



Meeting of the Technical Steering Committee (TSC) Board

Tuesday, August 14th 2018
11:00am ET

Meeting Logistics

- <https://zoom.us/j/556149142>
- United States : +1 (646) 558-8656
 - Meeting ID: 556 149 142

Antitrust Policy Notice

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.
- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at <http://www.linuxfoundation.org/antitrust-policy>. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrave of the firm of Gesmer Updegrave LLP, which provides legal counsel to the Linux Foundation.

Agenda

- Meeting schedule starting September
 - I will have conflict at existing time on Tue/Thur
 - Would like to see if folks are amenable to moving TSC meeting to same time on either Wednesday or Friday?
 - Need to implement first week of September
- Working example setting CPATH with OHPC modulefile conventions
- Discussion: possible registration of ohpc packages with other systems (e.g. spack)

Modulefile conventions (cont. from last time)

- A question spurred by PR has come up on GitHub regarding conventions for exposing header files for compilation (<https://github.com/openhpc/ohpc/pull/791>)
- Following up from last time, posted a working example to the PR on how to define global hooks in Lmod to set CPATH and LIBRARY_PATH based on our ohpc modulefile conventions (thanks to Cyrus for spurring a change in Lmod to make this work out of the box)
 - triggers if `OHPC_MOD_SET_CPATH=1`
 - enabled by defining custom load and unload hook in:
`/opt/ohpc/admin/lmod/lmod/libexec/SitePackage.lua`

Modulefile conventions (cont. from last time)

Load Hook

```
function ohpc_cpath_set(t)

    if (os.getenv("OHPC_MOD_SET_CPATH") == "1" ) then

        local modFile = io.open(t.fn,"r")

        local includeDir=""
        local libDir=""

        for line in modFile:lines() do

            includeDir=string.match(line,"^setenv%s**%w*_INC%s*(%S*)")
            if includeDir ~= nil then
                prepend_path("CPATH",includeDir)
            end

            libDir=string.match(line,"^setenv%s**%w*_LIB%s*(%S*)")
            if libDir ~= nil then
                prepend_path("LIBRARY_PATH",libDir)
            end
        end

        modFile:close()

    end

end
```

Unload Hook

```
function ohpc_cpath_unset(t)

    if (os.getenv("OHPC_MOD_SET_CPATH") == "1" ) then

        local modFile = io.open(t.fn,"r")

        local includeDir=""
        local libDir=""

        for line in modFile:lines() do
            includeDir=string.match(line,"^setenv%s**%w*_INC%s*(%S*)")
            if includeDir ~= nil then
                remove_path("CPATH",includeDir)
            end

            libDir=string.match(line,"^setenv%s**%w*_LIB%s*(%S*)")
            if libDir ~= nil then
                remove_path("LIBRARY_PATH",libDir)
            end
        end
        modFile:close()

    end

    hook.register("load", ohpc_cpath_set)
    hook.register("unload",ohpc_cpath_unset)
```

Scans modulefile being loaded/unloaded and looks for *_INC, *_LIB settings to modify CPATH and LIBRARY_PATH

Modulefile conventions (cont. from last time)

Example in action:

- Use existing openblas module from ohpc
- Note: need latest Lmod release which we have landed on 1.3.6 branch

Set env variable to opt-in to set CPATH and LIBRARY_PATH

```
$ export OHPC_MOD_SET_CPATH=1
```

NOTE: nothing set initially

```
$ echo $CPATH $LIBRARY_PATH
```

Now, load openblas

```
$ module load openblas
$ echo $CPATH $LIBRARY_PATH
/opt/ohpc/pub/libs/gnu7/openblas/0.2.20/include
/opt/ohpc/pub/libs/gnu7/openblas/0.2.20/lib
```

Unload - back to being empty

```
$ module unload openblas
$ echo $CPATH $LIBRARY_PATH
```

Possible registration of ohpc packages

- While at MUG had a brief discussion about interaction with other community build initiatives (e.g. EasyBuild and spack)
- All are self-managing dependencies, but we could register some ohpc packages (e.g. with spack)
 - https://spack.readthedocs.io/en/latest/build_settings.html#build-settings
 - define ohpc package paths in packages.yaml file

Possible registration of ohpc packages

- Quick example building the libconfig package in spack
 - out of the box, builds all of autotools (which needs perl which it also builds)
 - we can pre-register the ohpc variants via paths (in theory, also via modules)

```
packages:  
  autoconf:  
    paths:  
      autoconf@2.69 arch=linux-centos7-x86_64: /opt/ohpc/pub/utils/autotools  
      buildable: False  
  automake:  
    paths:  
      automake@1.15 arch=linux-centos7-x86_64: /opt/ohpc/pub/utils/autotools  
      buildable: False  
  libtool:  
    paths:  
      libtool@2.4.6 arch=linux-centos7-x86_64: /opt/ohpc/pub/utils/autotools/  
      buildable: False
```

- Avoid rebuild:
 - original build time: 302 secs
 - updated build time: 82 secs

Is this something we might want to consider?

- e.g. tool which scans local installation and adds to yaml file