

Lmod 8.6 New Features

Robert McLay

Nov. 9, 2021

Lmod 8.6 New Features



- ▶ Release of Lmod 8.6
- ▶ New Features in Lmod web page
- ▶ `$LMOD_QUARANTINE_VARS`
- ▶ `/etc/lmod/lmod_config.lua` configuration file
- ▶ `source_sh()` sourcing shell scripts inside a modulefile
- ▶ `LmodBreak()`: Stop processing modulefile but keep going!

New Features in Lmod web page

- ▶ https://lmod.readthedocs.io/en/latest/025_new.html
- ▶ A place where new features are and will continue to be announced.

\$LMOD_QUARANTINE_VARS

- ▶ A module at TACC turn-off \$LMOD_PAGER
- ▶ This#%& module made me mad.
- ▶ Tmod has a new feature kinda like this.
- ▶ \$LMOD_QUARANTINE_VARS was invented.

\$LMOD_QUARANTINE_VARS (II)

- ▶ `export LMOD_QUARANTINE_VARS=LMOD_PAGER:LMOD_REDIRECT`
- ▶ This means that a module can't change those variables.
- ▶ This only works with regular env. vars.
- ▶ You can't quarantine PATH like variables.
- ▶ A user sets this variable in their `/.bashrc` or similar file.
- ▶ This obviously won't work for modules loaded during the processing of `/etc/profile.d/*.sh` files

`/etc/lmod/lmod_config.lua` configuration file

- ▶ This file is evaluated during Lmod startup.
- ▶ This location is the default during configuration.
- ▶ A site can change this location at configuration.

```
-- Example /etc/lmod/lmod_config.lua
require("strict")
local cosmic = require("Cosmic"):singleton()

cosmic:assign("LMOD_SITE_NAME", "XYZZY")
local function foo()
    ...
end
sandbox_registration { foo = foo }
```

Sourcing shell scripts inside a modulefile w/ `source_sh()`

- ▶ This was implemented in Tmod 4.7
- ▶ Xavier told me that he did this during Covid Lockdown in France.
- ▶ Lmod 8.6 re-implements this feature in a similar way.
- ▶ It knows about env. vars and shell functions and aliases.

source_sh() Implementation

- ▶ It captures the env. vars/functions/alias before and after the running the shell script.
- ▶ It computes the difference and saves it into the ModuleTable in env.
- ▶ It can be safely unloaded, shown.
- ▶ script path and arguments must not change between load and unload.
- ▶ `module refresh` works
- ▶ Obvious points:
 - ▶ It is better to use `sh_to_modulefile` and convert once.
 - ▶ But `sh_to_modulefile` is not dynamic (e.g. `$HOME`)
 - ▶ Can't have run the script in the user environment before loading the script.

ml -mt

```
_ModuleTable_ = {  
  MTversion = 3,  
  mT = {  
    wrapperSh = {  
      fn = "/Users/mclay/w/lmod/rt/sh_to_modulefile/mf/wrapperSh/1.0.lua",  
      fullName = "wrapperSh/1.0",  
      loadOrder = 1,  
      mcmdT =  
        ["/home/user/w/lmod/rt/sh_to_modulefile/second.sh arg1"] = {  
          "setenv(\"SECOND\",\"FOO_BAR\")",  
        },  
        ["/Users/mclay/w/lmod/rt/sh_to_modulefile/tstScript.sh"] = {  
          "setenv(\"MY_NAME\",\"tstScript.sh\")",  
          "prepend_path(\"PATH\", \"/Users/mclay/w/lmod/rt/sh_to_modulefile/bin\")",  
          "set_alias(\"fooAlias\", \"foobin -q -l\")",  
          [[set_shell_function("banner", " \\  
local str=\"$1\";\\  
local RED='\27[1;31m';\  
local NONE='\27[0m';\  
echo \"$RED$str$NONE\"\  
")]],  
        },  
      },  
    },  
  },  
}
```

LmodBreak()

- ▶ Tmod3 had a function called `break`
- ▶ It stop processing current module
- ▶ But it kept going otherwise.
- ▶ So it is different than erroring out.
- ▶ Lmod: If there is an error, no modules are loaded.

LmodBreak() (II)

- ▶ Lmod 8.6 adds `LmodBreak(msg)` function.
- ▶ TCL Modules use `break msg`
- ▶ No changes to environment from current module are kept.
- ▶ All other changes are kept.

LmodBreak() (III)

```
--Stdenv.lua  
load(A)  
load(BRK)  
load(C)
```

```
--A/1.0.lua  
setenv("A", "1.0")
```

```
--BRK/1.0.lua  
setenv("BRK", "1.0")  
LmodBreak()
```

```
--C/1.0.lua  
setenv("C", "1.0")
```

ml StdEnv

```
export A=1.0  
export C=1.0
```

Conclusions

- ▶ Many new features added to Lmod 8.6
- ▶ `$LMOD_QUARANTINE_VARS`
- ▶ `/etc/lmod/lmod_config.lua` configuration file
- ▶ `source_sh()`
- ▶ `LmodBreak()`:

Future Topics

- ▶ Lmod Testing System?
- ▶ More internals of Lmod?
- ▶ Settag?
- ▶ collections?
- ▶ Guest Presentation of special issues?