

module による環境設定 (コンパイラ、アプリ)

自然科学研究機構
岡崎共通研究施設
計算科学研究センター(RCCS)

更新履歷

- 2019/9/5 初稿作成
- 2020/1/16 情報更新

イントロダクション

この資料ではRCCSで `module` コマンドを使って環境設定する方法について説明します。

アプリケーション実行の際には実行時に必要なライブラリや実行ファイルの場所をあらかじめ指定する必要があります。また、アプリケーションをビルドする際にもコンパイラを指定したりする必要が発生します。

この資料では、 **Environment Modules** を使って上記の環境設定を行う方法を説明します。

(なお、`module` を使わず、直接設定ファイルを読み込むことも可能です。)

目次

- デフォルト環境
- 利用可能なモジュール
- 環境の読み込み、切り替え
- ライブラリやユーティリティの利用
- ヒント

デフォルト環境

- ログイン直後でもいくつかの module が読み込まれた状態になっています。(独自設定を追加していない場合)
- 読み込んでいる module リストは **module list** コマンドで確認できます。
- 環境変数(PATH等)も適宜設定されています。

```
[user@ccfep4 ~]$ module list
Currently Loaded Modulefiles:
  1) intel_parallelstudio/2018update4
  2) pgilic
  3) pgi/18.1
  4) allinea/7.1
  5) cuda/9.1
```

デフォルトでは以下が設定されています

- インテルコンパイラ(2018u4)
- PGIコンパイラ
- CUDA 9.1環境

```
[user@ccfep4 ~]$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Target: x86_64-redhat-linux
Configured with: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --enable-shared --enable-threads=posix --enable-checking=release --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-languages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initfini-array --disable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86_64 --build=x86_64-redhat-linux
Thread model: posix
gcc version 4.8.5 20150623 (Red Hat 4.8.5-36) (GCC)
[user@ccfep4 ~]$ icc -v
icc version 18.0.5 (gcc version 4.8.5 compatibility)
[user@ccfep4 ~]$
```

(module help (パッケージ名) で詳細な情報も確認できます。)

利用可能なモジュール

- **module avail** コマンドで利用可能モジュール名一覧が得られます。
- https://ccportal.ims.ac.jp/installed_applications もご参照下さい。

```
[user@ccfep5]$ module avail

----- /local/ap1/1x/modules/suite -----
intel_parallelstudio/2015update1      intel_parallelstudio/2018update2      intel_parallelstudio/2019updates      scl/devtoolset-6
intel_parallelstudio/2017update4      intel_parallelstudio/2018update4(default) scl/devtoolset-3                      scl/devtoolset-7
intel_parallelstudio/2017update8      intel_parallelstudio/2019update1      scl/devtoolset-4                      scl/devtoolset-8

----- /local/ap1/1x/modules/comp -----
cuda/10.1                               intel/15.0.1                          intel/17.0.8                          intel/18.0.5(default) intel/19.0.5      pgi/16.5      pgi/18.1(default)
cuda/7.5                               cuda/9.1(default)                    intel/17.0.4                          intel/18.0.2                          intel/19.0.1      julia/1.3.1   pgi/17.5

----- /local/ap1/1x/modules/apl -----
mpi/intelmpi/2017.3.196                mpi/intelmpi/2019.1.144                mpi/openmpi/2.1.3/gnu5.3              mpi/openmpi/2.1.3/intel17              mpi/openmpi/3.1.0/gnu5.3              mpi/openmpi/3.1.0/intel17              mpi/openmpi/4.0.0/gnu5.3              mpi/openmpi/4.0.0/intel17
mpi/intelmpi/2017.4.262                mpi/intelmpi/2019.5.281                mpi/openmpi/2.1.3/gnu6.3              mpi/openmpi/2.1.3/intel18              mpi/openmpi/3.1.0/gnu6.3              mpi/openmpi/3.1.0/intel18              mpi/openmpi/4.0.0/gnu6.3              mpi/openmpi/4.0.0/intel18
mpi/intelmpi/2018.2.199                mpi/intelmpi/5.0.2.044                mpi/openmpi/2.1.3/gnu7.3              mpi/openmpi/2.1.3/intel19              mpi/openmpi/3.1.0/gnu7.3              mpi/openmpi/3.1.0/intel19              mpi/openmpi/4.0.0/gnu7.3              mpi/openmpi/4.0.0/intel19
mpi/intelmpi/2018.4.274                mpi/openmpi/2.1.3/gnu4.8              mpi/openmpi/2.1.3/gnu8.3              mpi/openmpi/3.1.0/gnu4.8              mpi/openmpi/3.1.0/gnu8.3              mpi/openmpi/3.1.0/gnu8.3              mpi/openmpi/4.0.0/gnu4.8              mpi/openmpi/4.0.0/gnu8.3
mpi/intelmpi/2019                      mpi/openmpi/2.1.3/gnu4.9              mpi/openmpi/2.1.3/intel15              mpi/openmpi/3.1.0/gnu4.9              mpi/openmpi/3.1.0/intel15              mpi/openmpi/4.0.0/gnu4.9              mpi/openmpi/4.0.0/intel15

----- /local/ap1/1x/modules/apl_ex -----
GRRM/11-g09                            espresso/5.1.2                          genesis/1.3.0-CUDA                    gromacs/2018.8/intel-CUDA              lammps/16Mar18/intel                    openmolcas/20190604
GRRM/14-g09(default)                  espresso/5.4                            genesis/1.4.0                          gromacs/2018.1/intel                  lammps/16Mar18/intel-CUDA              orca/4.2.1
abin1t/7.8.2                           espresso/6.1                            genesis/1.4.0-CUDA                    gromacs/2018.1/gnu                    lammps/22Aug18/intel                    psi4/1.1
abin1t/8.8.3(default)                  espresso/6.3(default)                  gromacs/2016.1/intel                  gromacs/2018.3/gnu                    lammps/22Aug18/intel-CUDA              reactionplus/1.0
amber/16/bugfix10                       gamess/2018Feb14                        gromacs/2016.1/intel-CUDA             gromacs/2018.3/intel                  lammps/22Aug18/intel-CUDA-volta        siesta/3.1
amber/16/bugfix15                       gamess/2018Sep30(default)              gromacs/2016.3/intel                  gromacs/2018.3/intel-CUDA             lammps/7Aug19/intel                    siesta/4.0.2(default)
amber/18/bugfix1                          gamess/2019Sep30                        gromacs/2016.3/intel-CUDA             gromacs/2018.6/gnu                    lammps/7Aug19/intel-CUDA              smash/2.2.0
amber/18/bugfix11-volta                  gaussian/g09/b01                        gromacs/2016.4/intel                  gromacs/2018.6/gnu-CUDA               gromacs/2019.4/gnu                    turbomole/7.2.1-MPI
amber/18/bugfix12                         gaussian/g09/c01                        gromacs/2016.4/intel-CUDA             gromacs/2018.6/intel                  gromacs/2019.4/intel                  turbomole/7.2.1-SMP
amber/18/bugfix16                         gaussian/g09/d01                        gromacs/2016.5/gnu                    gromacs/2018.6/intel-CUDA             gromacs/2019.4/intel-CUDA             namd/2.11
autodock/4.2.6                           gaussian/g09/e01                        gromacs/2016.5/gnu-CUDA               gromacs/2018.7/gnu                    gromacs/5.1.4/intel                  namd/2.11-CUDA
cp2k/6.1.0/gnu                            gaussian/g16/a03                        gromacs/2016.5/intel                  gromacs/2018.7/gnu-CUDA              gromacs/5.1.4/intel-CUDA              namd/2.13(default)
cp2k/6.1.0/gnu-CUDA                       gaussian/g16/b01                        gromacs/2016.5/intel-CUDA             gromacs/2018.7/intel                  gromacs/5.1.5/gnu                    namd/2.13-CUDA
cp2k/6.1.0/intel                          gaussian/g16/c01                        gromacs/2016.6/gnu                    gromacs/2018.7/intel-CUDA             gromacs/5.1.5/gnu-CUDA                namd/2.13-CUDA
cp2k/6.1.0/intel-CUDA                     genesis/1.1.6                            gromacs/2016.6/gnu                    gromacs/2018.8/gnu                    gromacs/5.1.5/intel                  ntchem/2013-5.0-mpi
crystal/14-104                            genesis/1.1.6-CUDA                       gromacs/2016.6/intel                  gromacs/2018.8/gnu-CUDA              gromacs/5.1.5/intel-CUDA              ntchem/2013-5.0-mpiomp
dirac/18.0                                genesis/1.3.0                            gromacs/2016.6/intel-CUDA             gromacs/2018.8/intel                  lammps/16Mar18/gnu                    ntchem/2013-5.0-serial
                                                                                        turbonamd/2.13-CUDA                   lammps/16Mar18/gnu-CUDA              nwchem/6.8

----- /local/ap1/1x/modules/apl_viewer -----
Tuscus/0.8.6 molten/5.7                 nbview/2.0                               vmd/1.9.3

----- /local/ap1/1x/modules/apl_util -----
allinea/7.1                             cmake/2.8.12.2(default)                 cmake/3.8.2

----- /local/ap1/1x/modules/lib -----
boost/1.53.0(default)                   boost/1.70.0                             mk1/2017.0.3                          mk1/2018.0.2                          mk1/2019.0.1                          spglib/1.11.1(default)
boost/1.59.0                             mk1/11.2.1                              mk1/2017.0.4                          mk1/2018.0.4(default)                 ncc1/2.3.7-1-cuda9.1(default)

----- /local/ap1/1x/modules/misc -----
inteldev_intellic pgilic
[user@ccfep5]$
```

(非標準バージョンのGCCはSoftware Collections (scl)のものを利用します)

環境の読み込み、切り替え

- 現在読み込んでいるバージョンから別のバージョンに切り替える場合は **module switch (モジュール名)** を実行します。
- 新規に環境を読み込む場合には **module load (モジュール名)** を実行します。以下、intel2017u8 と gcc7 環境の読み込み例を示します。

```
[user@ccfep4 ~]$ module list
Currently Loaded Modulefiles:
  1) intel_parallelstudio/2018update4   4) allinea/7.1
  2) pgilic                               5) cuda/9.1
  3) pgi/18.1
[user@ccfep4 ~]$ module switch intel_parallelstudio/2017update8
[user@ccfep4 ~]$ module load scl/devtoolset-7
[user@ccfep4 ~]$ module list
Currently Loaded Modulefiles:
  1) intel_parallelstudio/2017update8   4) allinea/7.1
  2) pgilic                               5) cuda/9.1
  3) pgi/18.1                           6) scl/devtoolset-7
[user@ccfep4 ~]$ icc -v
icc version 17.0.8 (gcc version 7.0.0 compatibility)
[user@ccfep4 ~]$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/opt/rh/devtoolset-7/root/usr/libexec/gcc/x86_64-redhat-linux/7/ltt-wrapper
Target: x86_64-redhat-linux
Configured with: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,lto --prefix=/opt/rh/devtoolset-7/root/usr --mandir=/opt/rh/devtoolset-7/root/usr/share/man --infodir=/opt/rh/devtoolset-7/root/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-shared --enable-threads=posix --enable-checking=release --enable-multilib --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --enable-plugin --with-linker-hash-style=gnu --enable-initfini-array --with-default-libstdcxx-abi=gcc4-compatible --with-isl=/builddir/build/BUILD/gcc-7.3.1-20180303/obj-x86_64-redhat-linux/isl-install --enable-libmpx --enable-gnu-indirect-function --with-tune=generic --with-arch_32=i686 --build=x86_64-redhat-linux
Thread model: posix
gcc version 7.3.1 20180303 (Red Hat 7.3.1-5) (GCC)
[user@ccfep4 ~]$
```

Intel 2017 u8 環境に
切り替え

GCC 7 環境を読み込み
(Software Collections を利用)

(**module purge** コマンドでモジュールをすべて unload することもできます)

ライブラリやユーティリティの利用(1)

- cmake や boost のような一部のソフトについてはシステム標準よりも新しいものが /local/apl/lx に導入されています。これらも module コマンドでロードできます。module avail 中の lib や apl_util にあります。

```
----- /local/apl/lx/modules/apl_util -----
allinea/7.1                cmake/2.8.12.2(default) cmake/3.8.2
----- /local/apl/lx/modules/lib -----
boost/1.53.0(default)      mk1/2017.0.3              mk1/2019.0.1
boost/1.70.0               mk1/2017.0.4              ncc1/2.3.7-1+cuda9.1(default)
boost/1.59.0               mk1/2018.0.2              spglib/1.11.1(default)
mk1/11.2.1                 mk1/2018.0.4(default)

```

2.8.12 がシステム標準(/usr/bin/cmake)です

1.53.0 はシステム標準(/usr/lib64/libboost*)です

```
[user@ccfep4 ~]$ module purge
[user@ccfep4 ~]$ module list
No Modulefiles Currently Loaded.
[user@ccfep4 ~]$ cmake --version
cmake version 2.8.12.2
[user@ccfep4 ~]$ module load intel_parallelstudio/2018update4
[user@ccfep4 ~]$ module load cmake/3.8.2
[user@ccfep4 ~]$ module load boost/1.70.0
[user@ccfep4 ~]$ module list
Currently Loaded Modulefiles:
  1) intel_parallelstudio/2018update4   3) boost/1.70.0
  2) cmake/3.8.2
[user@ccfep4 ~]$ cmake --version
cmake version 3.8.2

CMake suite maintained and supported by Kitware (kitware.com/cmake).
[user@ccfep4 ~]$
```


ライブラリやユーティリティの利用(2)

- Gromacs や Amber、 Gaussian 等のソフトについても module を用意しています。ジョブスクリプトでも利用できますが、フロントエンド(ccfep*)で簡単なテストをする際や解析プログラムを使う際にもこれらの module は有用かもしれません。

```
[user@ccfep8 ~]$ ampbdb
bash: ampbdb: command not found...
[user@ccfep8 ~]$ formchk
bash: formchk: command not found...
[user@ccfep8 ~]$ module list
Currently Loaded Modulefiles:
  1) intel_parallelstudio/2018update4   4) allinea/7.1
  2) pgilic                               5) cuda/9.1
  3) pgi/18.1
[user@ccfep8 ~]$ module load amber/18/bugfix12
[user@ccfep8 ~]$ ampbdb
Error: 'prmtop': No such file or directory

Usage: ampbdb -p 'Top' -c 'Coords' [Additional Options]
       ampbdb -p 'Top' < 'AmberRestart' [Additional Options]
  -p 'Top'      Topology file (default: prmtop).
  -c 'Coords'   Coordinate file.
  'AmberRestart' Amber restart file from STDIN.
  PDB is written to STDOUT.
  Use '-h' or '--help' to see additional options.

[user@ccfep8 ~]$ module load gaussian/g16/b01
[user@ccfep8 ~]$ module list
Currently Loaded Modulefiles:
  1) intel_parallelstudio/2018update4   5) cuda/9.1
  2) pgilic                               6) amber/18/bugfix12
  3) pgi/18.1                             7) gaussian/g16/b01
  4) allinea/7.1
[user@ccfep8 ~]$ formchk
Checkpoint file?
```

デフォルト環境では ampbdb や formchk は利用できない

amber18/bugfix12読み込み

パスが設定されたため、ampbdbが利用可能に(必要なファイルを用意していないため動作はしていません)

Gaussian を読み込むと、そのバージョンの formchk も利用可能

ヒント: 注意点(1)

- バージョンを細かく指定しない場合はデフォルトバージョンが読み込まれます。
 - また、場合によっては必要なパッケージも自動で読み込まれます。

```
% module purge
% module load amber gaussian
info: purge all the modules
info: loading intel_parallelstudio/2017update8
info: loading cuda/9.1
% module list
Currently Loaded Modulefiles:
  1) intel_parallelstudio/2017update8    3) amber/18/bugfix12
  2) cuda/9.1                            4) gaussian/g16/b01
```

自動設定の
結果出力

ヒント: 注意点(2)

- ログインシェルが /bin/csh で、ジョブスクリプトで module コマンドを使いたい場合、module コマンド利用前に“source /etc/profile.d/modules.csh”のように読み込む必要があります。(フロントエンド(ccfep*)で対話処理する際は不要です。)
- 単一のアプリについて複数バージョンの読み込みは原則できません。
 - intel_parallelstudio/2018update4 が読み込まれている状態で intel_parallelstudio/2017update8 を読み込むとエラーになります。
 - バージョン切り替え時には module switch (別バージョン) をご利用下さい
 - 一旦 unload して新規に load したり、一旦 purge してまっさらにしてから load することもできます。
- 一部アプリ(Anaconda等)については module 対応していません。この場合は設定ファイルを直接読み込んでください。

ヒント: コンパイラリスト(2020年1月時点)

- Intel Parallel Studio 2019 Update 5 (icc/icpc/ifort 19.0.5)
- Intel Parallel Studio 2019 Update 1 (icc/icpc/ifort 19.0.1)
- Intel Parallel Studio 2018 Update 4 (icc/icpc/ifort 18.0.5) (デフォルト)
- Intel Parallel Studio 2018 Update 2 (icc/icpc/ifort 18.0.2)
- Intel Parallel Studio 2017 Update 8 (icc/icpc/ifort 17.0.8)
- Intel Parallel Studio 2017 Update 4 (icc/icpc/ifort 17.0.4)
- Intel Parallel Studio 2015 Update 1 (icc/icpc/ifort 15.0.1)
- GCC 4.8.5 (デフォルト; システム標準のもの)
- GCC 8.3.1 (Devtoolset-8 Software Collections (scl))
- GCC 7.3.1 (Devtoolset-7 Software Collections (scl))
- GCC 6.3.1 (Devtoolset-6 Software Collections (scl))
- GCC 5.3.1 (Devtoolset-4 Software Collections (scl))
- GCC 4.9.2 (Devtoolset-3 Software Collections (scl))
- CUDA 8.0.61 (nvcc)
- CUDA 9.1.85 (nvcc) (デフォルト)
- CUDA 10.1.243 (nvcc)