv[])
4. {
5. NSAutoreleasePool \* pool = [[NSAutoreleasePool alloc] init];
6. NSLog (@"hello world");
7. [pool drain];
8. **return** 0;
9. }

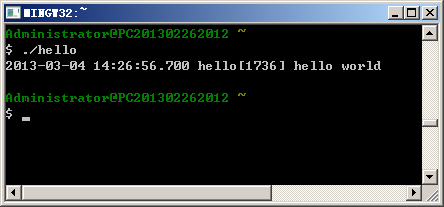
4、编译运行helloWorld：

$ cd /home/Administrator

$ gcc `gnustep-config --objc-flags` -L /GNUstep/System/Library/Libraries hello.m -o hello -lgnustep-base -lobjc

$ ./hello

输出如下：



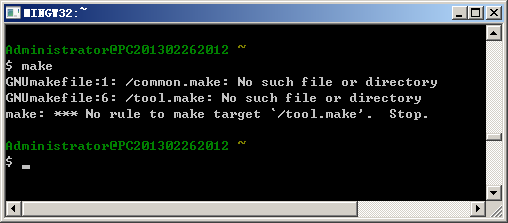
5、为了似的编译更简单，引入makefile工具：

创建GNUmakefile文件：

**[plain]** [view plaincopy](http://blog.csdn.net/sonikk/article/details/8633465)

1. include $(GNUSTEP\_MAKEFILES)/common.make
2. TOOL\_NAME = hello
3. hello\_OBJC\_FILES = hello.m
4. include $(GNUSTEP\_MAKEFILES)/tool.make

$ make



产生了错误：

GNUmakefile:1: /common.make: No such file or directory  
GNUmakefile:6: /tool.make: No such file or directory  
make: \*\*\* No rule to make target `/tool.make'.  Stop.

$ echo $(GNUSTEP\_MAKEFILES)

输出空，所以问题是环境变量未找到问题。

6、解决目录找不到的问题，并编译

安装目录：D:\program\GNUstep\msys\1.0\home\Administrator

查找目录：D:\program\GNUstep\GNUstep\System\Library\Makefiles

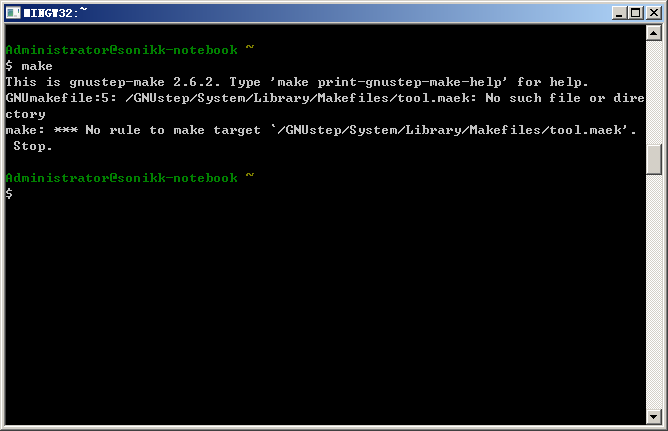
所以$(GNUSTEP\_MAKEFILES) = /../../GNUstep/System/Library/Makefiles

修改GNUmakefile文件为：

**[plain]** [view plaincopy](http://blog.csdn.net/sonikk/article/details/8633465)

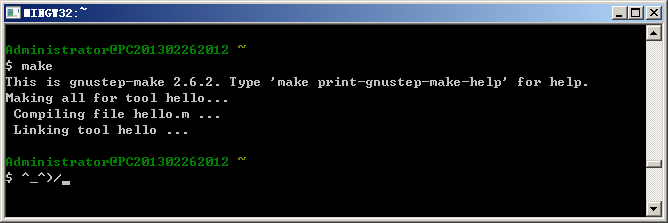
1. GNUSTEP\_MAKEFILES = /../../GNUstep/System/Library/Makefiles
2. include $(GNUSTEP\_MAKEFILES)/common.make
3. TOOL\_NAME = hello
4. hello\_OBJC\_FILES = hello.m
5. include $(GNUSTEP\_MAKEFILES)/tool.make

$ make



This is gnustep-make 2.6.2. Type 'make print-gnustep-make-help' for help.  
GNUmakefile:5: /GNUstep/System/Library/Makefiles/tool.maek: No such file or directory  
make: \*\*\* No rule to make target `/GNUstep/System/Library/Makefiles/tool.maek`.  
Stop.

如果出现以上信息，则说明查找目录不对，为什么呢？因为使用的磁盘格式为FAT32，应当将GNUstep卸载重新安装在NTFS格式的磁盘上：



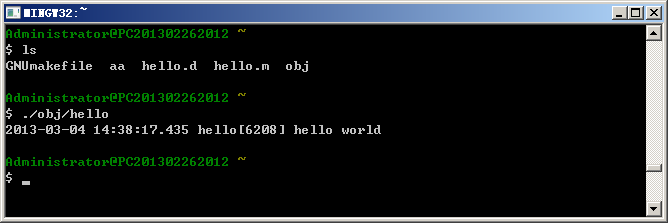
正确执行GNUmakefile

$ ls

你会看到多生成了一个目录叫obj

$ ./obj/hello

> 2013-03-04 14:38:17.435 hello[6208] hello world



OK，恭喜你大功告成！

-更新：2013-3-22 18:15:05

如果是多个.h和.m文件被引用，那就变成了一个工程，对于一个工程来说，写单条的编译语句就不再方便了，这时GNUmakefile就有用武之地了：

如果main函数所在的文件叫main.m

main.m当中又通过 #import "test1.h" 引用了一个类，

类的文件结构是test1.h test1.m，并且和main.m放在同一目录下

修改GNUmakefile内容如下：

**[plain]** [view plaincopy](http://blog.csdn.net/sonikk/article/details/8633465)

1. GNUSTEP\_MAKEFILES = /../../GNUstep/System/Library/Makefiles
3. include $(GNUSTEP\_MAKEFILES)/common.make
5. # 编译生成的目标
6. TOOL\_NAME = main
8. # 头文件
9. main\_HEARDER\_FILES = test1.h
11. # 源文件
12. main\_OBJC\_FILES = main.m test1.m
13. # main\_OBJC\_FILES = $(mycar\_HEARDER\_FILES:.h=.m) main.m
15. include $(GNUSTEP\_MAKEFILES)/tool.make

$ make

再次ok ~

-更新：2013-3-22 19:52:06

给MingW添加中文支持：

(添加中文的显示)

GNUstep\msys\1.0\etc\profile  在末尾添加如下几行：

**[plain]** [view plaincopy](http://blog.csdn.net/sonikk/article/details/8633465)

1. #export PS1='[\u@msys \W]$ '
2. export PS2='> '
3. export PS4='+ '
4. export LANG=en
6. alias l='/bin/ls --show-control-chars --color=tty'
7. alias la='/bin/ls -aF --show-control-chars --color=tty'
8. alias ll='/bin/ls -alF --show-control-chars --color=tty'
9. alias ls='/bin/ls --show-control-chars -F --color=tty'

(添加中文的输入)

GNUstep\msys\1.0\etc\inputrc.default  对应部分替换为：

**[plain]** [view plaincopy](http://blog.csdn.net/sonikk/article/details/8633465)

1. # disable/enable 8bit input
2. set meta-flag on
3. set input-meta on
4. set output-meta on
5. set convert-meta off

(添加中文的输入)  
GNUstep\msys\1.0\home\Administrator\.inputrc  对应部分替换为：

**[plain]** [view plaincopy](http://blog.csdn.net/sonikk/article/details/8633465)

1. # disable/enable 8bit input
2. set meta-flag on
3. set input-meta on
4. set output-meta on
5. set convert-meta off