Linux Lab

July 30, 2012

This exercise is meant to familiarize you with commands in Linux in order to be more comfortable using Duke's computational resources. Things you will need:

- laptop (or work with a group member who has a laptop)

- access to Duke's computational cluster

- matlab file(s) loaded in your directory on the cluster

1   Linux basics

1.1   Navigation and File Commands

1. View current location.

   (a) Type `pwd` at the command line. The system should return the following `/afs/econ.duke.edu/home/m/mynetID` where `m` is the first letter of your netID and `mynetID` is your netID.

2. Change directories.

   (a) Move down in the file tree by using the command `cd DirectoryName`. For example: type `cd Matlab` if you have a subdirectory called `Matlab` in your home directory.
   (b) Move up in the file tree by typing `cd ..` (two periods).
   (c) Move to your home directory by typing "`cd` ".

3. List directory contents.

   (a) To view the contents of a directory, type `ls`. Files appear in black and directories in blue.

4. Create a directory.

   (a) Change to your home directory. Create a directory called Matlab_Module by typing `mkdir Matlab_Module`.
   (b) Change to this directory and create a directory within it called `Linux_Lab`.
   (c) Change to this directory.

5. Create a file.

   (a) Create a blank m-file in the directory `/afs/econ.duke.edu/home/m/mynetID/Matlab_Module/Linux_Lab`. This is done by typing `touch linuxlab.m`.

6. Rename a file.

   (a) Rename your newly created file `linuxlab1.m`. This is done by typing `mv linuxlab.m linuxlab1.m`. The general syntax is `mv old_file_name new_file_name`. The mv stands for "move."

7. Copy a file.

   (a) Copy the file `linuxlab1.m` to a new filename in your directory called `linuxlab2.m`. To do this, type `cp -f linuxlab1.m linuxlab2.m`.

8. Copy a directory.

   (a) Copy the directory `Linux_Lab` to a new directory called `Linux_Lab1`. To do this, first move up the directory tree by typing `cd ..`. Then type `cp -r Linux_Lab Linux_Lab1`. Type `ls` and make sure that you have two directories in this folder now.

9. Remove a file.

   (a) Go into the new directory `Linux_Lab1`. In this directory, delete the file `linuxlab2.m` by typing `rm linuxlab2.m`. Double check that this command was executed by typing `ls` and verifying that `linuxlab2.m` doesn't exist.

   (b) NOTE: Be very careful when using the `rm` command. This deletes the file forever. It is very powerful and should be used with caution. Do not ever use wildcards when specifying files to delete.

10. Remove a directory.

   (a) Go back to the directory Matlab_Module and delete the directory you just copied (Linux_Lab1) by typing `rm -r Linux_Lab1`.

   (b) NOTE: Be even more careful when using the `rm` command to delete directories. This deletes the directory forever!

   (c) Verify that the action was done by typing `ls`. You should only have one directory now, `Linux_Lab`.

1.2 System Info

1. Check who's online right now by typing `w`

2. Check which server you're logged into by typing `uname -a`

3. Check who you're logged in as by typing `whoami`

4. Check the credentials of a user by typing `finger [netID]`, e.g. `finger tmr17`

2 Computational Jobs

Before attempting this section, please move one or more of your problem set files into a directory on the Duke Econ cluster. It can be Problem Set 2.1, or one of the problem sets from the Matlab 1 module.

2.1 Submitting a job

1. Change directories to the folder where your m-file and function files reside.

2. Submit the m-file for batch computation using the `matsub` command, e.g. `matsub ps3main.m ps3main.log`

3. Now download the example shell script from Sakai (Resources/Matlab2 Materials/Examples/example.sh) and copy it into the same folder where you did your `matsub` from.

4. Edit the example shell script to have a new name, a different email address, and the same filenames as your `matsub` command.

5. Now submit your shell script with the command `qsub -q all.q myshell.sh`

2.2 Checking job status

1. After submitting your job, check its status by typing `qstat`

2. Before submitting a job, look at the status of the cluster by typing `qstat -u ''*''`

3. Check which node your job is running on by typing `qstat -f`

4. Check the status of all nodes with `qstat -u ''*'' -f`

5. Check the detailed status of a particular job with `qstat -j [jobID]`

2.3 Deleting jobs

1. Delete a job that is still running with `qdel [jobID]`

3 Logout

Type `exit` or `ctrl+d`