Version 9.0 (new)

CmpE 275

Section: Introductions

Part 0: Course, Lang, Lmod

Welcome To 275

On track

- Agenda – Day Zero+
  - Introductions
  - Questions
  - Next lecture: lab bare metal
- Take away
  - Organization and goals
  - Install languages (Java, C++, Python)
  - Install tools (Lua, Bash, Lmod, Cmake, …)
  - Install Pkgs (gRPC, Protobuf, Netty, …)

John Gash
john.gash@sjsu.edu

~~I'm on campus one-ish days a week:~~

Monday (275 – Online), and by appointment (Slack, Zoom)
(My office hours start after class and conclude when we are done)

How to find me

1. In person – around campus, after class
2. Email (spotty)
3. Department mailbox (graveyard)

Class Online

Zoom

- Password through Instructor
- Recorded, made available to the class

Slack

- Invites made available for next week

What to install for lectures

(Refer to the greensheet for dates)

- Java – And supporting tools (Maven, …)
- C++11 – GNU, Clang, MSVS – And Boost, OpenMPI, Cmake, …
- Python – 2.7.x or 3.x

- Scripting – Lua, Bash or something equiv.
- Tools/Pkgs – Lmod, gRPC, Protobuf

- Editors – IDE and an editor
  (vim/nano, Atom, Sublime)

(Expect additional tools/packages)

*If you are not familiar with **Java (v11,13), C++, Python (2, 3)**, it is your responsibility to spend additional time to learn.*

Grading (from greensheet)

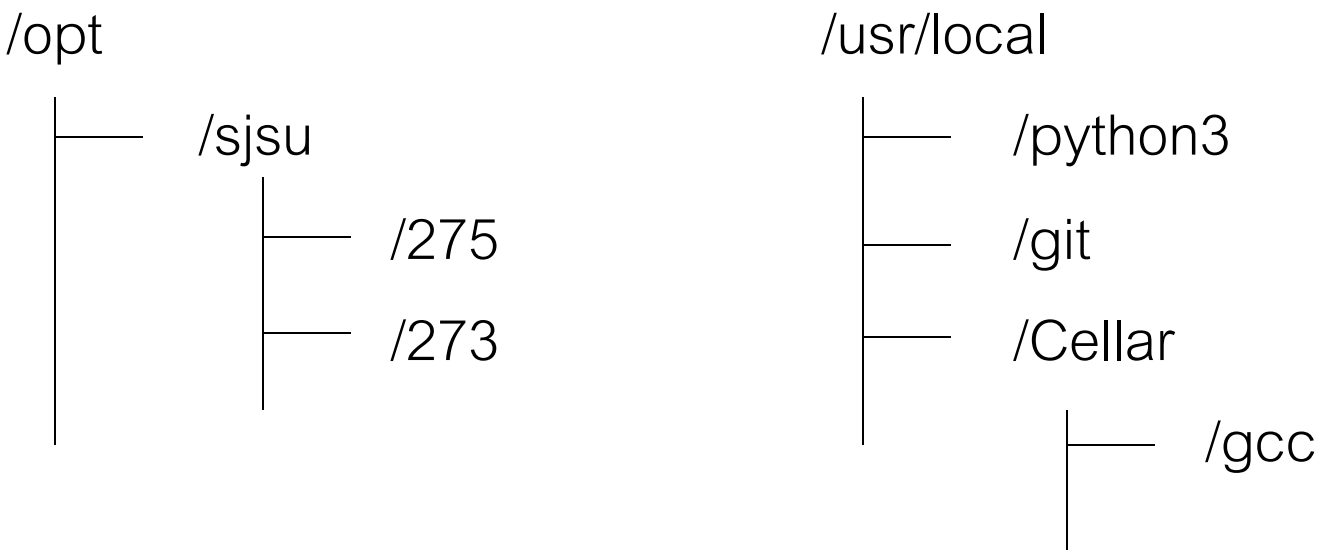
Reverse Lectures (3 instances)	30 pts
• 10 points x 3	
• 2-3 person teams	
• Deep dives	
Individual Research	10 pts
Final Exam & Review	20 pts
	5 pts

Organize content on your [computer] to support multiple versions of a tool/package

- Course software requirements
- How many versions do you have?
  - Databases: MySQL, Postgres, Mongo, Redis
  - Tools: Git, Maven 2,3,n
  - Languages: Java, Lua, C++
    - Python versions 2.x and 3.x
    - C++ compilers (GNU, Clang, MS, etc.)

Organizing with /opt or /usr/local

- Download source code or use an installer (e.g., homebrew, apt-get, gem)
- Place content in an easily located directory
  - No spaces in a path name! → ~~C:/Program Files~~



Environment and configuration help

- We will work with many software packages and languages. There are several tools that can help manage packages; two examples are:
  - Operating System
    - Linux-ish is best. However, MS is possible
  - Dotkits
    - https://github.com/chaos/dpkg-dotkit/tree/master/dotkit
  - Lmod (preferred)
    - https://github.com/TACC/Lmod
    - LUA-based configuration management

Lmod

- Environment module management
  - Ref: http://modules.sourceforge.net/man/modulefile.html
- Dynamically change configurations of packages/tools
- Lua-based (lua 5.1 + libraries)
- Documentation
  - https://lmod.readthedocs.io

Module Quick Reference

Command	Description
module avail	List available modules
module load *package*	Load a selected module
module list	Show modules currently loaded
module unload *package*	Unload a previously loaded module
module purge	Unload all loaded modules
module reset	Reset loaded modules to system defaults
module update	Reload all currently loaded modules
module display *package*	Display the contents of a selected module
module spider	List all modules (not just available ones)
module keyword *key*	Search for available modules by keyword
module module help	Display module help

Ref: https://computing.llnl.gov/tutorials/sierra/

Shell (bash or csh) configuration

- For just you
  - Bash: edit ~/.bashrc
  - Csh: edit ~/.cshrc
- To enable lmod for all users
  - Mac OS X: add to /etc/bashrc
  - Linux: add a file to /etc/profile.d

- Bashrc vs profile
  - Bashrc is for interactive shells
  - Profile or noninteractive (note some OSs will invoke both for interactive shells)

To configure lmod in terminals

Append this text to your ~/.bashrc or /etc/bashrc

```
if [ -f /usr/local/lmod/lmod/init/bash ] ; then
    export LMOD_ROOT=/usr/local/lmod/7.8
    export MODULEPATH=/usr/local/lmod/modulefiles
    export LMOD_COLORIZE=yes
    . /usr/local/lmod/lmod/init/bash
fi
```

Notes:

1. *Change the installation paths according to your computer*
2. *You may have to reboot for changes, or source your bashrc or /etc/bashrc*

Sidenote: /etc/profile or your own ~/profile.d

- Create dir: /etc/profile.d
  - Add scripts (chmod for exe)
- In /etc/profile or your (~) file

```
# execute env configuration setup
for i in /etc/profile.d/*.sh; do
    . $i >/dev/null 2>&1
done
unset i
```

Note: Change the installation paths accordingly'

Example of an environment using lmod (modules)

miso-2:~ gash$ module avail

```
---------------- /usr/local/lmod/modulefiles ----------------
   gcc/6.1                          java/maven.3.2.3
   gcc/6.5                          java/protobuf-2.6.1
   gcc/8.2          (D)             java/protobuf-3.4.1
   java/ant                         java/protobuf-3.6.1
   java/grpc                        lua/5.1                  (D)
   java/jdk/11.0.2                  lua/5.3.5
   java/jdk/13.0    (D)             ruby/2.5.3_1
```

Using lmod to switch to a new C/C++ compiler

```
miso-2:~ gash$ module add gcc/8.2
miso-2:~ gash$ module list

Currently Loaded Modules:
  1) gcc/8.2

miso-2:~ gash$ which gcc
/usr/local/Cellar/gcc/8.2.0/bin/gcc
miso-2:~ gash$ module unload gcc/8.2
miso-2:~ gash$ which gcc
/usr/bin/gcc
```

We verify which gcc is on the $PATH

Lmod uses lua to code configuration files ([base]/modulefiles/gcc/8.2.lua)

```
help([[
gcc/g++/gfortran version 8.2. Contains OpenMPI 3.0.1_1
]])

whatis("Version: 8.2")
whatis("Keywords: Compiler, gcc, g++", "mpi")
whatis("Description: GNU compiler 8.2 w/ MPI supporting
C++ 11 and some 17")

local base pathJoin("/usr/local/Cellar/gcc","8.2.0")

setenv( "CC", pathJoin(base,"bin/gcc-8"))
setenv( "CXX", pathJoin(base,"bin/g++-8"))
prepend_path( "PATH", pathJoin(base,"bin"))
prepend_path( "LD_LIBRARY_PATH",pathJoin(base,"lib"))
prepend_path( "MANPATH",pathJoin(base,"share"))
```

Reverse Lectures
(Short overview, more later)

- Small team presentation – 2 or 3 people in a team
- Focused investigation that is validated with testing and measuring
- Opportunity to deeply research and investigate a specific (bound) concept
- It is not:
  - A survey, a comparison (unless relevant to performance)
  - How to use a tool
  - Kitchen-sink discussion
  - Overview

To consider

- Introductions and focus/overview should be one slide. Don't spent too much time and multiple slides, You will run out of time!
- Demonstrations can take up time when transitioning. Be careful that showing examples are clear and reinforcing your key points.
- When presenting data (tabular or graphically) it is clear what is being shown. Label appropriately, and state conclusion.
- Practice presenting to help reduce worries and build confidence. This will help you to ensure your presentation fits within the time limit.
- Material/Subject selection involves digging into a topic that may require coding, reviewing internal code, and literature searches. Please ensure that you allocate enough time for your investigation.
- A high level summary is not acceptable. Again topics are not surveys, summaries, or tutorials of a product, library, or concept. They are instead focused, deeply investigated, and often narrow in breath. How much can one learn about 'X'?
- **Use one computer to host all class presentations to save transition time**