

# Software Modules

Peter Weisbrod, SCC, KIT



# Software (=Environment) modules

By default manual setup of \$PATH, \$LD\_LIBRARY\_PATH ... for compilers, libraries and software packages etc.

→ Getting complicated if multiple versions of same software installed

## Solution:

- dynamic modification of the session environment by  
→ instruction sets stored in *modulefiles*

## HowTo?

- *load* and *unload* instruction sets (= modulefiles)
- How to use modulefiles in general?

```
$ module help
```

- More information:
  - about the tool in use: Lmod → <https://lmod.readthedocs.io/en/latest/>

# modulefiles: available / search

- Display all **available** modulefiles (modules which can be loaded directly)

`$ module avail` = `$ ml av`

```
----- /opt/bwhpc/ka/modulefiles -----
cae/abaqus/2024      cae/ansys/2023R2      cae/comsol/6.3      cae/starccm+/2310      cae/starccm+/2410
cae/ansys/2022R1     cae/ansys/2024R1     cae/starccm+/2302   cae/starccm+/2402     cae/starccm+/2502      (D)
cae/ansys/2023R1     cae/ansys/2025R1 (D) cae/starccm+/2306   cae/starccm+/2406     math/mathematica/14.2

----- /opt/bwhpc/common/modulefiles -----
cae/ansys/2024R2_no_license  lib/hdf5/1.14-gnu-14.2-serial
cae/openfoam/v2406           lib/hdf5/1.14-intel-2025.0-serial
cae/openfoam/v2412           (D) lib/hdf5/1.14-llvm-19.1-serial
compiler/aocc/5.0.0          lib/hdf5/1.14-nvidia-24.9-serial
compiler/gnu/11.4            lib/hdf5/1.14-nvidia-25.1-serial      (D)
compiler/gnu/14.2            (D) lib/icu4c/icu4c-77-1-gnu14.2
compiler/intel/2025.0        lib/netcdf/4.9-aocc-5.0.0-serial
compiler/llvm/19.1          lib/netcdf/4.9-gnu-11.4-serial
cs/ollama/0.5.11            lib/netcdf/4.9-gnu-14.2-serial
devel/code-server/4.96.4    (E) lib/netcdf/4.9-intel-2025.0-serial
devel/cuda/11.8              lib/netcdf/4.9-llvm-19.1-serial      (D)
devel/cuda/12.8              (D) math/R/4.4.2
devel/miniforge/24.11.0-python-3.12  math/julia/1.10.8      (E)
devel/python/3.11.7-gnu-11.4  math/julia/1.11.4      (E,D)
devel/python/3.11.7-gnu-14.2  math/matlab/R2024b
devel/python/3.12.3-gnu-11.4  mpi/impi/2021.14-intel-2025.0
```

- Search: Display all **available** „compiler“ modulefiles

`$ module avail compiler`

```
----- /opt/bwhpc/common/modulefiles -----
compiler/aocc/5.0.0      compiler/intel/2025.0      toolkit/nvidia-hpc-sdk/25.1-byo-compiler
compiler/gnu/11.4       compiler/llvm/19.1
compiler/gnu/14.2 (D)  toolkit/nvidia-hpc-sdk/24.9-byo-compiler

Where:
D: Default Module
```

# modulefiles: spider / search (1)

- Display all **possible** modulefiles

```
$ module spider
```

-----  
The following is a list of the modules and extensions currently available:  
-----

```
cae/abaqus: cae/abaqus/2024  
cae/ansys: cae/ansys/2022R1, cae/ansys/2023R1, cae/ansys/2023R2, cae/ansys/2024R1, cae/ansys/2024R2_no_license, ...  
cae/comsol: cae/comsol/6.3  
cae/openfoam: cae/openfoam/v2406, cae/openfoam/v2412  
cae/starccm+: cae/starccm+/2302, cae/starccm+/2306, cae/starccm+/2310, cae/starccm+/2402, cae/starccm+/2406, ...  
compiler/aocc: compiler/aocc/5.0.0  
compiler/gnu: compiler/gnu/11.4, compiler/gnu/14.2  
compiler/intel: compiler/intel/2025.0  
compiler/llvm: compiler/llvm/19.1  
cs/ollama: cs/ollama/0.5.11
```

- Search: Display all **possible** „gnu compiler“ modulefiles

```
$ module spider compiler/gnu
```

```
-----  
compiler/gnu:  
-----
```

```
Versions:  
  compiler/gnu/11.4  
  compiler/gnu/14.2
```

## modulefiles: spider / search (2)

- Some clusters (HoreKa, Helix, JUSTUS2) use hierarchical module system
- In hierarchical module system, availability depends on loaded modules
- Display all **possible variants** of a modulefiles

```
$ module spider mpi/openmpi/5.0
```

```
-----  
mpi/openmpi: mpi/openmpi/5.0  
-----
```

You will need to load all module(s)  
on any one of the lines below before  
the "mpi/openmpi/5.0" module is available  
to load.

```
compiler/gnu/.13.2  
compiler/gnu/13.3  
compiler/gnu/14.1  
compiler/intel/2023.1.0  
compiler/intel/2023.1.0_llvm  
compiler/intel/2024.0_llvm  
compiler/llvm/17.0  
compiler/llvm/18.1
```

# modulefiles: help / whatis

- Show help of modulefiles, e.g. `$ module help cae/openfoam`

```
---- Module Specific Help for "cae/openfoam/v2412" ----  
The OpenFOAM (Open Field Operation and Manipulation) CFD  
Toolbox can simulate anything from complex fluid flows  
...  
Local documentation in:  
  $FOAM_DOC_DIR  
...  
Online documentation in:  
  https://wiki.bwhpc.de/e/OpenFoam  
Homepage:  
  https://www.openfoam.org  
...  
In case of problems, create a ticket at  
'https://bw-support.scc.kit.edu/'
```

Version fallback is the  
defined default (here v2412)

- Show short info modulefile

```
$ module whatis cae/openfoam
```

```
cae/openfoam/v2412 : Open Source CFD Toolbox OpenFOAM version v2412
```

# modulefiles: show

- Show all instructions of modulefile

```
$ module show compiler/gnu/13
```

```
-----  
/software/all/lmod/modulefiles/Core/compiler/gnu/13.lua:  
-----  
...  
setenv("CC", "/opt/gcc/13/bin/gcc")  
setenv("CFLAGS", "-O2 -march=native")  
setenv("OMP_PROC_BIND", "true")  
...  
prepend_path("PATH", "/opt/gcc/13/bin")  
prepend_path("LD_LIBRARY_PATH", "/opt/gcc/13/lib64")  
...  
whatis("Sets up GCC C/C++/Fortran compiler version 13 in your environment...  
help([[The GNU Compiler Collection includes front ends for C, C++,  
Objective-C, Fortran, Java, Ada, and Go, as well as libraries for these  
Languages (libstdc++, libgcj,...). GCC was originally written as the  
compiler for the GNU operating system. The GNU system was developed  
to be 100% free software, free in the sense that it respects the  
user's freedom.  
  
In case of problems, please https://support.nhr.kit.edu/  
SCC support end: As soon as GNU compiler version 15 is available!  
]])  
prepend_path("MODULEPATH", "/software/all/lmod/modulefiles/Compiler/gnu/13")  
family("compiler")
```

Setting environment  
variables

Modifying environment  
variables

Content of printout  
functions

*module show* does NOT load the modulefile

# modulefiles: show

■ Show all instructions of modulefile

```
$ module show compiler/gnu/14.2
```

```
-----  
/opt/bwhpc/common/modulefiles/Core/compiler/gnu/14.2.lua:  
-----  
setenv("GNU_VERSION","14.2.0")  
setenv("GNU_HOME","/opt/bwhpc/common/compiler/gnu/14.2.0")  
setenv("GNU_BIN_DIR","/opt/bwhpc/common/compiler/gnu/14.2.0/bin")  
...  
prepend_path("PATH","/opt/bwhpc/common/compiler/gnu/14.2.0/bin")  
prepend_path("LD_LIBRARY_PATH","/opt/bwhpc/common/compiler/gnu/14.2.0/lib64")  
...  
conflict("compiler/intel")  
conflict("compiler/pgi")  
whatis("GNU compiler suite version 14.2.0 (gcc, g++, gfortran,...  
help([[This module provides the GNU compiler collection version 14.2.0  
via commands gcc, g++, gfortran and gccgo. The GNU compiler has been build ...  
...  
cpp      - GNU pre processor  
gcc      - GNU C compiler  
g++     - GNU C++ compiler  
gfortran - GNU Fortran compiler (Fortran 95/2003/2008 ...  
...  
In case of problems, submit a trouble ticket at  
'https://bw-support.scc.kit.edu'.  
  
The full version is: compiler/gnu/14.2.0  
]])
```

Setting environment variables

Modifying environment variables

Conflict setup

*module show* does NOT load the modulefile



# modulefiles: categories & dependencies

- Module names already implicate dependencies:

→ **Category/softwarename/version\_attributes-dependencies**

e.g. **devel/python/3.13.1-gnu-14.2**

→ python package version 3.13.1, compiled with GNU 14.2

- Categories:

compiler/	for compiler, e.g. intel, gnu, pgi, open64
devel/	for debugger, e.g. ddt, and development tools, e.g. cmake, itrac
mpi/	for MPI libraries, e.g. impi, openmpi, mvapich(2)
numlib/	for numerical libraries, e.g. Intel MKL, ACML, nag, gsl, fftw
lib/	for other libraries, e.g. netcdf, global array
bio/	for biology software, e.g. bowtie, abyss, mrbayes
cae/	for CAE software, e.g. ansys, abaqus, fluent
chem/	for chemistry software, e.g. gromacs, dacapo, turbomole
math/	for mathematics software, e.g. matlab, R
phys/	for physics software, e.g. geant4
vis/	for visualisation software, e.g. vmd, tigervnc

# Exercise 1

- 1. Find all modulefiles that start with „mpi“

## Exercise 1 - Solution

- 1. Find all modulefiles that start with „mpi“

```
$ module -t -r spider '^mpi'
```

```
mpi/impi/2021.14-intel-2025.0
```

```
mpi/openmpi/5.0-gnu-11.4
```

```
mpi/openmpi/5.0-gnu-14.2
```

# modulefiles: load / list

- Modulefiles are sorted in categories, software name and versions:

```
$ module load <category>/<software_name>/<version>
```

- Load a **default** software:

```
$ module load <category>/<software_name>
```

- e.g. Intel compiler

```
$ module load compiler/intel mpi/impi
```

→ loads currently Intel compiler suite 2025.0

→ loads currently Intel-MPI 2021.14 for Intel compiler suite 2025.0

- Display all loaded modules

```
$ module list = $ ml
```

Currently Loaded Modules:

1) compiler/intel/2025.0 2) mpi/impi/2021.14-intel-2025.0

# modulefiles: load dependencies /conflicts (1)

## Dependencies

- e.g.: some applications require particular compiler libraries

```
$ module load devel/python/3.11.7-gnu-11.4  
$ module list
```

autoloaded gnu  
compiler 11.4

Currently Loaded Modules:

1) compiler/gnu/11.4 2) devel/python/3.11.7-gnu-11.4

## Conflicts:

- a) load different software version in the same session, e.g. Intel:

```
$ module load compiler/gnu/11.4  
$ module load compiler/gnu/14.2
```

The following have been reloaded with a version change:  
1) compiler/gnu/11.4 => compiler/gnu/14.2

- b) load module with dependencies on other modules

```
$ module load compiler/gnu/14.2  
$ module load devel/python/3.11.7-gnu-11.4
```

requires gnu  
compiler 11.4

The following have been reloaded with a version change:  
1) compiler/gnu/14.2 => compiler/gnu/11.4

## Exercise 2

- 1. Load latest OpenMPI with default GNU compiler (Hint: Option `-d` to show only default version)

## Exercise 2 - Solutions

### 1. Load latest OpenMPI with default GNU compiler

```
$ module -d avail compiler/gnu
compiler/gnu/14.2

$ module load compiler/gnu/14.2

$ module -r spider 'mpi/openmpi.*'
→ mpi/openmpi/5.0

$ module load mpi/openmpi/5.0
```

```
# Pitfall on clusters with hierarchical modules: Loading openmpi before compiler
$ module load mpi/openmpi/5.0
```

```
Lmod has detected the following error:
These module(s) or extension(s) exist but cannot be loaded as requested:
"mpi/openmpi/5.0"
  Try: "module spider mpi/openmpi/5.0" to see how to load the module(s).
```

# modulefiles: unload/swap/purge

- To remove module *foo*:

```
$ module unload foo
```

→ be aware that you might create **inconsistencies**

```
$ module load mpi/impi/2021.14-intel-2025.0  
$ module unload compiler/intel/2025.0
```

Lmod Warning:

-----  
The following dependent module(s) are not currently loaded:  
compiler/intel/2025.0 (required by: compiler/intel/2025.0)  
-----

- Swap = remove + load

e.g.: 

```
$ module swap compiler/gnu compiler/intel
```

Removes loaded GNU  
version and loads  
default INTEL version

- To remove **ALL** modules at once:

```
$ module purge
```

```
$ module list  
No modules loaded
```



## Exercise 3

- 1. Determine system gcc version (without modulefile system)
  
  
  
  
  
  
  
  
  
  
- 2. Load newest gcc version (hint: newest → highest version number of compiler/gnu)

## Exercise 3 - Solution

- 1. Determine system gcc version (without modulefile system)

```
$ module purge
$ module list
No modules loaded
$ gcc --version
gcc (GCC) 11.4.1 20231218 (Red Hat 11.4.1-4)
```

- 2. Load newest gcc version (hint: newest → highest version number of compiler/gnu)

```
$ module spider compiler/gnu
Versions:
  compiler/gnu/11.4
  compiler/gnu/14.2
$ module load compiler/gnu/14.2
$ gcc --version
gcc (GCC) 14.2.0
```

# Private modulefiles

- Each user can create own modulefiles:

e.g. modulefiles that adds path of own programs, `$HOME/special`, to `$PATH`

→ content of this modulefile „*mybin.lua*“

```
-- Own Lua modulefile to prepend $HOME/special to $PATH
--
prepend_path("PATH", os.getenv("HOME") .. "/special")
```

→ place „*mybin.lua*“ under `$HOME/privatemodules`

→ to make all own modules visible to “module avail” command, enter:

```
$ module use $HOME/privatemodules
```

→ note: own module have higher priority than systems ones

```
$ module avail
```

```
-- /home/kit/ka_scc/ka_ab1234/privatemodules --
mybin
```

- Remove own modules:

```
$ module unuse $HOME/privatemodules
```