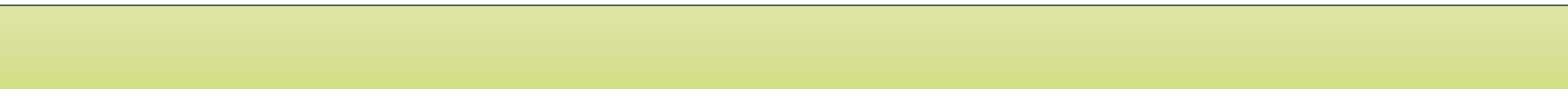


編譯器及函式庫

國立臺灣師範大學物理學系 陳俊明

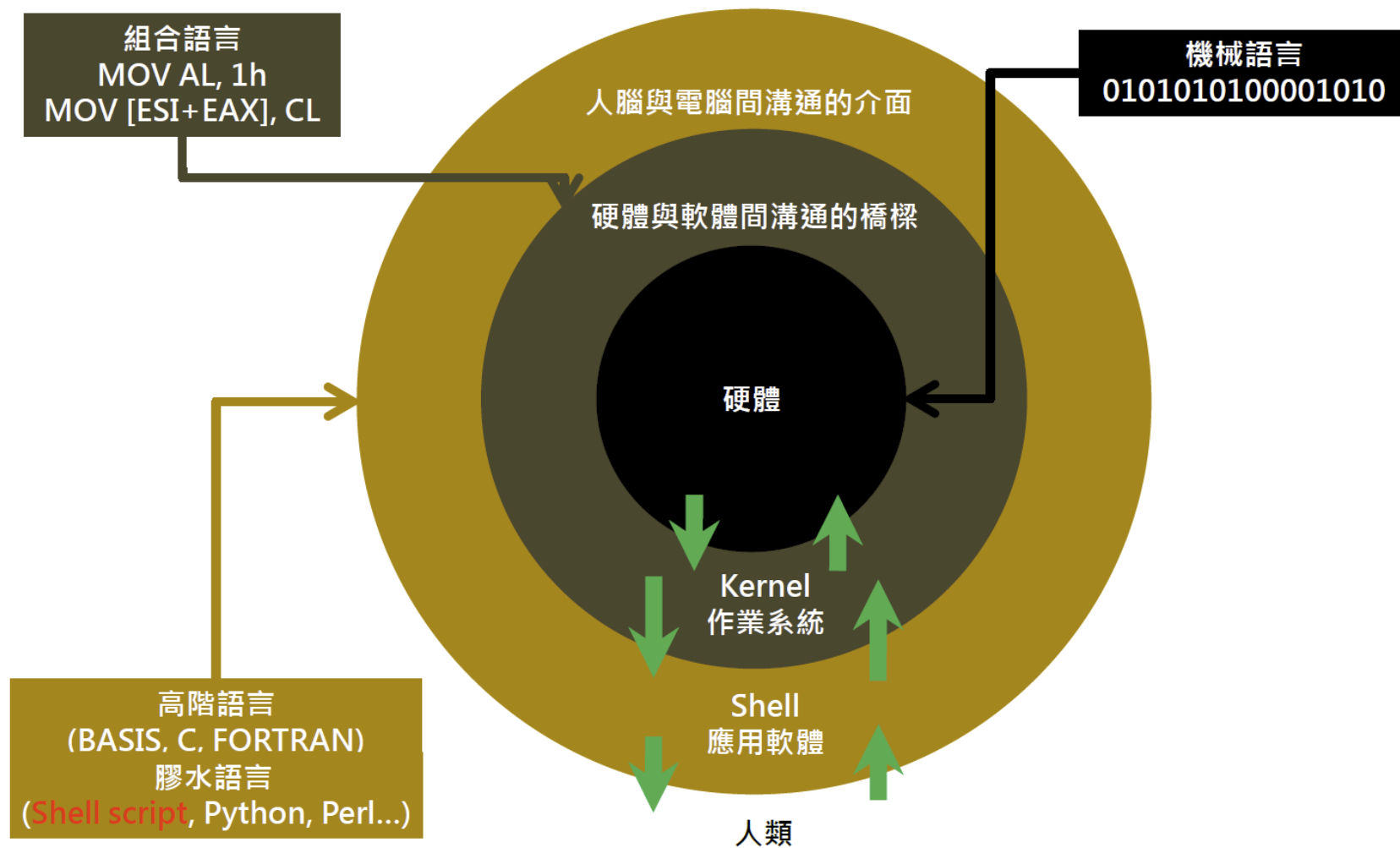
chunming@ntnu.edu.tw



摘要

- 什麼是編譯器？
- 安裝 GNU, Intel, Nvidia HPC SDK 編譯器
- 分別使用 GNU 跟 Intel 的編譯器安裝 OpenMPI
- 編譯並執行你自己的程式
- 編譯以下的函式庫：
 - FFTW3
 - BLAS
 - LAPACK
 - GSL
 - HDF5
- 使用MKL (Math Kernel Library) 或其它函式庫

什麼是編譯器？



常用的編譯器



- GNU Compiler
依循 GNU License 的編譯器

- Intel Compiler
由 Intel 開發的編譯器，對 Intel CPU 具最高支援度及效能最佳化



- PGI Compiler
由 PGI 開發的編譯器，支援 OpenACC



- NVIDIA HPC SDK™

以PGI Compiler為基礎並整合CUDA開發的編譯器及函式庫，使 HPC 開發人員能夠對從 GPU 基礎到 CPU 以及整個互連的整個 HPC 平台進行編程。

安裝 GNU 編譯器

- 安裝需要的套件

```
[root@master ~]# yum install gcc gcc-gfortran gcc-c++ libtool
```

- 手動編譯 GCC，前往 <https://gcc.gnu.org/releases.html> 下載編譯，順便安裝需要的套件

```
[root@master ~]# yum install wget bzip2
```

```
[root@master ~]# wget http://mirror.linux-ia64.org/gnu/gcc/releases/gcc-14.2.0/gcc-14.2.0.tar.gz
```

- 用 wget 指令下載 gcc-x.y.z.tar.gz，解開程式碼準備進行安裝

```
[root@master ~]# tar zxvf gcc-14.2.0.tar.gz
```

- 進入程式碼並下載 GCC 編譯必要的程式

```
[root@master gcc-14.2.0]# ./contrib/download_prerequisites
```

安裝 GNU 編譯器

- 建立安裝目的目錄

```
[root@master gcc-14.2.0]# mkdir -p /opt/gcc/14.2.0
```

- 組態要安裝的參數

```
[root@master gcc-14.2.0]# ./configure --prefix=/opt/gcc/14.2.0 --enable-checking=release --enable-languages=c,c++,fortran --disable-multilib
```

- 進行編譯，-j 後面的數字視核心數而有所不同

```
[root@master gcc-14.2.0]# make -j (CPU cores)
```

- 進行安裝

```
[root@master gcc-14.2.0]# make install
```

編譯時間大約會花 1 ~ 5 小時

使用 GNU 編譯器

- 使用系統預設的 GCC 不用額外設定，除非要指定其他版本的 GCC 才要設定

```
[root@master ~]# export PATH=/opt/gcc/14.2.0/bin:$PATH  
[root@master ~]# export LD_LIBRARY_PATH=/opt/gcc/14.2.0/lib64:$LD_LIBRARY_PATH
```

- 用 gcc , g++ , gfortran 指令編譯

```
[root@master ~]# gcc -o test test.c  
[root@master ~]# g++ -o test test.cpp  
[root@master ~]# gfortran -o test test.f90
```

使用 GNU 編譯器

- 安裝不同版本GNU 編譯器

利用SCL(Software Collection) 在系統上建構並同時安裝不同版本的軟體
使用SCL不會影響系統原來的版本。

```
[root@master ~] yum install gcc-toolset-11  
[root@master ~] scl enable gcc-toolset-11 bash  
[root@master ~] gcc --version
```


安裝 Intel 編譯器(Intel® oneAPI Base Toolkit)

- 下載 Intel® oneAPI Base Toolkit :

<https://www.intel.com/content/www/us/en/developer/tools/oneapi/base-toolkit.html>

```
[root@master ~]# wget https://registrationcenter-download.intel.com/akdlm/IRC_NAS/e6ff8e9c-ee28-47fb-abd7-5c524c983e1c/l_BaseKit_p_2024.2.1.100_offline.sh
```

- 進行安裝

```
[root@master ~]# chmod +x l_BaseKit_p_2024.2.1.100_offline.sh  
[root@master ~]# ./l_BaseKit_p_2024.2.1.100_offline.sh
```

標準安裝大約需要 25 GB 空間，參考 VM 調整大小：<https://sam.liho.tw/?p=6965>

安裝 Intel® oneAPI HPC Toolkit

- 下載 Intel® oneAPI Base Toolkit :

<https://www.intel.com/content/www/us/en/developer/tools/oneapi/hpc-toolkit.html>

```
[root@master ~]# wget https://registrationcenter-download.intel.com/akdlm/IRC_NAS/d461a695-6481-426f-a22f-b5644cd1fa8b/l_HPCKit_p_2024.2.1.79_offline.sh
```

- 進行安裝

```
[root@master ~]# chmod +x l_HPCKit_p_2024.2.1.79_offline.sh  
[root@master ~]# ./l_HPCKit_p_2024.2.1.79_offline.sh
```

使用 Intel 編譯器

- 用 `source` 指令載入環境

```
[root@master ~]# source /opt/intel/oneapi/setvars.sh intel64
```

- 用 `icx` , `icpx`, `ifort/ifx` 指令編譯

```
[root@master ~]# icx -o test test.c  
[root@master ~]# icpx -o test test.cpp  
[root@master ~]# ifx -o test test.f90
```

PGI 編譯器

- 下載PGI編譯器：https://www.pgroup.com/support/release_archive.php
- 目前已不提供免費下載，改為NVIDIA HPC SDK

安裝 NVIDIA HPC SDK

- NVIDIA HPC SDK: <https://developer.nvidia.com/hpc-sdk>
- 下載 「Linux x86_64 Tarball」
- 解壓縮後開始安裝

```
[root@master ~]# wget https://developer.download.nvidia.com/hpc-sdk/24.7/nvhpc_2024_247_Linux_x86_64_cuda_12.5.tar.gz  
[root@master ~]# tar xpf nvhpc_2024_247_Linux_x86_64_cuda_12.5.tar.gz  
[root@master ~]# nvhpc_2024_247_Linux_x86_64_cuda_12.5/install
```

安裝 NVIDIA HPC SDK

```
[root@master nvhpc_2023_237_Linux_x86_64_cuda_12.2]# ./install
```

Welcome to the NVIDIA HPC SDK Linux installer!

You are installing NVIDIA HPC SDK 2023 version 23.7 for Linux_x86_64.
Please note that all Trademarks and Marks are the properties
of their respective owners.

Press enter to continue...

A network installation will save disk space by having only one copy of the
compilers and most of the libraries for all compilers on the network, and
the main installation needs to be done once for all systems on the network.

- 1 Single system install
- 2 Network install

Please choose install option:

1|

安裝 NVIDIA HPC SDK

```
Please specify the directory path under which the software will be installed.  
The default directory is /opt/nvidia/hpc_sdk, but you may install anywhere you wish,  
assuming you have permission to do so.  
Note: directory /opt/nvidia/hpc_sdk was created.  
Installation directory? [/opt/nvidia/hpc_sdk]  
Installing NVIDIA HPC SDK version 23.7 into /opt/nvidia/hpc_sdk
```

- 用 `nvcc` , `nvc++`, `nvfortran` 指令編譯

```
[root@master ~]# module load /opt/nvidia/hpc_sdk/modulefiles/nvhpc/22.11  
[root@master ~]# nvcc -o test test.c  
[root@master ~]# nvc++ -o test test.cpp  
[root@master ~]# nvfortran -o test test.f90
```

Compiler 對照表

	GNU	PGI	Intel	Nvidia HPC SDK	OpenMPI	IntelMPI	
						Intel	Other
C	gcc	pgcc	icc/icx	nvc	mpicc	mpiicc	mpicc/ mpigcc
C++	g++	pgc++	icpc/icpx	nvc++	mpiCC/mpic++/mpicxx	mpiicpc	mpicxx/mpigxx
Fortran	gfortran	pgfortran	ifort/ifx	nvfortran	mpifort/mpif77/mpif90	mpiifort	mpif77/mpif90/mpifc

安裝 OpenMPI - Source Code

- 下載 OpenMPI : <https://www.open-mpi.org/>
- 解開安裝檔

```
[root@master ~]# wget https://download.open-mpi.org/release/open-mpi/v4.1/openmpi-4.1.6.tar.gz  
[root@master ~]# tar zxvf openmpi-4.1.6.tar.gz
```

- 進入解壓縮後的目錄

```
[root@master ~]# cd openmpi-4.1.6
```

- 設定安裝組態

```
[root@master ~]# ./configure --prefix=/path/to/install
```

- 編譯 (X 為該台電腦核心數)

```
[root@master ~]# make -j X
```

- 安裝

```
[root@master ~]# make install
```

安裝 OpenMPI - Source Code

- GNU 組態

```
[root@master ~]# ./configure --prefix=/opt/openmpi/4.1.6_gcc_8.5 --enable-mpi-f77 --enable-mpi-f90 --enable-mpi-cxx CC=gcc CXX=g++ FC=gfortran F77=gfortran
```

- Intel 組態

```
[root@master ~]# source /opt/intel/oneapi/setvars.sh intel64  
[root@master ~]# ./configure --prefix=/opt/openmpi/4.1.5_intel_2024.2 --enable-mpi-f77 --enable-mpi-f90 -  
-enable-mpi-cxx CC=icx CXX=icpx FC=ifx F77=ifx
```

help選項可以讓你了解更多參數, 如"./configure -h"

編譯函式庫

- 下載編譯函式庫

- BLAS : <http://www.netlib.org/blas/>
- LAPACK : <http://www.netlib.org/lapack/>
- FFTW : <http://www.fftw.org/>
- GSL : <https://www.gnu.org/software/gsl/>
- HDF5 : <https://www.hdfgroup.org/solutions/hdf5/>

- 解壓縮函式庫原始碼檔

```
tar zxvf library_source.x.y.z.tar.gz
```

- 進入解壓縮後的目錄

```
cd library_source.x.y.z
```

- 先讀 **README / INSTALL** 檔案，設定組態 (`./configure --prefix=/path/to/install`) 或是修改 `Makefile / Make.in / Make.XXXX`

編譯函式庫

- 編譯

```
[root@master ~]# make
```

- 安裝：有些程式沒有 **install** 選項，直接複製 *.a 檔案到你到的路徑

```
[root@master ~]# make install 或 # cp *.a /path/to/install
```

- 設定動態及靜態函式庫的環境變數、設定 header files 引入的環境變數

```
[root@master ~]# export LD_LIBRARY_PATH=/path/to/install/lib:$LD_LIBRARY_PATH
```

```
[root@master ~]# export LIBRARY_PATH=/path/to/install/lib:$LIBRARY_PATH
```

```
[root@master ~]# export INCLUDE=/path/to/install/include:$INCLUDE
```

```
[root@master ~]# export C_INCLUDE_PATH=/path/to/install/include:$C_INCLUDE_PATH
```

安裝系統函式庫

記得將計算節點一併設定

- 建議安裝 EPEL

```
[root@master ~]# yum config-manager --set-enabled powertools  
[root@master ~]# yum install epel-release
```

- 安裝好 EPEL 才能直接安裝相關系統套件

```
[root@master ~]# yum install blas lapack fftw hdf5 gsl blas-devel lapack-devel fftw-devel hdf5-devel gsl-devel  
glibc-static
```

- 檢查系統安裝的套件

```
[root@master ~]# rpm -q blas lapack fftw hdf5  
blas-3.8.0-8.el8.x86_64  
lapack-3.8.0-8.el8.x86_64  
fftw-3.3.5-11.el8.x86_64  
hdf5-1.10.5-4.el8.x86_64
```

使用 MKL 或其他函式庫

連結種類	參數	範例
引入 header 檔	-I	-I/opt/openmpi/4.1.5-intel-2024.2/include
動態連結	-L	-L/opt/intel/oneapi/mkl/2024.2.0/lib/intel64 -lmkl_core
靜態連結	libXXX.a	/opt/intel/oneapi/mkl/2024.2.0/lib/intel64/libmkl_core.a

- 動態連結

```
[root@master ~]# ifort -o test test.f90 -I/opt/intel/oneapi/mkl/2024.2.0/include -L/opt/intel/oneapi/mkl/2024.2.0/lib/intel64 -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm -ldl
```

- 靜態連結

```
[root@master ~]# MKLROOT=/opt/intel/oneapi/mkl/2024.2.0/lib/intel64  
[root@master ~]# ifort -o test.x test.f90 -I/opt/intel/oneapi/mkl/2024.1.0/include -Wl,--start-group ${MKLROOT}/libmkl_intel_lp64.a ${MKLROOT}/libmkl_sequential.a ${MKLROOT}/libmkl_core.a-Wl,--end-group -lpthread -lm -ldl
```

Intel MKL Link Line Advisor

<https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl-link-line-advisor.html>

Intel® oneAPI Math Kernel Library Link Line Advisor

Introduction

The Intel® oneAPI Math Kernel Library (oneMKL) is designed to run on multiple processors and operating systems. It is also compatible with several compilers and third-party libraries, and provides different interfaces to the functionality. To support these different environments, tools, and interfaces, oneMKL provides multiple libraries to choose from.

To see which libraries are recommended for a particular use case, specify the parameters in the drop-down lists below.

Intel® oneAPI Math Kernel Library (oneMKL) Link Line Advisor v6.19

Reset

Select Intel® product:	oneMKL 2022
Select OS:	Linux*
Select programming language:	Fortran
Select compiler:	Intel® Fortran Classic
Select architecture:	Intel® 64
Select dynamic or static linking:	Static
Select interface layer:	Fortran API with 32-bit integer
Select threading layer:	Sequential
Select OpenMP library:	<Select OpenMP>
Enable OpenMP offload feature to GPU:	☐
Select cluster library:	☐ Parallel Direct Sparse Solver for Clusters (BLACS required) ☐ Cluster Discrete Fast Fourier Transform (BLACS required) ☐ ScaLAPACK (BLACS required) ☐ BLACS
Select MPI library:	<Select MPI>
Select the Fortran 95 interfaces:	☐ BLAS95 ☐ LAPACK95
Link with Intel® oneMKL libraries explicitly:	☒
Link with DPC++ debug runtime compatible libraries:	☐

Use this link line:
-Wl,--start-group \${MKLROOT}/lib/intel64/libmkl_intel_lp64.a \${MKLROOT}/lib/intel64/libmkl_sequential.a \${MKLROOT}/lib/intel64/libmkl_core.a -Wl,--end-group -lpthread -lm -ldl

常用最佳化參數

- GNU

```
[root@master ~]# gcc -O2 -ffast-math -ftree-vectorize -funroll-loops
```

- Intel

```
[root@master ~]# icc -O3 -axCORE-AVX2 ##### intel 11 or later  
[root@master ~]# icc -O3 -axCORE-AVX512 ##### intel 15 or later
```

CPU指令集視CPU規格而定，查看 `/proc/cpuinfo` 檔案

參考：[Optimize Options \(Using the GNU Compiler Collection \(GCC\)\)](#), [Intel® 64 and IA-32 Architectures Optimization Reference Manual](#),

回家作業

- 練習自行安裝 FFTW3 在自己的 VM (可用 GNU)
- 編譯 fftw_example.c 程式
 - https://github.com/undees/fftw-example/blob/master/fftw_example.c
 - 參考：
gcc -o test.exe fftw_example.c -I/opt/gnu-math/fftw-3.3.10/include -L/opt/gnu-math/fftw-3.3.10/lib -lfftw3 -lm
 - 可以嘗試拿掉 -lm 試試看？