



File Systems

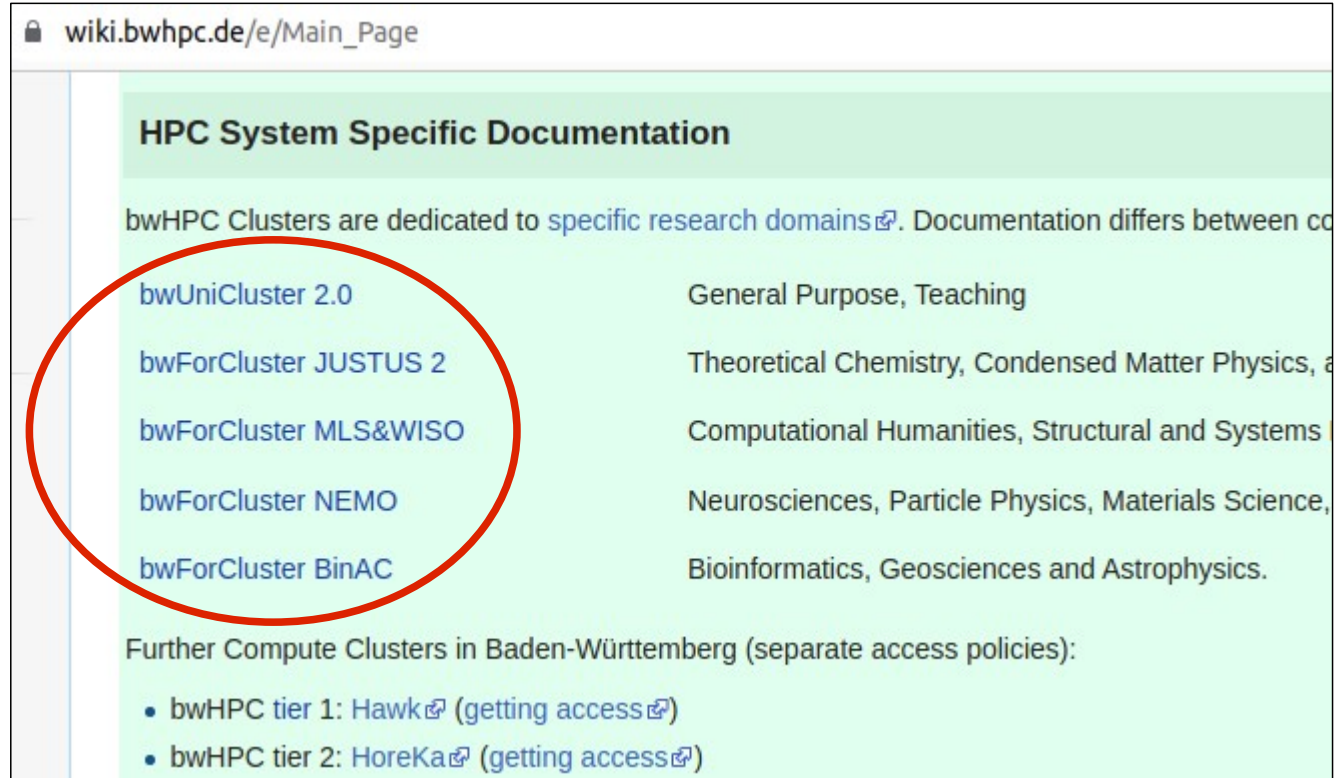
Mehmet Soysal, SCC, KIT



Reference: bwHPC Wiki

■ Most information given by this talk can be found at <https://wiki.bwhpc.de>

- select cluster
- then select File Systems



wiki.bwhpc.de/e/Main_Page

HPC System Specific Documentation

bwHPC Clusters are dedicated to [specific research domains](#). Documentation differs between co

bwUniCluster 2.0	General Purpose, Teaching
bwForCluster JUSTUS 2	Theoretical Chemistry, Condensed Matter Physics, a
bwForCluster MLS&WISO	Computational Humanities, Structural and Systems I
bwForCluster NEMO	Neurosciences, Particle Physics, Materials Science,
bwForCluster BinAC	Bioinformatics, Geosciences and Astrophysics.

Further Compute Clusters in Baden-Württemberg (separate access policies):

- [bwHPC tier 1: Hawk](#) ([getting access](#))
- [bwHPC tier 2: HoreKa](#) ([getting access](#))

Material: Slides & Scripts

■ https://indico.scc.kit.edu/e/bwhpc_course_2022-04-07

■ bwUniCluster:

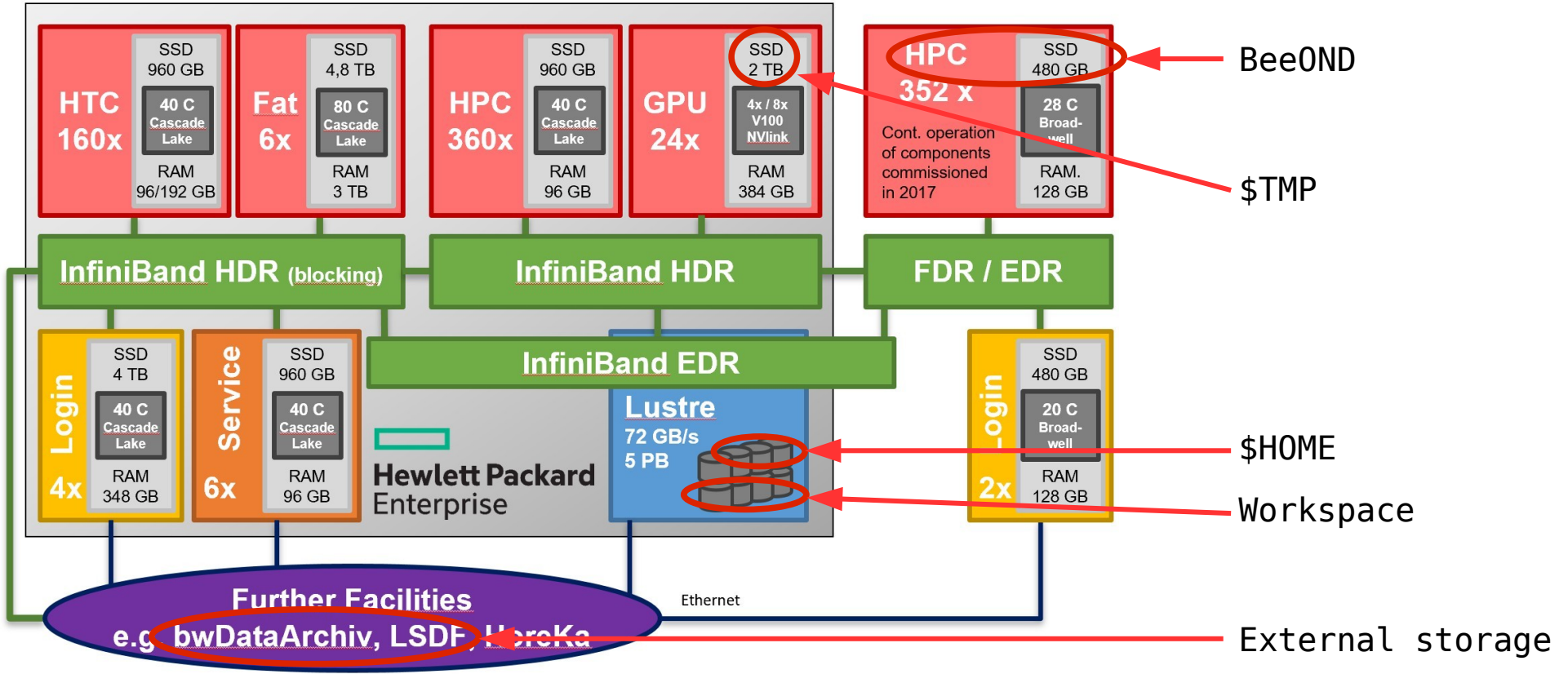
/opt/bwhpc/common/workshops/2022-04-07/

How to read the following slides

Abbreviation/Colour code	Full meaning
\$ command -option value	\$ = prompt of the interactive shell The full prompt may look like: user@machine:path\$ The command has been entered in the interactive shell session
<integer> <string>	<> = Placeholder for integer, string etc
foo, bar	Metasyntactic variables

File Systems

Example bwUniCluster 2.0: File Systems and their location



File System properties of bw clusters

Property	\$HOME	Workspace	\$TMP	BeeOND ¹	LSDF ¹
Visibility	all nodes	all nodes	local node	job nodes	login + job
Lifetime	permanent	few weeks	job runtime	job runtime	permanent
Usable capacity	40 GB - 1 TB	10 TB - 40 TB	128 GB - 7 TB	N * 500 GB	per project
Usable inodes	2 mil - unlinited	1 mil - unlimited	unlimited	unlimited	per project
Backup	yes, except MLS&WISO	no	no	no	yes
Total perf.	medium, 100s - 1000s MB/s	huge, 10s GB/s	100s MB/s per node	N * 100s MB/s	10s GB/s

¹ Only available on bwUniCluster 2.0

How to use each File System (1)

- \$HOME = Home directory
 - Software, configuration files, final results
 - **Omit** heavy I/O
- Workspaces = Working directories with lifetime
 - Intermediate results, huge input/output data sets
 - Scratch data which needs to be shared between nodes
 - **Omit** small files, tiny block sizes, lots of metadata operations
- \$TMP, \$TMPDIR, \$SCRATCH = Separate file system on each node using local disks
 - Data which is only needed during job runtime on the local node
 - Possibly transfer data here within a batch job
 - **All** sorts of I/O **allowed**

How to use each File System (2)

- BeeOND = Private file system for batch job
 - Data which is only needed during job runtime on the batch job nodes
 - Possibly transfer data here within a batch job
 - **All** sorts of I/O **allowed**, only available on bwUniCluster 2.0
- External storage
 - Archive scientific data, move data here when data sets become too large
 - Each organization has different solutions, example is LSDF at KIT
 - **Use** huge files or compressed archives
- Summary
 - Use \$HOME for permanent data
 - Use workspaces for huge files and sequential I/O
 - Use \$TMP or BeeOND with many (> 10000) small files or random I/O

\$HOME = Home directory

- \$HOME is visible on all nodes of a cluster
- Properties of \$HOME on different clusters

Cluster	Quota capacity limit	Quota file limit	Backup
JUSTUS 2	400 GB per user	2 mill. per user	Yes
MLS&WISO	200 GB per user	unlimited	No
NEMO	100 GB per user	unlimited	Yes
BinAC	40 GB per user	unlimited	Yes
BwUniCluster 2.0	1000 GB per user also limit per organization	10 mill. per user	Yes

\$HOME on bwUniCluster 2.0

■ HowTo goto:

```
$ cd $HOME
```

```
$ cd
```

■ User's quota usage and limits:

```
$ lfs quota -uh $(whoami) $HOME
```

■ Organization quota usage and limits:

```
$ lfs quota -ph $(grep $(echo $HOME | sed -e "s|/[^\ ]*/[^\ ]*$||") \  
/pfs/data5/project_ids.txt | cut -f 1 -d\ ) $HOME
```

Workspaces = Working directories with lifetime

- **Workspace:** lifetime on allocated folder
 - Available on all clusters, visible on all nodes of a cluster
 - HowTo:
 - <https://wiki.bwhpc.de/e/Workspace>

<code>\$ ws_allocate foo 10</code>	Allocate workspace <i>foo</i> for 10 days
<code>\$ ws_list</code>	List your workspaces
<code>\$ ws_find foo</code>	Get absolute path of workspace <i>foo</i>
<code>\$ ws_extend foo 5</code>	Extend lifetime of your workspace <i>foo</i> by 5 days from now. Number of extensions depends on cluster.
<code>\$ ws_release foo</code>	Manually erase your workspace <i>foo</i>
<code>\$ ws_... -F ffuc ...</code>	Select non default workspace file system with -F (works for any command)

Properties of Workspaces on different clusters

Cluster	Capacity limit	File limit	Max lifetime	Max extensions
JUSTUS 2	20 TB per user	5 mill. per user	90 days	unlimited
MLS&WISO	10 TB per user	unlimited	30 days	10 times
NEMO	10 TB per user	1 mill. per user	100 days	99 times
BinAC	unlimited	unlimited	30 days	3 times
BwUniCluster 2.0	40 TB per user	30 mill. per user	60 days	3 times

BeeOND = Private file system for batch job

■ BeeOND (BeeGFS On-Demand)

- Available only on **bwUniCluster 2.0**
- Private file system for batch job, created at job start and destroyed at job end
 - Make sure you have copied your data back to a workspace or \$HOME within your job
- Parallel file system, **visible on** nodes allocated to a **batch job**
- Uses local disks (SSDs) of each node to store the data
 - Capacity is limited: 500 GB * *number of nodes used in batch job*
- Request creation in job script or on command line:

```
#SBATCH --constraint=BEEOND
```

```
$ sbatch -C BEEOND ...
```

- Use path below `/mnt/odfs/${SLURM_JOB_ID}` to access BeeOND, e.g.

```
$ cd /mnt/odfs/${SLURM_JOB_ID}/stripe_default
```

- HowTo:

→ https://wiki.bwhpc.de/e/BwUniCluster_2.0_Hardware_and_Architecture#BeeOND_.28BeeGFS_On-Demand.29

LSDF Online Storage = External storage for special users

■ LSDF Online Storage

- Available only on bwUniCluster 2.0 for special KIT users
 - intended usage for scientific measurement data and data-intensive scientific simulation results
 - <https://www.scc.kit.edu/en/services/11228.php>
- Visible on login nodes and on batch job nodes if access was requested
 - Access from external with different protocols is also possible
- Request access in job script or on command line:

```
#SBATCH --constraint=LSDF
```

```
$ sbatch -C LSDF ...
```

- Use environment variables \$LSDF, \$LSDFPROJECTS, \$LSDFHOME to access, e.g.

```
$ cd ${LSDF}
```

- HowTo:

→ https://wiki.bwhpc.de/e/BwUniCluster_2.0_Hardware_and_Architecture#LSDF_Online_Storage

Exercise 1

■ Allocate two workspaces

```
$ ws_allocate test 30
```

```
Info: could not read email from users config ~/.ws_user.conf.
```

```
Info: reminder email will be sent to local user account
```

```
Info: creating workspace.
```

```
/pfs/work7/workspace/scratch/myuser-test
```

```
remaining extensions : 3
```

```
remaining time in days: 30
```

```
$ ws_allocate scratch 50
```

```
Info: could not read email from users config ~/.ws_user.conf.
```

```
Info: reminder email will be sent to local user account
```

```
Info: creating workspace.
```

```
/pfs/work7/workspace/scratch/myuser-scratch
```

```
remaining extensions : 3
```

```
remaining time in days: 50
```

Exercise 2

■ List workspaces

```
$ ws_list
id: scratch
  workspace directory : /pfs/work7/workspace/scratch/myuser-scratch
  remaining time      : 49 days 23 hours
  creation time       : Wed Oct  6 18:59:11 2021
  expiration date     : Thu Nov 25 17:59:11 2021
  filesystem name     : pfs5wor7
  available extensions : 3
id: test
  workspace directory : /pfs/work7/workspace/scratch/myuser-test
  remaining time      : 29 days 23 hours
  creation time       : Wed Oct  6 18:55:17 2021
  expiration date     : Fri Nov  5 17:55:17 2021
  filesystem name     : pfs5wor7
  available extensions : 3
```


Exercise 3

- Find workspace path and switch to it

```
$ ws_find scratch
/pfs/work7/workspace/scratch/myuser-scratch

$ ws_find test
/pfs/work7/workspace/scratch/myuser-test

$ cd $(ws_find test)
$ pwd
/pfs/work7/workspace/scratch/myuser-test
```

Exercise 4

■ Extend the lifetime of a workspace

```
$ ws_extend test 60
Info: could not read email from users config ~/.ws_user.conf.
Info: reminder email will be sent to local user account
Info: extending workspace.
Info: changed mail address to myuser
Info: changed reminder setting.
/pfs/work7/workspace/scratch/myuser-test
remaining extensions : 2
remaining time in days: 60
$ ws_list test
id: test
    workspace directory : /pfs/work7/workspace/scratch/myuser-test
    remaining time      : 59 days 23 hours
    creation time       : Wed Oct 6 19:01:02 2021
    expiration date     : Sun Dec 5 18:01:02 2021
    filesystem name     : pfs5wor7
    available extensions : 2
```

Exercise 5

■ Release workspaces

```
$ ws_release test
```

```
$ ws_release scratch
```

```
$ ws_list
```

Software Modules

Robert Barthel, SCC, KIT



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Hochschule
für Technik
Stuttgart



Hochschule Esslingen
University of Applied Sciences

Universität
Konstanz



UNIVERSITÄT
MANNHEIM



Universität Stuttgart

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



KIT
Karlsruher Institut für Technologie



ulm university universität
uulm



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

```
$ module avail = $ ml av
```

```
----- /opt/bwhpc/common/modulefiles/Core -----  
bio/freesurfer/6.0.0                devel/code-server/3.12.0          (E)    lib/CGAL/5.3  
bio/fsl/6.0.4                       devel/code-server/4.0.2          (E,D)  lib/hdf5/1.12.0-pgf  
bio/nest/2.18.0                     (T)    devel/cuda/9.2                   (E)    lib/hdf5/1.12.0-gnu  
bio/nest/2.20.1                     (T,D)  devel/cuda/10.0                  (E)    lib/hdf5/1.12.0-gnu  
cae/cgns/3.4.1-intel-19.1           devel/cuda/10.1                  (E)    lib/hdf5/1.12.0-int  
cae/cgns/4.1.2-gnu-8.3              (D)    devel/cuda/10.2                  (E)    lib/hdf5/1.12.0-int  
cae/openfoam/v2006-impi              devel/cuda/11.0                  (E)    lib/hdf5/1.12.1-gnu  
cae/openfoam/v2006                  devel/cuda/11.4                  (E,D)  lib/hdf5/1.12.1-gnu  
cae/openfoam/v2012                  devel/cudnn/9.2                  (E)    lib/hdf5/1.12.1-int  
cae/openfoam/v2106-impi              devel/cudnn/10.0                 (E)    lib/hdf5/1.12.1-int  
cae/openfoam/v2112                  devel/cudnn/10.1                 (E)    lib/hdf5/1.12.1-int  
cae/openfoam/4.1-extend              devel/cudnn/10.2                 (E,D)  lib/netcdf/4.7.3-ir  
cae/openfoam/5.x-impi                devel/forge/20.1                 lib/netcdf/4.7.4-gr  
cae/openfoam/5.x                    devel/adb/9.2                    math/R/3.6.3
```

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

```
$ module avail compiler
```

```
----- /opt/bwhpc/common/modulefiles/Core -----  
compiler/clang/9.0                  compiler/gnu/11.2                 compiler/llvm/11.0  
compiler/gnu/8.3.1                  compiler/intel/19.0               compiler/llvm/12.0  
compiler/gnu/9.3                    compiler/intel/19.1               compiler/llvm/13.0  
compiler/gnu/10.2 (D)               compiler/intel/2021.4.0_llvm      compiler/pgi/2020
```

modulefiles: spider / search (1)

- Display all **possible** modulefiles

```
$ module spider
```

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

```
bio/freesurfer: bio/freesurfer/6.0.0  
  
bio/fsl: bio/fsl/6.0.4  
  
bio/nest: bio/nest/2.18.0, bio/nest/2.20.1  
  NEST is a command line tool for simulating neural networks  
  
cae/abaqus: cae/abaqus/2020  
  
cae/ansys: cae/ansys/2020R2, cae/ansys/2021R2  
  
cae/cgns: cae/cgns/3.4.1-intel-19.1, cae/cgns/4.1.2-gnu-8.3  
  
cae/comsol: cae/comsol/5.6  
  
cae/cst: cae/cst/2020  
  
cae/lsdyna: cae/lsdyna/9.3.1, cae/lsdyna/12.0.0  
  
cae/openfoam: cae/openfoam/v2006-impi, cae/openfoam/v2006, cae/openfoam/v2012, cae/openfoam/v2106-impi, ...
```

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

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

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

```
Versions:  
  compiler/gnu/8.3.1  
  compiler/gnu/9.3  
  compiler/gnu/10.2  
  compiler/gnu/10.3  
  compiler/gnu/11.1  
  compiler/gnu/11.2
```

modulefiles: spider / search (2)

- Display all **possible variants** of a modulefiles

```
$ module spider mpi/openmpi/4.0
```

```
-----  
mpi/openmpi: mpi/openmpi/4.0  
-----
```

You will need to load all module(s)
on any one of the lines below before
the "mpi/openmpi/4.0" module is available
to load.

```
compiler/clang/9.0  
compiler/gnu/10.2  
compiler/gnu/8.3.1  
compiler/gnu/9.3  
compiler/intel/19.0  
compiler/intel/19.1  
compiler/llvm/10.0  
compiler/pgi/2020
```


modulefiles: help / whatis

- Show help of modulefiles, e.g. `$ module help chem/turbomole`

```
--Module Specific Help for "chem/turbomole/7.4.1" -----  
-----  
| Loading Parallel version          |  
-----  
* Code_words are: SMP (shared memory parallel) and  
                  MPI (message passing interface)  
* To load for e.g. SMP, execute:  
  export PARA_ARCH=SMP  
  module load chem/turbomole/7.4.1  
...  
...  
-----  
| Support                          |  
-----  
...
```

Version fallback is the defined default (here 7.4.1)

- Show short info modulefile

```
$ module whatis chem/turbomole
```

```
chem/turbomole/7.4.1 : Quantum chemistry package Turbomole version 7.4.1
```

modulefiles: show

- Show all instructions of modulefile

```
$ module show compiler/gnu/11.2
```

```
-----  
/opt/bwhpc/common/modulefiles/Core/compiler/gnu/11.2.lua:  
-----  
setenv("GNU_VERSION","11.2.0")  
setenv("GNU_HOME","/opt/bwhpc/common/compiler/gnu/11.2.0")  
setenv("GNU_BIN_DIR","/opt/bwhpc/common/compiler/gnu/11.2.0/bin")  
...  
prepend_path("PATH","/opt/bwhpc/common/compiler/gnu/11.2.0/bin")  
prepend_path("LD_LIBRARY_PATH","/opt/bwhpc/common/compiler/gnu/11.2.0/lib64")  
...  
conflict("compiler/intel")  
conflict("compiler/pgi")  
whatis("GNU compiler suite version 11.2.0 (gcc, g++, gfortran,...  
help([[This module provides the GNU compiler collection version 11.2.0  
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/11.2.0  
]])
```

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. **numlib/petsc/3.13.4-gnu-10.2-openmpi-4.0**

→ PETSc package version 3.13.4, compiled with GNU 10.2 and OpenMPI 4.0

- 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/2019  
mpi/impi/2020  
mpi/impi/2021.4.0  
mpi/openmpi/default  
mpi/openmpi/4.0  
mpi/openmpi/4.1
```

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 19.1

→ loads currently Intel-MPI 2020 for Intel compiler suite 19.1

- Display all loaded modules

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

Currently Loaded Modules:

1) compiler/intel/2021.4.0 2) mpi/impi/2021.4.0



modulefiles: load dependencies /conflicts (1)

Dependencies

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

```
$ module load numlib/gsl/2.6-intel-19.1  
$ module list
```

Currently Loaded Modules:

1) compiler/intel/19.1 2) numlib/gsl/2.6-intel-19.1

Autoloaded Intel-Suite 19.1

Conflicts:

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

```
$ module load compiler/intel/19.0  
$ module load compiler/intel/2021.4.0
```

The following have been reloaded with a version change:
1) compiler/intel/19.0 => compiler/intel/2021.4.0

- b) load module with dependencies on other modules

```
$ module load compiler/intel/2021.4.0  
$ module load numlib/gsl/2.6-intel-19.1
```

Requires Intel-Suite 19.1

The following have been reloaded with a version change:
1) compiler/intel/2021.4.0 => compiler/intel/19.1

Exercise 2

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

Exercise 2 - Solutions

1. Load latest OpenMPI with default INTEL compiler

```
$ module -d avail compiler/intel
compiler/intel/2021.4.0

$ module load compiler/intel/2021.4.0

$ module -r spider '.*openmpi.*'
→ mpi/openmpi/4.1

$ module load mpi/openmpi/4.1
```

```
# Pitfall: Loading openmpi before compiler
```

```
$ module load mpi/openmpi/4.1
```

Lmod has detected the following error:

These module(s) or extension(s) exist but cannot be loaded as requested:

"mpi/openmpi/4.1"

Try: "module spider mpi/openmpi/4.1" 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 numlib/gsl/2.6-intel-19.1  
$ module unload compiler/intel/19.1
```

```
-----  
The following dependent module(s) are not currently loaded:  
compiler/intel/19.1 (required by: numlib/gsl/2.6-intel-19.1)  
-----
```

- Swap = remove + load

e.g.:

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

Removes default 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) 8.4.1 20200928 (Red Hat 8.4.1-1)
```

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

```
$ module spider compiler/gnu
Versions:
  compiler/gnu/8.3.1
  compiler/gnu/9.3
  ...
  compiler/gnu/11.1
  compiler/gnu/11.2
$ module load compiler/gnu/11.1
$ gcc --version
gcc (GCC) 11.1.0
```

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

- Remove own modules:

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

```
$ module avail
```

```
--- /home/kit/scc/ab1234/privatemodules ---
    mybin
-----
```