

How to invoke pyx code from C

井民全, Jing, mqjing@gmail.com

這文章 go through 了一次, 如何從 C 來呼叫 pyx module 的 code.

內容包含:

- 最簡單的 pyx code
- 如何自動轉換 pyx code 到 C 產生 object files
- 最簡單的呼叫 pyx module 範例 from C

Enjoy

Jing.

Pyx Code

```
cdef public void my_python_module():  
    print('1234')
```

Quick

```
# transfer and build  
cython my_python.pyx  
gcc my_python.c -o my_python.o -shared -pthread -fPIC -fwrapv -O2 -Wall -fno-strict-aliasing  
-I/usr/include/python3.5
```

```
# usage  
main.c
```

```
#include <Python.h>  
#include "my_python.h"  
  
int main() {  
    Py_Initialize();
```

```
PyInit_my_python(); // --- init the python module
my_python_module(); // <--- call my python module
Py_Finalize();
return 0;
}
```

```
gcc main.c -ldl my_python.o -I/usr/include/python3.5
-L/usr/lib/python3.5/config-3.5m-x86_64-linux-gnu -lpython3.5
```

Transfer the pyx code to C module

Step 1: Generate the C code

cython my_python.pyx

Or
python setup.py build_ext --inplace

Result

```
jing@jing-pc:~/test/cython/call_cython/test$ cython my_python.pyx
jing@jing-pc:~/test/cython/call_cython/test$ ls
main.c  my_python.c  my_python.h  my_python.pyx
jing@jing-pc:~/test/cython/call_cython/test$
```

Step 2: Build the object file

```
gcc my_python.c -o my_python.o -shared -pthread -fPIC -fwrapv -O2 -Wall -fno-strict-aliasing
-I/usr/include/python3.5
```

Result

```
jing@jing-pc:~/test/cython/call_cython/test$ gcc my_python.c -o my_python.o -shared -pthread -fPIC -fwrapv -O2 -Wall -fno-strict-aliasing -I/usr/include/python3.5
jing@jing-pc:~/test/cython/call_cython/test$ ls
main.c  my_python.c  my_python.h  my_python.o  my_python.pyx
jing@jing-pc:~/test/cython/call_cython/test$
```

Usage

main.c

```
#include <Python.h>
#include "my_python.h"

int main() {
    Py_Initialize();
    PyInit_my_python(); // <--- init the python module
    my_python_module(); // <--- call my python module
    Py_Finalize();
    return 0;
}
```

Build and Run

Step 1: build the source code

```
gcc main.c -ldl my_python.o -I/usr/include/python3.5
-L/usr/lib/python3.5/config-3.5m-x86_64-linux-gnu -lpthon3.5
```

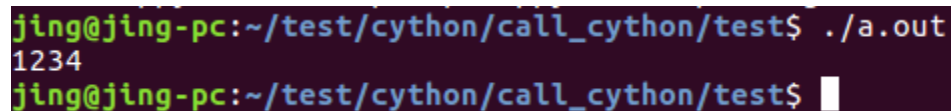
Step 2: Setup the shared library path, LD_LIBRARY_PATH

```
LD_LIBRARY_PATH=.
```

```
export LD_LIBRARY_PATH
```

Step 3: Run

```
./a.out
```



```
jing@jing-pc:~/test/cython/call_cython/test$ ./a.out
1234
jing@jing-pc:~/test/cython/call_cython/test$
```

