

- ★ gfortran のインストールは済んでいるものとして、話を進めます。
- ★ os は windows 8.1 pro
- ★ プログラムは Fortran 90 の形式で作成。
- ★ 並列計算用のプログラムは、別途、すでに作成、用意してあることを想定。
- ★ インストールなどは管理者権限が必要と思いますが(未確認、標準アカウントでも可能かもしれない)、その他は、標準アカウントでも可能なようです。

++++
1) まず、Microsoft MPI をダウンロード、インストールする。

本項の内容は、下記ページの情報を参考にしました。ありがとうございました。

<https://qiita.com/MusicScience37/items/a483a5d5c1d1b94eedf7>

ダウンロードページ

<https://www.microsoft.com/en-us/download/details.aspx?id=56727>

msmpisetup.exe と msmpisdsk.msi の両方をダウンロード。

ダブルクリックで実行しインストール。

どちらが先というのは、よく分からないが、msmpisetup.exe からで良いかもしれない。

インストールが完了後、コマンドプロンプトで、「set MSMPI」を入力、実行すると、下記のような表示がされるはず。

```
-----  
MSMPI_BIN=C:\Program Files\Microsoft MPI\Bin\  
MSMPI_INC=C:\Program Files (x86)\Microsoft SDKs\MPI\Include\  
MSMPI_LIB32=C:\Program Files (x86)\Microsoft SDKs\MPI\Lib\x86\  
MSMPI_LIB64=C:\Program Files (x86)\Microsoft SDKs\MPI\Lib\x64\  
-----
```

なお、自分が使用したコンピュータでは、「Microsoft HPC Pack 2008 R2」というものがあらかじめインストールされており、こちらも使えたのだけれど、

試行錯誤の段階で、Microsoft MPI を使ってみることにしたので、Microsoft HPC Pack 2008 R2 はアンインストールした。

(アンインストールしなければ、エラーメッセージが出て、msmpisetup.exe のインストールが出来なかったため)

このとき、いちど、下記のような状況になった。

- ① msmpisetup.exe のインストール、失敗
- ② msmpisdsk.msi のインストール、完了
- ③ Microsoft HPC Pack 2008 R2 のアンインストール

すると、上記、set MSMPI の入力で表示される内容が、本来、表示されるはずの内容より、欠けた内容となっていたので、msmpisetup.exe のインストールを再度実行した。そのときは、インストールは無事済み、その後の set MSMPI で表示される内容も、問題なくなった。

++++

+++++

2) 下準備 その1 ライブラリファイルの下準備

本項の内容は、下記ページの情報を参考にしました。ありがとうございました。

<https://qiita.com/michioga/items/fae993892a808d488648>

上記ページで紹介されている方法と、ほぼ同様に行う。

とりあえず、まず、コマンドプロンプトで、
プログラム実行フォルダに移動しておく。

例えば、プログラム実行フォルダが、C:\Users\gakusei\keisan であるなら、
コマンドプロンプトで、cd C:\Users\gakusei\keisan を入力し、実行。

後は、下記を順次、入力、実行。

① gendef C:\Windows\System32\msmpi.dll

② dlltool -d msmpi.def -l libmsmpi.a -D C:\Windows\System32\msmpi.dll

msmpi.dll というファイルが、C:\Windows\System32 フォルダの中にあるかどうか。
無ければ、この資料で紹介する方法は、使えないと思いますので、
別の方法をとることになるかと思います。

また、gendef, dlltool というコマンドが有効かどうか。

gfortran のインストールや、パスを通すことができているならば、可能と思いますが、
この資料ではそのあたりの説明は省きます。

なお、上記 ② のコマンド実行後、約 7～8 秒間ほどでしょうか、
プログラム実行フォルダ内に、拡張子が o などのファイルが、
非常にたくさん生成されますが、実行が終わると、
不要なファイルは自動削除され、必要なファイルのみ残りますので、そのままお待ちください。

もし、実行が終わらず、ファイルの生成が止まらないようでしたら、
コマンドプロンプト上で、ctrl キーを押しながら c キーを押し、
実行を強制終了、その後、拡張子が o 等のファイルを、手動で削除するという
ことが、必要になるでしょう。

+++++

++++
3) 下準備 その2 mpi.f90 および mpifptr.h ファイルの下準備

本項の内容は、下記ページの情報を参考にしました。ありがとうございました。

<https://github.com/dev-cafe/autocmake/issues/85>

<http://penguinitis.g1.xrea.com/study/FrontFlowRed/FFR-gfortran-mingw64.html>

前述の通り、Microsoft MPI インストールが済むと、
フォルダ C:\Program Files (x86)\Microsoft SDKs\MPI\Include 内に、
「 mpi.f90 」というファイルが置かれているはずである。

また、フォルダ C:\Program Files (x86)\Microsoft SDKs\MPI\Include\x64 内に、
「 mpifptr.h 」というファイルが置かれているはずである。

上記2つのファイルを、プログラム実行フォルダにコピーして置いておく。

次に、mpi.f90 ファイル内で、「 PARAMETER (MPI_ADDRESS_KIND=INT_PTR_KIND()) 」と
記載されている行があるので、この行を「 PARAMETER (MPI_ADDRESS_KIND=8) 」と
書き換える。(INT_PTR_KIND() の部分を、8 に書き換える)

私は、書き換えたファイルを、もともとのファイルとは区別して扱いたいので、
ファイル名を「 edit_mpi.f90 」と付けて、別ファイルとして保存する。

次に、コマンドプロンプトに戻る。

前述の状況を引き継ぎ、プログラム実行フォルダで、
コマンドプロンプトへの入力、実行の作業を進める。

下記コマンドを入力、実行し、モジュールファイルを作成する。

```
gfortran -c -fno-range-check edit_mpi.f90
```

コマンド実行後、mpi_constants.mod, mpi_base.mod, mpi_sizeofs.mod, mpi.mod という、
4つのモジュールファイルが作成される。
このうち、mpi.mod から、他の3つのモジュールを呼び出して使うことができるようである。

++++

+++++
4) 下準備 その3 並列計算を行う fortran プログラム内への必要事項の追記

本項の内容は、下記ページの情報を参考にしました。ありがとうございました。

<https://github.com/dev-cafe/autocmake/issues/85>

別途、あらかじめ作成してあるはずの、並列計算用 fortran プログラム内で、「use mpi」の文を記載しておく。

私は、上記ページの情報を参考にし、下記のように記載しました。

use mpi

implicit none

(以降、プログラム内で使用する変数の型宣言など、プログラムの内容が続く)

+++++
5) 並列計算用プログラムのコンパイルとリンク、実行ファイルの実行

本項の内容は、下記ページの情報を参考にしました。ありがとうございました。

<https://github.com/dev-cafe/autocmake/issues/85>

前述の状況を引き継ぎ、プログラム実行フォルダで、コマンドプロンプトへの入力、実行の作業を進める。

下記コマンドを入力、実行し、並列計算用プログラムのコンパイルとリンクを行う。

(並列計算用プログラムの名前は heiretu.f90 としてある)

gfortran heiretu.f90 -o heiretu libmsmpi.a

上記コマンド実行後、プログラム実行フォルダ内に、heiretu.exe という、実行ファイルが作成されるはずである。

これを用い、例えば、4並列の並列計算を実行したい場合は、下記コマンドのように入力、実行する。

mpiexec -n 4 heiretu.exe

上記コマンド、mpiexec に対応する実行ファイル mpiexec.exe は、フォルダ C:\Program Files\Microsoft MPI\Bin 内にある。

このフォルダへパスを通す作業は、Microsoft MPI インストール時に自動で行われるようであり、絶対パスを書いてコマンドを実行する必要は無いようである。

(もし、絶対パス記載で、コマンド実行する場合は、下記の通り
"C:\Program Files\Microsoft MPI\Bin\mpiexec.exe" -n 4 heiretu.exe)

++++
付録) バッチファイルを利用したプログラムの実行

並列計算用プログラムのテストや改良を行うとき、
毎回、コマンドプロンプトを起動し、プログラム実行フォルダに移動し...、という手順は、
若干めんどろなので、バッチファイルによる実行を利用する。

ただし、上記3)、4)の手順のうち、
下記の項目は、プログラム実行フォルダにおいて、
あらかじめ、ユーザーが手動で済ませておく。

- ★ mpi.f90 および mpifptr.h のプログラム実行フォルダへのコピー、
- ★ mpi.f90 ファイル内の記載「 PARAMETER (MPI_ADDRESS_KIND=INT_PTR_KIND()) 」の変更、
- ★ 自分で用意した 並列計算用 fortran プログラムへの use mpi 文の追記、

上記の項目は、一度、実施しておけば、以降、繰り返し実施する必要が無いので、
あらかじめ済ませておく。

以下に、プログラムの繰り返し実行に便利と思われるバッチファイルの内容を記載する。

以下の内容を、テキストファイルにコピーして保存し、
ファイルの拡張子を、.txt から .bat に変更する。

この拡張子 .bat に変更したファイルをマウスでダブルクリックするなどすれば、
記載の内容がコマンドプロンプト上で実行される。

なお、この拡張子 .bat のファイルは、プログラム実行フォルダに置いておく。

このとき、あらかじめコマンドプロンプトを起動しておく必要は無い。

```
-----  
@echo off
```

```
echo.  
echo.
```

```
echo ctrl キー 押しながら c キー 押すと, 強制終了できます.  
echo 文字や数値など, なんらかの入力を求められる部分で enter キー を押しても 強制終了します.
```

```
:STEP_1
```

```
rem -----
```

```
echo.  
echo.
```

```
set /p BUNKI_1="ライブラリの準備が必要? 必要なら y, 不要なら n を入力 -> "
```

```
if %BUNKI_1% == y ( goto LIBRA
```

```
    ) else ( goto STEP_2
```

```
    )
```

```
:LIBRA
```

```
echo.  
echo.
```

```
gendif C:¥Windows¥System32¥msmpi.dll
```

```
echo.  
echo.
```

```
dlltool -d msmpi.def -l libmsmpi.a -D C:¥Windows¥System32¥msmpi.dll
```

```
rem -----
```

:STEP_2

rem -----

echo.
echo.

set /p BUNKI_2="モジュールの準備が必要？ 必要なら y, 不要なら n を入力 -> "

if %BUNKI_2% == y (goto MODU

) else (goto STEP_3

)

:MODU

echo.
echo.

set /p FILE_NAME_1="モジュールファイル作成のためのファイル名を入力(拡張子は除く) -> "

echo.
echo.

echo on

gfortran -c -fno-range-check %FILE_NAME_1%.f90

@echo off

rem -----

:STEP_3

rem -----

echo.
echo.

set /p FILE_NAME_2="並列計算プログラムのファイル名を入力(拡張子は除く) -> "

echo.
echo.

echo on

gfortran %FILE_NAME_2%.f90 -o %FILE_NAME_2% libmsmpi.a

@echo off

rem -----

:STEP_4

rem -----

echo.
echo.

set /p HEIRETU_NUM="並列数を入力(mpiexec -n x の x に対応する値を入力) -> "

rem -----

:STEP_5

rem -----

echo.
echo.

echo on

mpiexec -n %HEIRETU_NUM% %FILE_NAME_2%.exe

@echo off

rem -----

:STEP_6

rem -----

echo.
echo.

set /p BUNKI_3="計算の再実行 k, 並列数やり直し h, ファイル名入力・コンパイルやり直し f, やめる n 入力 ->
"

if %BUNKI_3% == k (goto STEP_5

) else if %BUNKI_3% == h (goto STEP_4

) else if %BUNKI_3% == f (goto STEP_3

) else (goto SAIGO

)

rem -----

:SAIGO

echo.
echo.

pause

+++++

+++++

参考にさせて頂いたページのリスト)

<https://qiita.com/MusicScience37/items/a483a5d5c1d1b94eedf7>

<https://qiita.com/michioga/items/fae993892a808d488648>

<https://github.com/dev-cafe/autocmake/issues/85>

<http://penguinitis.g1.xrea.com/study/FrontFlowRed/FFR-gfortran-mingw64.html>

その他 参考にさせて頂いたページのリスト)

<https://amanotk.github.io/fortran-resume-public/chap10.html>

+++++

★ インクルードファイル mpif.h は使えないのか？

使える。ただ、mpi.f90 ファイルと同様に、

ファイル内の「PARAMETER (MPI_ADDRESS_KIND=INT_PTR_KIND())」の部分の記載を、
「PARAMETER (MPI_ADDRESS_KIND=8)」などと書き換える必要がある。

そうでないと、「Error: Function 'int_ptr_kind' at (1) has no IMPLICIT type」のエラーメッセージが出る。

なお、intel fortran の場合は、同じ mpif.h ファイルに何も手を加えなくても、
エラーメッセージが出ることは無い。有償であるぶん、何か配慮、工夫があるのだと思う。

Microsoft MPI を利用する場合、mpif.h は、
フォルダ C:\Program Files (x86)\Microsoft SDKs\MPI\Include に入っているが、

上述の通り、ファイル内を編集するのであれば、
もう、プログラム実行フォルダにコピーしておいて良いのではないかと考える。
(もととのファイルは、無編集で残しておいたほうが、良いのではないかと考えるから)

同様に、mpif.h ファイル内でインクルードされるファイル mpifptr.h も、
こちらはファイル内を編集するわけでは無いが、
ついでにプログラム実行フォルダにコピーしておいて良いのではないかと考える。
(mpifptr.h は、フォルダ C:\Program Files (x86)\Microsoft SDKs\MPI\Include\x64 などに入っている)

mpif.h ファイルを利用する場合は、前述したような、モジュールファイルを用意する手間が無くなる。
また、モジュールファイルが作成されないので、プログラム実行フォルダ内が、少しスッキリする。

まとめると、モジュールファイルの作成ではなく、mpif.h ファイルを利用する場合は、次のような手順が考えられる。

1) プログラム実行フォルダに、mpif.h と、mpifptr.h をコピーして置いておく。

2) 区別するため、mpif.h の名前を edit_mpif.h などと変え、ファイル内の、

PARAMETER (MPI_ADDRESS_KIND=INT_PTR_KIND()) の記載を、

PARAMETER (MPI_ADDRESS_KIND=8) と書き換える。

3) ライブラリファイルを用意する手順は、mpi.f90 ファイルを利用する場合と同じように行う。

① gendef C:\Windows\System32\msmpi.dll

② dlltool -d msmpi.def -l libmsmpi.a -D C:\Windows\System32\msmpi.dll

4) モジュールファイル作成の手間は無くなる。

5) 並列計算用プログラムには、下記のように、'include edit_mpif.h' などの文を追記しておく。

implicit none

include 'edit_mpif.h'

(以降、プログラム内で使用する変数の型宣言など、プログラムの内容が続く)

6) 下記コマンドで、インクルードファイル(プログラム実行フォルダに置いた edit_mpif.h と mpifptr.h)の
インクルードと、ライブラリファイル libmsmpi.a のリンクを同時に行う(プログラム名は heiretu.f90 としてある)。

gfortran -fno-range-check heiretu.f90 -o heiretu libmsmpi.a

オプション -fno-range-check は必要。無いとエラーメッセージが出て、実行ファイルが作成されない。

上記コマンド実行後、プログラム実行フォルダに、実行ファイル heiretu.exe が作成される。

(終わり)