What are Environment Modules?

Tell Me

1. The environment modules package is a tool that allows you to quickly and easily modify your shell environment to access different software packages.
2. Research Computing offers a large (and growing) number of software packages to users, and each package may contain several tools, manual pages and libraries, or it may require special setup to work properly.
3. Some software packages come in several versions or flavors, many of which conflict with each other.
4. Modules allow you to tailor your shell to access exactly the packages you need by setting up the relevant environment variables for you, and automatically avoiding many possible conflicts between packages.

Command Summary

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>module avail</td>
<td>List available modules</td>
</tr>
<tr>
<td>module load</td>
<td>Load module named</td>
</tr>
<tr>
<td>module unload</td>
<td>Unload module named</td>
</tr>
<tr>
<td>module whatis</td>
<td>Give description of module</td>
</tr>
<tr>
<td>module list</td>
<td>List modules that are loaded in your environment</td>
</tr>
<tr>
<td>module purge</td>
<td>Unload all currently loaded modules from your environment</td>
</tr>
<tr>
<td>module display</td>
<td>Give the rules for module</td>
</tr>
</tbody>
</table>

Example Usage
$ module avail

------------------------- /usr/share/Modules/modulefiles ----------------------------
dot module-git module-info modules null use.own

---------------------- /apps/usr/modules/compilers ----------------------------------
bazel/0.3.0          intel/14.0.3          pymods/2.7.12 scala/2.10.4(default)
bazel/0.4.5(default)  intel/16.0.0(default) pymods/2.7.5  scala/2.11.7
gcc/4.9.3           perlmods/5.16.3       pypy/5.3.1     yasm/1.3.0
gcc/5.3.0(default)   pgi/14.4             python/2.7.12
ghc/7.10.3          pgi/15.9(default)     python/3.5.1

------------------------------- /apps/usr/modules/lib -------------------------------
cblas/1.10           hdf5/1.8.16-pgi netcdf/4.4.0(default)
glew/1.13.0          hdf5/1.8.16-pgi-mpi netcdf/4.4.0-intel-mpi
google-code/2015     htslib/1.4.1          netcdf/4.4.0-mpi
hdf/0.2.11(defaut)   libgomp/4.9.997     netcdf/4.4.0-pgi
hdf/4.2.11-intel     netcdf/4.3.3.1     netcdf/4.4.0-pgi
hdf/4.2.11-pgi       netcdf/4.3.3.1-intel netcdf/4.4.0-pgi-mpi
hdf5/1.8.16(default)  netcdf/4.3.3.1-intel-mpi openblas/0.2.18-nehalem
hdf5/1.8.16-intel    netcdf/4.3.3.1-pgi  openblas/11.14.1-mpi
hdf5/1.8.16-mpi      netcdf/4.3.3.1-pgi-mpi trilinos/11.14.1-mpi

------------------------------- /apps/usr/modules/apps -------------------------------
abaqus/2017            gromacs/5.1.2(default)      poy/5.1.2-ib(default)
abaqus/6.10-2         gromacs/5.1.2-avx2       qlime/1.9.1
abaqus/6.13-4(defaut)  gromacs/5.1.2-cuda       quickflash/1.0.0
abyss/1.9.0           gromacs/5.1.2-cuda-avx2 quickflash/1.0.0-ib
abyss/1.9.0-ib       gromacs/5.1.2-mpi        R/3.1.1
allpaths-lg/52488     gromacs/5.1.2-mpi-avx2  R/3.2.13(default)
ansa/13.13.10         gromacs/5.1.2-mpi-cuda   R/3.3.1
art/03.19.15          gromacs/5.1.2-mpi-cuda-avx2 raxml/7.4.2
ascidicod/8.6.9       gromacs/5.1.2-mpi-ib       raxml/7.4.2-mpi
augustus/3.2.3        gromacs/5.1.2-mpi-ib-avx2 raxml/8.2.4(default)
bamtools/2.4.1        itk/4.9.0             raxml/8.2.4-mpi
bcftools/1.3.1        jellyfish/2.2.6        repdenovo/0.0
bedtools2/2.26.0      lammps/15May15         rosetta/2015.02
bioconductor/3.2.1    lammps/16Feb16(default)  rosetta/2016.10(default)

$ module avail matlab

--------------------- /apps/usr/modules/apps ----------------------
matlab/R2014a      matlab/R2015b(default)     matlab/R2016b

$ module display matlab/R2015b

-------------------------------------------------------------------
$ module load matlab/R2015b

$ module list
Currently Loaded Modulefiles:
1) pymods/2.7.3  2) perlmods/5.16.3  3) matlab/R2015b

How the Modules are Organized and Grouped
The modules are organized into "categories", which include: /apps/usr/modules/mpi, compilers, apps, and /apps/sys/Modules/3.2.6/modulefiles. Under each category, you will see "groups" of applications: openmpi, intel, pgi, to name a few. Within each group, there may be several versions to choose from. The group and version are separated with a "slash" (/).

**Default Modules**

You probably noticed some modules listed above are suffixed with a "(default)". The "default" module is the module that will get loaded if you do not specify a version number. For example, we can load the "intel/16.0.0" module by omitting the version number:

```bash
$ module load intel

$ module list
Currently Loaded Modulefiles:
  1) pymods/2.7.5
  2) perlmods/5.16.3
  3) intel/16.0.0
```

Note: If you plan to load a version of a module that is not the default, then you must specify the version in the module load command.

**Conflicts and Prerequisites**

Some modules conflict with others, and some modules are prerequisites of others. Environment Modules handles both scenarios.

The following is an example of trying to load a module that is dependent upon another:

```bash
$ module display gromacs/4.6.7-cuda
```

```
module-whatis GROMACS is a versatile package to perform molecular dynamics, 
i.e. simulate the Newtonian equations of motion for systems with hundreds 
to millions of particles. It is primarily designed for biochemical molecules 
like proteins, lipids and nucleic acids that have a lot of complicated bonded 
interactions, but since GROMACS is extremely fast at calculating the nonbonded 
interactions, many groups are also using it for research on non-biological 
systems, e.g. polymers.
conflict gromacs
prereq cuda
```

```bash
setenv GROMACS /apps/pkg/gromacs-4.6.7/rhel7_u2-x86_64/gnu-cuda
setenv GMXBIN /apps/pkg/gromacs-4.6.7/rhel7_u2-x86_64/gnu-cuda/bin
setenv GMXLDLIB /apps/pkg/gromacs-4.6.7/rhel7_u2-x86_64/gnu-cuda/lib
setenv GMXDATA /apps/pkg/gromacs-4.6.7/rhel7_u2-x86_64/gnu-cuda/share
setenv GNUXMAN /apps/pkg/gromacs-4.6.7/rhel7_u2-x86_64/gnu-cuda/share/man
setenv GMXLIB /apps/pkg/gromacs-4.6.7/rhel7_u2-x86_64/gnu-cuda/share/gromacs/top
prepend-path PATH /apps/pkg/gromacs-4.6.7/rhel7_u2-x86_64/gnu-cuda/bin
prepend-path MANPATH /apps/pkg/gromacs-4.6.7/rhel7_u2-x86_64/gnu-cuda/share/man
prepend-path LD_LIBRARY_PATH /apps/pkg/gromacs-4.6.7/rhel7_u2-x86_64/gnu-cuda/lib
```

```bash
$ module load gromacs/4.6.7-cuda
```

```
gromacs/4.6.7-cuda(12):ERROR:151: Module 'gromacs/4.6.7-cuda' depends on one of the module(s) 
gromacs/4.6.7-cuda(12):ERROR:102: Tcl command execution failed: prereq cuda
```

To resolve the above error, simply load the "prereq" module first, then load the original module. First you might want to see what "cuda" modules are available:

```bash
$ module avail cuda
```

```
---------------------- /apps/usr/modules/gpu ----------------------
cuda/7.5(default)    cuda/8.0
```

Select one that you would like to load to satisfy Gromacs's requirement. You can do this in a single command:

```bash
$ module load cuda/8.0 gromacs/4.6.7-cuda
```

**More information**

You can find more information about Environment Modules on SourceForge.net [http://modules.sourceforge.net/](http://modules.sourceforge.net/)

**Related FAQs**

- High-Performance Computing (HPC)
- What are Environment Modules?
- How do I schedule a job with Torque?
- Why did I receive a "MOAB job resource violation" email?
- Can I transfer files using the URC batch scheduler (Torque/Moab)?