# C调用FORTRAN子程序

将FORTRAN子程序或子函数编译为静态链接库文件(lib)，被C语言调用。

编译Fortran的lib文件时，必须要打开：

Fortran-> External procedures -> Calling convention -> C, Reference

Fortran代码示例：

module testfor

use iso\_c\_binding

implicit none

contains

subroutine testingfor(x,string,l) bind(c, name="testingfor")

end subroutine

end module

C或C++调用lib时，

extern "C" {

void testingfor(double \* x); // C++

}

# FORTRAN调用C程序

我尝试了两种方式：

（1）将C语言程序vec(),编译为lib文件，然后再FORTRAN程序中，写：

interface

subroutine vec(r,len) bind(c)

use,intrinsic :: iso\_c\_binding

implicit none

integer(c\_int) :: len

real(c\_double) :: r(0:len)

end subroutine vec

end interface

这种方法最方便。注意： FORTRAN与C的传递参数给函数的不同方式。（传值或传地址）

（2）在C语言程序中，编写接口API：

#define add ADD // Fortran 调用使用，全部大写

#define arrayadd ARRAYADD

#ifdef \_\_cplusplus /\* wrapper to enable C++ usage \*/

extern "C" {

#endif

void add(float \*, float \*, float \*);

void arrayadd(int \*,float \*, float \*, float \*);

#ifdef \_\_cplusplus /\* wrapper to enable C++ usage \*/

}

#endif

当FORTRAN调用以上C语言的lib库时，如果没有定义：

#define add ADD // Fortran 调用使用，全部大写

需要在Intel FORTRAN编译器中设置：

FORTRAN -》 External Procedures -》Upper Case (/names:uppercase)

注意：32位和64位程序之间的调用！必须匹配~