# Linux, Command Line and HPC Environment (Newcomer)

## Monday, June 12

Directed to new users, those with no experience using the command line. If words like ssh, cp, ls, emacs, vi, qsub, pipes, scp and globus are new to you, this class is for you. The topics to be covered are:

* Logging into remote systems
* Top 10 commands to learn
* Text Editors
* Working with a HPC cluster
* Managing inputs and outputs
* Transferring files between systems

At the end of the day you should feel comfortable entering a remote Linux machine, exploring your files and directories, executing basic commands on the terminal, submitting jobs on the cluster and transferring your data back to your computer.

# Scientific Workflows (Building, HPC Running, and post-processing)

## Tuesday, June 13

Second day is devoted to current HPC users. Focus is into improve skills to do a more efficient job, compiling software needed for a research, selecting the right queue and setup for a given job, do some scripting to automatize repetitive tasks and managing efficiently the outcome of many calculations.

* Building/installing software
* Using Python packages (pip and virtualenv)
* Basic Scripting
* Managing the 3 HPC variables (cores, memory and time)
* Job submission, monitoring, debugging and optimization.
* Chaining Jobs with dependencies
* Executing many jobs at one, job arrays.
* Parallel jobs (MPI)

At the end of this class is expected a productivity gain on the use of HPC resources. More time focus on your research rather than executing and feeding jobs on the HPC cluster.

# A glimpse on advanced topics

## Wednesday, June 14

Fast track on topics not covered on the previous two days. Small examples to get an idea about how some tools are used in Research Computing. We cover very fast topics of interest to advanced users. From casual scripters to programmers, to current developers. These topics will be covered with just one or two examples that offer a simple but effective first experience on more advanced topics.

* Bash/Python Scripting
* Programming in C, Fortran and Python
* Version Control with Github
* Optimization, Profiling and Debugging
* Parallel programming with MPI and OpenMP
* Creating Python Modules
* Test Driven Development
* Continuous Integration

# Open Lab

## Thursday, June 16

Open lab day where students can bring up topics to discuss/review and to go over any challenges they currently are having regarding computational science