

Basic Linux Tutorial

CS390/590 Fall 2014

This tutorial goes over the very basics of using a Linux or generally a Unix terminal. All these commands can be found at <http://ss64.com/bash/> or by use of the 'man' command from the terminal (see help below). Single quotes are used to highlight a command in a sentence they are not actually typed in the terminal.

General Information

Command line: The Linux terminal is often referred to as a command line interface (CLI). This is the same for Windows, however windows uses a completely different language and does not unto itself support SSH (a protocol for secure transmission of data) and therefore cannot be used with these instruction or by itself even to login to UNM (though UNM does support ftp, this does not allow one to run Matlab remotely). CLI and terminal are used interchangeably for Linux.

File Extensions: In cases where you are not executing a file, you will need to include the file extension as part of the file name, or Linux will not find the file.

Arrow Keys: Up/Down keys select previously used commands

Control Tab: Starting a word then pressing control tab will complete the word if only 1 option is available at that point.

Command Arguments: Many commands like 'ls' support arguments to change their functionality. These are done by adding a '-' and then the arguments with no spaces between any. Note, press enter after typing in the command to execute it.

For example adding the argument for hidden files 'a' and directory and file information 'l' to ls is done like this;

```
ls -al
```

Copy/Paste with CLI: Oddly enough the keyboard shortcuts do not work for copy/paste function in the terminal. Use the mouse to select the area to copy and right click and select copy. Likewise for paste.

Help: Help in linux is 'man', short for manual. From the terminal simply type 'man' without the quotes followed by a space and then the name of the command you want help about.

For example, to find information about the 'ls' command type the following in a terminal;

```
man ls
```

Directory (Folder) Information

Folders on Linux are called directories. Note that Linux uses spaces to delimit between words, if a directory name contains a space, surround the name in double quotes to access it.

Note that ./ refers to the current directory you are in and ~/ refers to the directory path that you are currently on.

Creating Directories: To create a directory type 'mkdir <directory name>'. To create the directory mfile type

```
mkdir mfile
```

Changing Directories: To change a directory, use the 'cd' command followed by the method you want to use to change the directory as shown below;

- cd directory1/directory2/directory3/... :move to the directory listed. The directory must be either in the path you currently on or you must start from your home directory. So if dir2 is not in your path but is just under the home directory you may type 'cd ~/dir2' to go directly to that directory.
- cd / :move to top level directory (generally this is above the home directory)
- cd ~ :move to home directory
- cd .. :move up one directory

Looking at information about directories and files: This uses the 'ls' command (list I believe).

- ls : This will list all files and directories alphabetically in the terminal.
- -a: This argument shows all hidden files (these start with a .). To use simply add the argument to the ls command like so;

```
ls -a
```

- -l: This argument shows all files and directories alphabetically with their type and permissions. To use simply add the -l to the ls command like so;

```
ls -l
```

Understanding the ls -l output.

This can be a bit confusing at first. There is a line of dashes, rs, ws and xs, 10 total that tell about the permissions on the file. The first letter shows the whether it is a file or directory, the remaining are grouped into 3 groups of 3 (user, group and other) which tell what each type can do with the directory or file. The creator of the file is the user and is represented on the system by the user name (3rd column). The group mentioned on the 4th column and everyone else is other. This is further explained below.

1st column (special note that this is a default setting it can be re-arranged). This columns looks like a 10 of -s,rs,ws, and xs (example : -rwx-wxr--). They have the following meaning;

-: file (1st character) or no functionality (remaining character)
d: directory (only 1st character)
r: read
w: write
x: execute

As mentioned above they are broken into the following order

file or directory|user rwx|group rwx|other rwx

-|rwx|-wx|r-- actually looks like -rwx-wxr-- <file or directory name>

Changing Permissions

You may change permission on any file or directory you have access too, though only up to your level (in general, don't mess with permissions unless you are the owner (user) of the file. To do this you use the command 'chmod' followed by the whose permissions you are changing, what you are changing them too and the name of the file or directory you are changing. It is quite common to have upload a file and have the settings be different than what you want.

To identify who you are changing use the following characters;

- u: user (first group of 3—generally you)
- g: group (second group of 3)
- o: other (third group of 3)
- a: all (everyone)

To identify what you are going to do use the following characters followed by the function you want to add/subtract (r,w,x);

- +: add function
- -: subtract function

So all combined to change the file named mfile from -rwxrwx--- to -rwxr--r—which removes the permissions from the group for writing and executing the file and give 'other' the ability to read

the file use to steps one for each add/subtract (don't try to add and subtract at the same time, it is possible but the method is more complicated to figure out do man chmod or google to see).

```
chmod g-wx mfile  
chmod o+r mfile
```

Additionally there are arguments chmod can take, one of which is -R. This is a recursive command which just means that it will change the directory you are in and all directories and files it contains. Its argument goes after chmod but before what you are changing as follows;

```
chmod -R o-r mydirectory
```

This removes read permission from mydirectory and all files and directories it contains.

Copying a file/directory locally:

Note that file and directory can be used interchangeably here.

Use 'cp' command. To copy one file to another type 'cp' then the file you are copying and what you are copying too. Note that you can use file paths for both the file you are copying and where you are copying too.

For example: If you have a file name helloworld.m and want to copy it to the same directory but rename it hw.m type;

```
cp helloworld.m hw.m
```

To copy it to the directory nextDir which is in your current directory type;

```
cp helloworld.m ./nextDir/hw.m
```

Deleting files or directories:

Note that these are two different commands, deleting a directory results in all files being deleted as well. Linux calls deleting files removing.

For files use 'rm' followed by the file (as usual you may specify the path from the directory you are in if the file is not in your current directory). For example to delete hw.m type;

```
rm hw.m
```

For directories use the command 'rmdir' followed by the directory. For example to delete nextDir type;

```
rmdir nextDir
```

Remote Access:

Remote access gives you the ability to upload, download and run your files remotely on the UNM (this tutorial assumes you are using the CS servers but applies to any Linux server) servers. This allows you to run more time consuming things on more powerful computers. It does not however excuse poor programming, so please review your code for optimizations when reasonable.

1. Login to the server

Linux uses SSH to communicate remotely. Windows does not support this, thus to login using SSH a program that supports SSH must be installed. These (are free) include but are not limited too;

- Putty: This is a simple terminal like program for remotng in, its not a full blown linux solution like the others.
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- Cigwin: This is basically Linux on your windows <https://www.cygwin.com/>
- Virtual Machines: Like virtualbox (oracle) and virtual player (vmware). These allow loading complete operating systems onto your machine generally from .iso files. This is similar to what you would do if you got a new computer with no operating system, except they exist inside the virtual machine software so you do not lose your current system.

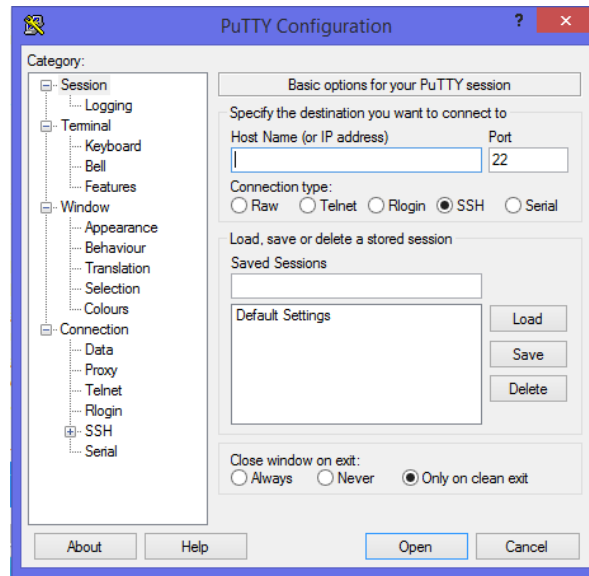
To login using a standard terminal (like you find on a mac or linux, Putty is a bit different in its current rev see below for it), you use the 'ssh' command followed by your username@<server ip>. Note that the CS department uses load balancing so the server you login into may not be the server you name, however any server will give you access to your files. It will then ask for you password (your CS password not your UNM password). After which you will be online and treat your current terminal as if you are typing on a CS computer (without graphics), it will not see files on your local computer. You may open as many terminals as you want so if you need local access just open another one.

So to login into phobos, assuming you are me (which of course won't work unless you hacked my password...I will find out...).

```
ssh wjust@phobos.cs.unm.edu
```

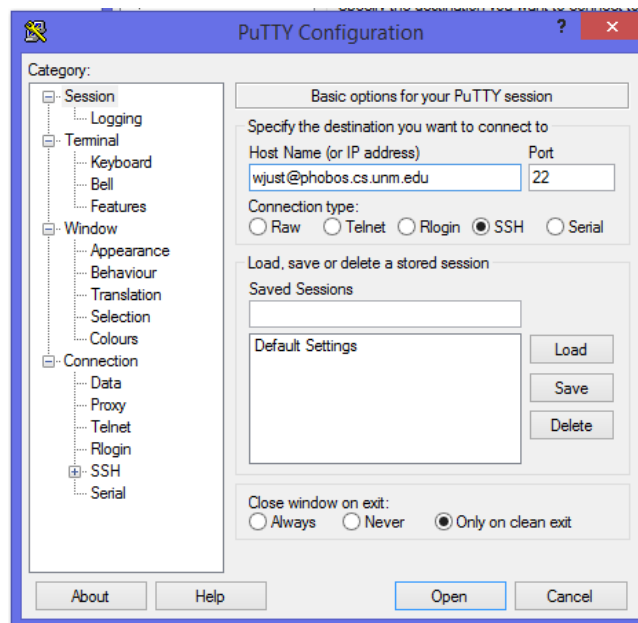
Type in password and there you are on the CS machine.

To login with Putty (make sure it is installed). Double click the Putty Icon (or otherwise start the Putty program). You get a window like below (next page);



Under 'Host Name (or IP address)' type the same thing you would type after the SSH command give above: `wjust@phobos.cs.unm.edu`

Make sure SSH is checked (right under where you just typed). The window should now look like.



Press the 'Open' button at the bottom right (left of the Cancel button) and a new window will open asking for your password. Proceed like normal from there.

2. Logout

To logout; type 'exit'.

3. Upload a file to the server.

Linux uses the 'scp' command (secure copy) to upload and download files. To upload a file from the directory you are currently in type `scp <file name> <user name>@<server ip>:<directory path>`. Note: Linux will just write over a file, it will not ask permission.

For example to upload hw.m file to the mFile directory that is just underneath my home (default) directory at UNM, type;

```
scp hw.m wjust@phobos.cs.unm.edu:~/mFile
```

Note: To do this the terminal must be set to your laptop (local) not remotely logged into UNM.

Putty is the same except you must install the pscp.exe file and you use the windows command line (not the terminal you got in login above) and type 'pscp' instead of 'scp'. Except that ~ is not necessary (and doesn't seem to be recognized, so just leave it off, it will start at your home directory). You will also need to add the path to pscp to the windows PATH variable as well.

4. Download a file from the server.

Downloading a file is done a similar fashion again using the 'scp' command. You change switch the two statements after the 'scp' command. So its `scp <server ip>:~/<file path/file name> <directory to download too>`.

For example to download the file hw.m (located in my mfile directory) to my current directory (the one the terminal says I am in) type;

```
scp wjust@phobos.cs.unm.edu:~/mfile/hewd.m ./
```

Note again your terminal must be local not logged into UNM.

Putty is the same except as mentioned in uploading the file.

5. Running Matlab

To run a program in Matlab that is on the UNM server you first login remotely as above. Then type 'matlab' (there are arguments to this but not required). You will then get a >> prompt. You can still use 'ls' and 'cd' commands to move around your directory when in the 'Matlab' mode.

Move to the directory your program is in and type the program name without the .m extension.

For example to run hw.m, enter 'Matlab' mode as above and type;

hw

Playing Nice with others:

While the servers are certainly more powerful than the laptops and can handle a very large load (lots of computation from a variety of sources), programs that are handling massive amounts of data can still bogged down the servers and if poorly handled or there are simply too many can cause a server crash. This is generally unpleasant for all. To help with this, Linux has the command 'nice'.

The command 'nice' takes the argument -n followed by a number between -20 and 19 (with a default of 0...most likely if no argument is given). That determines the priority. -20 is highest while 19 is lowest priority. Generally don't use a negative number for the priority, the default should be okay. It is then followed by the program you want to run.

In this case, the program that is going to play 'nice' is Matlab not your program (Matlab does not recognize the nice command). So to run your programs 'nice' instead of starting Matlab by typing 'matlab' type;

```
nice matlab  
nice -n 10 matlab (if for a priority of 10)
```

From the Matlab prompt you run normal.

Using 'nice' is greatly appreciated by the CS department and if you do something that is horribly resource intensive will keep you in good standing with the CS folks.