

# LEARNING BATCH COMPUTING ON SOONER

Henry Neeman, Director  
OU Supercomputing Center for Education & Research (OSCER)  
University of Oklahoma

1. Log in to `sooner.oscer.ou.edu`, using Secure Shell (`ssh`) and the username and password that you've been given. Details for how to do this can be found at:

[http://www.oscer.ou.edu/ssh\\_install.php](http://www.oscer.ou.edu/ssh_install.php)

If the PC you're using has **Windows** as its operating system, then look for the description of how to download, install, run and configure PuTTY.

If the PC you're using has **MacOS or Linux** as its operating system, then bring up a terminal window and at the Unix prompt type the following command:

```
ssh -C -X -Y yourusername@sooner.oscer.ou.edu
```

except that `yourusername` will be replaced by your username.

2. If this is your **FIRST TIME** logging in to Sooner:

- (a) Please **IMMEDIATELY** change your password. Details for how to do this can be found at:

[http://www.oscer.ou.edu/password\\_change.php](http://www.oscer.ou.edu/password_change.php)

- (b) Please **IMMEDIATELY** do the following:

```
echo youremailaddress@yourinstitution.edu > ~/.forward
```

## NOTES

- i. You should replace `youremailaddress@yourinstitution.edu` with your e-mail address.
  - ii. After your e-mail address comes a blank space, then a greater than symbol, then a blank space, then tilde slash period **forward**, with no spaces between them.
- (c) Please **IMMEDIATELY** send an e-mail to `support@oscer.ou.edu` with your e-mail address and your username, so that we can update our username database.
  - (d) Copy the directory named `NCSI2010` from Henry's home directory:

```
cp -r ~hneeman/NCSI2010 ~
```

Note the tilde (`~`) at the end of this command. It means "my home directory" and is **EXTREMELY IMPORTANT**.

3. Check to make sure that you're in your home directory:

**pwd**

This is short for "print working directory." The output should be something like this:

```
/home/yourusername
```

where `yourusername` will be replaced with your user name.

You should now have your own copy of the `NCSI2010` directory, as a subdirectory of your home directory. Check to make sure that you do:

**ls**

Note that this command is lower case L followed by lower case S (that is, "ell ess" which is short for "list"), **NOT** "one ess."

You should see a list of files and subdirectories, one of which should be `NCSI2010`.

4. Change directory into your `NCSI2010` directory, like this:

**cd NCSI2010**

5. Make sure that you're in your `NCSI2010` directory, like this:

**pwd**

6. Make sure that you have multiple subdirectories: `AreaUnderCurve`, `NBody` and `GameOfLife`, like this:

**ls**

This command is lower case L, lower case S (that is, "ell ess"), meaning "list the contents of this directory."

7. Now change directory into your `AreaUnderCurve` subdirectory, like so:

**cd AreaUnderCurve**

8. Using the appropriate commands from above, make sure that you're in your `AreaUnderCurve` subdirectory, and that it has two subdirectories named `C` and `Fortran90`.

9. Now change directory into your `C` subdirectory or your `Fortran90` subdirectory, like so:

**cd C**

10. Using the appropriate commands from above, make sure that you're in the appropriate subdirectory, and that it has a directory named `Serial` in it.

11. Make your executable. The command to compile will be:

**make**

12. Using your preferred Unix text editor (for example, nano, pico, vi, emacs), edit the batch script file `area_under_curve.bsub`.

In particular:

- (a) Change `yourusername` to your user name.
- (b) Change `youremailaddress@yourinstitution.edu` to your e-mail address.

13. While you're editing the batch script file, carefully read its contents.

Notice that the batch script file has many comments, (lines that begin with a pound sign # but **NOT** with #BSUB), which describe how to submit jobs to the batch queue, how to monitor jobs in the batch queue, and, if necessary, how to kill jobs in the batch queue.

**NOTE:** You are **ABSOLUTELY FORBIDDEN** to run programs interactively; instead, you **MUST** run **ALL** programs in the batch queue.

14. Once you're done editing and reading, submit the `area_under_curve.bsub` batch script, using the `bsub` command as described in the batch script file you just read.

15. Check the status of your batch job using the `bjobs` command, as described in the batch script file. You may need to check its status repeatedly.

16. You can check the contents of standard output (stdout) and standard error (stderr) using the `bpeek` command:

```
bpeek JOBID
```

replacing `JOBID` with the appropriate batch job identifier that you saw from the `bjobs` command. You can run the `bpeek` and/or `bjobs` commands repeatedly until your job is no longer listed.

17. Once the batch job finishes (which you'll know because it no longer shows up when you do the `bjobs` command), find out which files have been created most recently in your current working directory:

```
ls -ltr
```

This command is lower case L, lower case S, space, hyphen, lower case L, lower case T, lower case R (that is, "ell ess space hyphen ell tee are"), meaning "list with long listing (lots of information), with files listed ordered by time, with the most recent at the bottom."

18. Examine the contents of the `stdout` and `stderr` files created by the batch job. Is the output what you expected? Why or why not?

19. You're welcome to play with the other example programs in the other subdirectories. You can go up a level in the directory tree by using this command:

```
cd ..
```

Notice that this command is lower case C, lower case D (for "change directory"), followed by a space, followed by two periods ("dot dot") with no spaces between them, where "dot dot" means "the parent of the current working directory I'm in."

20. Once you've done this, you should check what directory you're in, and what files and subdirectories are in them, using the appropriate commands described above.
21. Why shouldn't you edit these files on your Windows PC, and then upload them to Sooner?

In principle you can, but in practice it's a **TERRIBLE IDEA**.

Here's why:

In Windows, almost all text editors embed hidden special characters (for formatting and so on) in the text file that you're editing, and also they express carriage returns differently than in Unix (including Linux).

Furthermore, in some of these files (especially batch script files and makefiles), **where** the carriage returns occur is **EXTREMELY IMPORTANT** (and in the case of makefiles, so too with tabs as opposed to spaces), and the text editors in Windows cannot be relied on to keep those in the proper places.

So, if you edit a file in Windows, there's no guarantee that, when you upload it to a Unix machine (including Linux), it'll be usable.

There's a command that can fix some (but not all) of these problems:

```
dos2unix filename
```

This is a very handy command, but it can't fix everything, so you take **HUGE RISKS** if you choose to edit a file in Windows and then try to use it in Unix.