Slide 1

Administrivia

- Remember that project proposals are due next Monday.

Slide 2

The “Modules” Package

- You may have heard that to make use of some non-default versions of things you should “use the `module` command”? and that you could put such a command in your `.bashrc` file?
- `module` is the user interface to the Environment Modules package.
- (Whether this is Linux-specific or available for other UNIX-like systems I do not know! maybe look carefully at the project Web page `modules.sourceforge.net`.)

Slide 3

Environment Modules

- Basic idea is to provide a clean method for modifying, in a reversible way, environment variables such as `$PATH`.
- Command `module` (actually a shell alias) performs various functions using “module files”. Environment variable `$MODULEPATH` lists directories that will be searched.

Slide 4

`module` Command Subcommands

- `avail` to list available modules; `show` or `display` to show information about a particular module.
- `list` to show currently loaded modules.
- `load` or `add` to load a module.
- `unload` or `rm` to load a module; `purge` to unload all modules.
- `help` to — what it says.
- `use` to add to the search path for modules.

Slide 5

Module Files

- What happens when you “load” a module — system processes a “module file”.
- System-wide modules are (typically?) in `/usr/share/Modules/modulefiles` (ones that come with package) and `/etc/modulefiles` (installation-specific).
- Users can have their own modules as well.
- Syntax for writing module files is — well, look at examples?

Slide 6

Minute Essay

- (None really — sign in?)