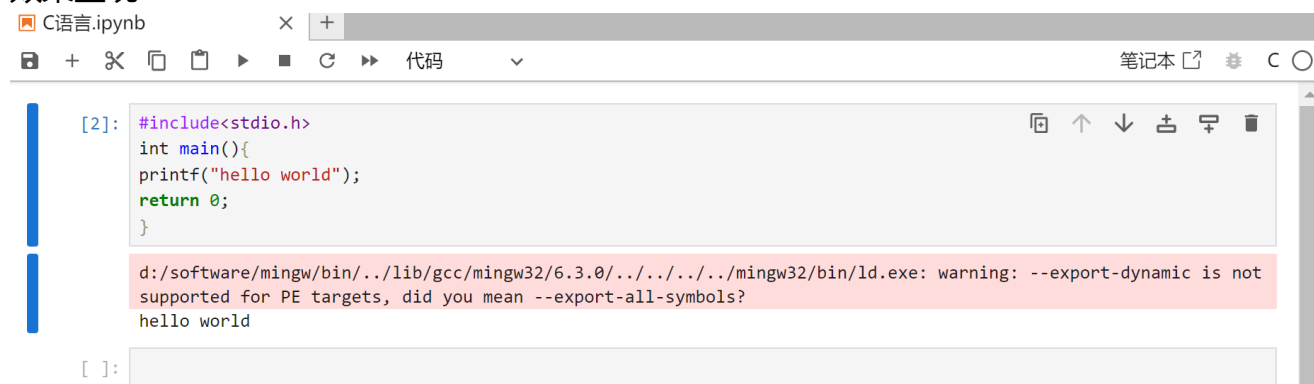


【教程】Windows系统上在jupyter上搭建C语言环境

众所周知，jupyter主要用于写python代码，但是我们可以通过更换核心的方式来写C语言代码。

效果呈现：



```
[2]: #include<stdio.h>
int main(){
printf("hello world");
return 0;
}

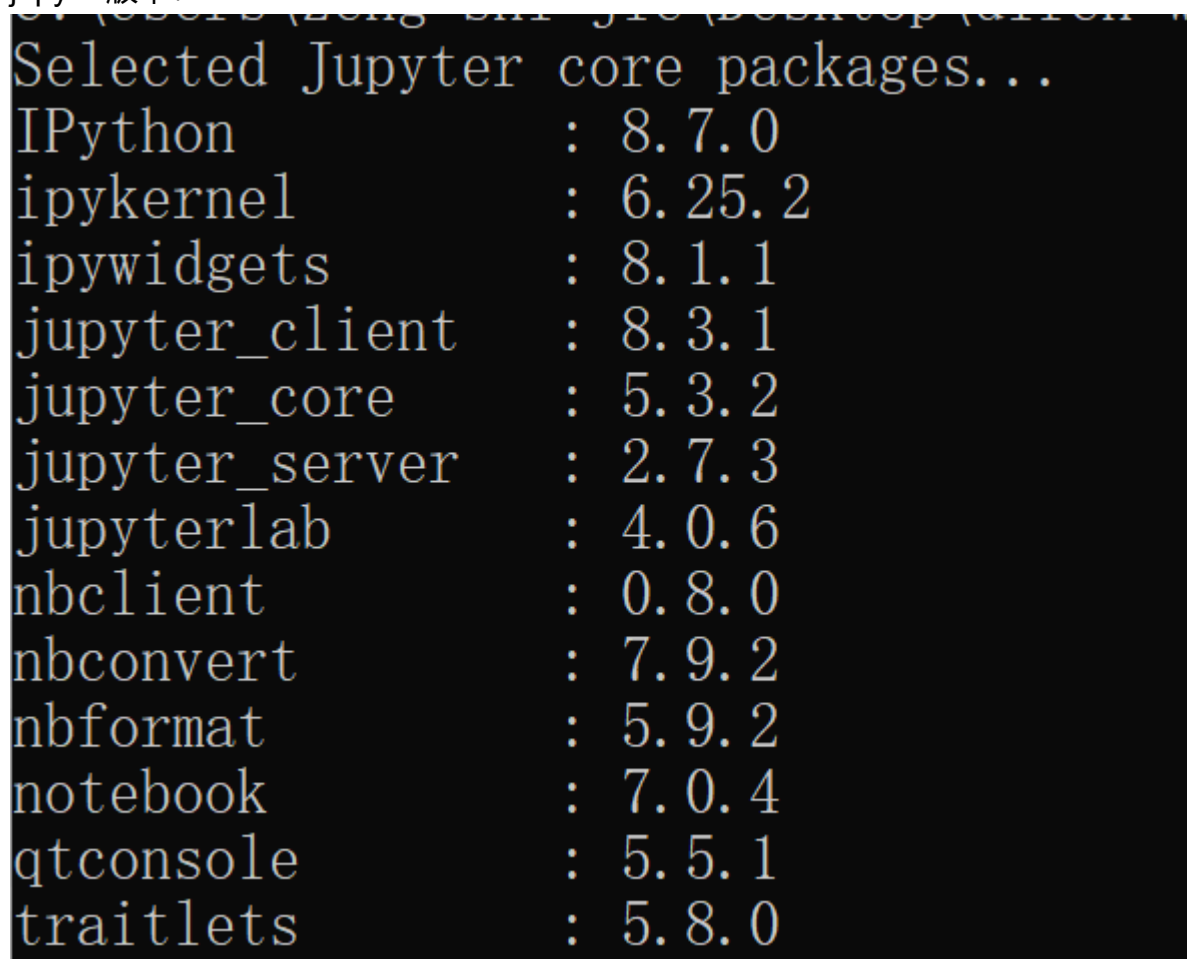
d:/software/mingw/bin/./lib/gcc/mingw32/6.3.0/./.././../mingw32/bin/ld.exe: warning: --export-dynamic is not supported for PE targets, did you mean --export-all-symbols?
hello world

[ ]:
```

其中的警告不清楚如何去除，但是能用。

一、环境介绍

jupyter版本：



```
Selected Jupyter core packages...
IPython                : 8.7.0
ipykernel               : 6.25.2
ipywidgets              : 8.1.1
jupyter_client          : 8.3.1
jupyter_core            : 5.3.2
jupyter_server          : 2.7.3
jupyterlab              : 4.0.6
nbclient                : 0.8.0
nbconvert               : 7.9.2
nbformat                : 5.9.2
notebook                : 7.0.4
qtconsole               : 5.5.1
traitlets               : 5.8.0
```

gcc版本（C编译器）：

```
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=d:/software/mingw/bin/../../liblto
Target: mingw32
Configured with: ../src/gcc-6.3.0/configure --build=x86_64-w64-mingw32 --with-mpfr --with-mpc=/mingw --with-isl=/mingw
generic --enable-languages=c,c++,objc,obj-c++,fortran --enable-shared --enable-threads --with-dwarf2 --disable-lto
--with-libintl-prefix=/mingw --with-libintl-prefix=/mingw --with-libintl-prefix=/mingw
Thread model: win32
gcc version 6.3.0 (MinGW.org GCC-6.3.0-1)
```

这里使用的是MinGW (Minimalist GNU for Windows) 来下载构建windows平台上的GCC















二、指引

注意：本指引并不介绍jupyter的安装说明，也不清楚如何在linux上构建这个环境。

安装MinGW

下载[cygwin版的MinGW](#)

在确认好安装路径之后进行安装，安装完之后进入到MinGW\bin路径下，找到

Data (D:) > software > mingw > bin				在 bin 中搜索
<div> <div> <div></div> <div></div> <div></div> </div> <div> <div>排序</div> <div>查看</div> <div></div> </div> </div>				
名称	修改日期	类型	大小	
 libisl-15.dll	2017/5/30 6:02	DLL 文件	2,067 KB	
 libmingwex-0.dll	2017/12/7 4:09	DLL 文件	165 KB	
 libmpc-3.dll	2017/5/26 20:39	DLL 文件	106 KB	
 libmpfr-4.dll	2018/1/18 5:11	DLL 文件	370 KB	
 libquadmath-0.dll	2017/5/30 5:00	DLL 文件	472 KB	
 libssp-0.dll	2017/5/30 5:00	DLL 文件	15 KB	
 mingw32-gcc.exe	2017/7/25 1:03	应用程序	973 KB	
 mingw32-gcc-6.3.0.exe	2017/7/25 1:03	应用程序	973 KB	
 mingw32-gcc-ar.exe	2017/7/25 1:03	应用程序	70 KB	
 mingw32-gcc-nm.exe	2017/7/25 1:03	应用程序	70 KB	
 mingw32-gcc-ranlib.exe	2017/7/25 1:03	应用程序	70 KB	
 mingw-get.exe	2013/10/5 3:20	应用程序	51 KB	
 mingwm10.dll	2017/12/7 4:09	DLL 文件	11 KB	
 nm.exe	2017/5/22 18:18	应用程序	957 KB	

这个exe可执行文件。

安装GCC和MSYS

在[当前目录](#)下右箭打开终端/命令提示符，执行

```
mingw-get.exe install gcc
```

```
PS D:\software\mingw\bin> mingw-get.exe install gcc
install: gcc-6.3.0-1-mingw32-lic.tar.xz
mingw-get.exe: *** ERROR *** package gcc-6.3.0-1-mingw32-lic.tar.xz is already installed
install: gcc-core-4.8.2-mingw32-dev.tar.meta
mingw-get.exe: *** ERROR *** package gcc-core-4.8.2-mingw32-dev.tar.meta is already installed
install: gcc-core-6.3.0-1-mingw32-bin.tar.xz
mingw-get.exe: *** ERROR *** package gcc-core-6.3.0-1-mingw32-bin.tar.xz is already installed
install: gcc-core-4.8.1-4-mingw32-doc.tar.lzma
mingw-get.exe: *** ERROR *** package gcc-core-4.8.1-4-mingw32-doc.tar.lzma is already installed
install: gcc-6.3.0-1-mingw32-man.tar.xz
mingw-get.exe: *** ERROR *** package gcc-6.3.0-1-mingw32-man.tar.xz is already installed
install: gcc-6.3.0-1-mingw32-info.tar.xz
mingw-get.exe: *** ERROR *** package gcc-6.3.0-1-mingw32-info.tar.xz is already installed
install: gcc-6.3.0-1-mingw32-lang.tar.xz
mingw-get.exe: *** ERROR *** package gcc-6.3.0-1-mingw32-lang.tar.xz is already installed
PS D:\software\mingw\bin> |
```

以及

```
mingw-get.exe install msys
```

```
PS D:\software\mingw\bin> mingw-get.exe install msys
install: msys-base-2013072300-msys-bin.meta
mingw-get.exe: *** ERROR *** package msys-base-2013072300-msys-bin.meta is already installed
```

构建libdl.a

下载好 [dlfcn-win32 for MinGW](#) ,并在该目录下创建 config.mak 文件,内容为

```
PREFIX=D:\software\mingw
BUILD_STATIC=yes
CC=gcc
AR=ar
RANLIB=ranlib
```

其中 PREFIX 为你的MinGW安装根目录。

现在将整个克隆下来的文件夹，复制到路径：

```
D:\software\mingw\msys\1.0\home\你的用户名
```

注意，上面的 D:\software\mingw 是MinGW的安装路径， msys 则是上一步中安装的msys，如果没有则往回看。下面同理。

回到 D:\software\mingw\msys\1.0 路径下，双击运行 msys.bat ，打开类unix终端，依次运行：

```
cd dlfcn-win32-1.4.1
make clean
make libdl.a
```

```
Because@Laptop ~
$ cd dlfcn-win32-1.4.1

Because@Laptop ~/dlfcn-win32-1.4.1
$ pwd
/home/Because/dlfcn-win32-1.4.1
```

```
Because@Laptop ~/dlfcn-win32-1.4.1
$ make clean
rm -f \
    src/dlfcn.o \
    libdl.dll libdl.a libdl.def libdl.dll.a libdl.lib libdl.exp \
    tmptest.c tmptest.dll \
    test-dladdr.exe test-dladdr-static.exe \
    test.exe test-static.exe testdll.dll testdll2.dll testdll3.dll
```

```
Because@Laptop ~/dlfcn-win32-1.4.1
$ make libdl.a
gcc -Wall -O3 -fomit-frame-pointer -Isrc -o src/dlfcn.o -c src/dlfcn.c
ar cru libdl.a src/dlfcn.o
ranlib libdl.a
```

然后你可以在

D:\software\mingw\msys\1.0\home\你的用户名\dlfcn-win32-1.4.1 路径下找到刚才编译出来的libdl.a,

将 libdl.a 复制到 D:\software\mingw\lib 目录。

mingw > msys > 1.0 > home > Because > dlfcn-win32-1.4.1 > 在 dlfcn-win3			
名称	修改日期	类型	大小
.github	2025/1/11 12:56	文件夹	
cmake	2025/1/11 12:56	文件夹	
cmake-test	2025/1/11 11:53	文件夹	
src	2025/1/11 13:05	文件夹	
tests	2025/1/11 12:56	文件夹	
tools	2025/1/11 12:56	文件夹	
.appveyor.yml	2023/7/4 14:57	YML 文件	10 KB
.gitignore	2023/7/4 14:57	GITIGNORE 文件	4 KB
.travis.yml	2023/7/4 14:57	YML 文件	2 KB
CMakeLists.txt	2023/7/4 14:57	文本文档	1 KB
config.mak	2025/1/11 12:41	MAK 文件	1 KB
configure	2023/7/4 14:57	文件	6 KB
COPYING	2023/7/4 14:57	文件	2 KB
dlfcn-win32-config.cmake.in	2023/7/4 14:57	IN 文件	1 KB
libdl.a	2025/1/11 13:05	A 文件	8 KB
Makefile	2023/7/4 14:57	文件	3 KB
README.md	2023/7/4 14:57	MD 文件	4 KB

安装适用于windows的C语言核心

(实际上只起到一个中转站的作用，编译器还是用的gcc)
打开普通的终端，输入：

```
pip install jupyter_c_kernel
```

将定位到 Python311\Lib\site-packages\jupyter_c_kernel\ 文件夹 下的kernel.py里面所有的 -rdynamic 全部换成 -export-dynamic 。

将定位到 Python311\Lib\site-packages\jupyter_c_kernel\ 文件夹 下的 install_c_kernel 修改为 install_c_kernel.py，并运行命令

```
python install_c_kernel.py
```

名称	修改日期	类型	大小
__pycache__	2025/1/11 12:06	文件夹	
resources	2025/1/11 10:59	文件夹	
 __init__.py	2025/1/11 10:59	JetBrains PyCharm ...	0 KB
 __main__.py	2025/1/11 10:59	JetBrains PyCharm ...	1 KB
 install_c_kernel.py	2025/1/11 10:59	JetBrains PyCharm ...	2 KB
 kernel.py	2025/1/11 12:05	JetBrains PyCharm ...	7 KB

最后

```
jupyter lab
```

即可。

三、参考文献指路

<https://zhuanlan.zhihu.com/p/648326046>

因为知乎的这篇文章写于23年8月份，笔者在25年1月份按图索骥的时候发现其中有一些不甚详细的内容（比如make命令是怎么来的），以及一些其他小的问题（比如 `dlfcn.c` 文件不需要改， `dlfcn.h` 文件不需要复制过去），因此重新整理了一份教程。