Introduction to Linux, Emacs, and g++

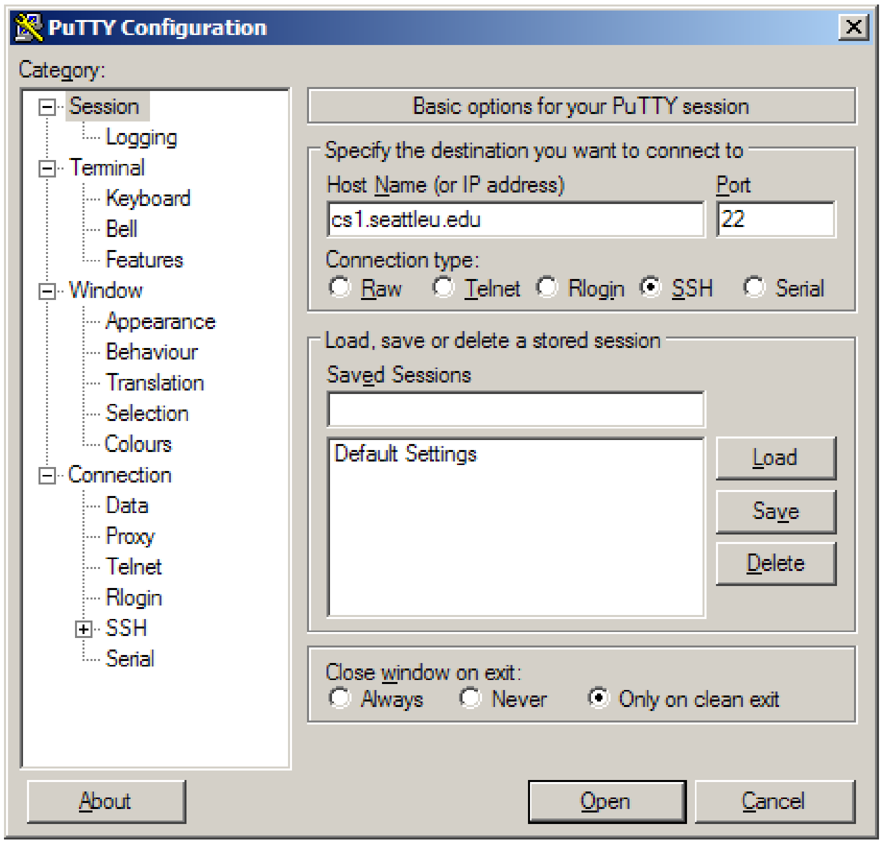
Eric Larson
Computer Science and Software Engineering
Seattle University

Accessing the Linux machine

By enrolling in a CSSE course, you will automatically receive a Linux account on the department's Linuc machine 'cs1.seattleu.edu'. The computer is administered by the computer science department and resides in the department's server room.

You will be accessing the computer remotely using an SSH (secure shell) client program called Putty. Putty is available on campus lab machines (Windows only) and can be downloaded for use off campue (see Working off Campus section). Mac users - see section on Macs later in the document.

When Putty starts, you will see a configuration screen. Type 'cs1.seattleu.edu' as the host name and click Open.

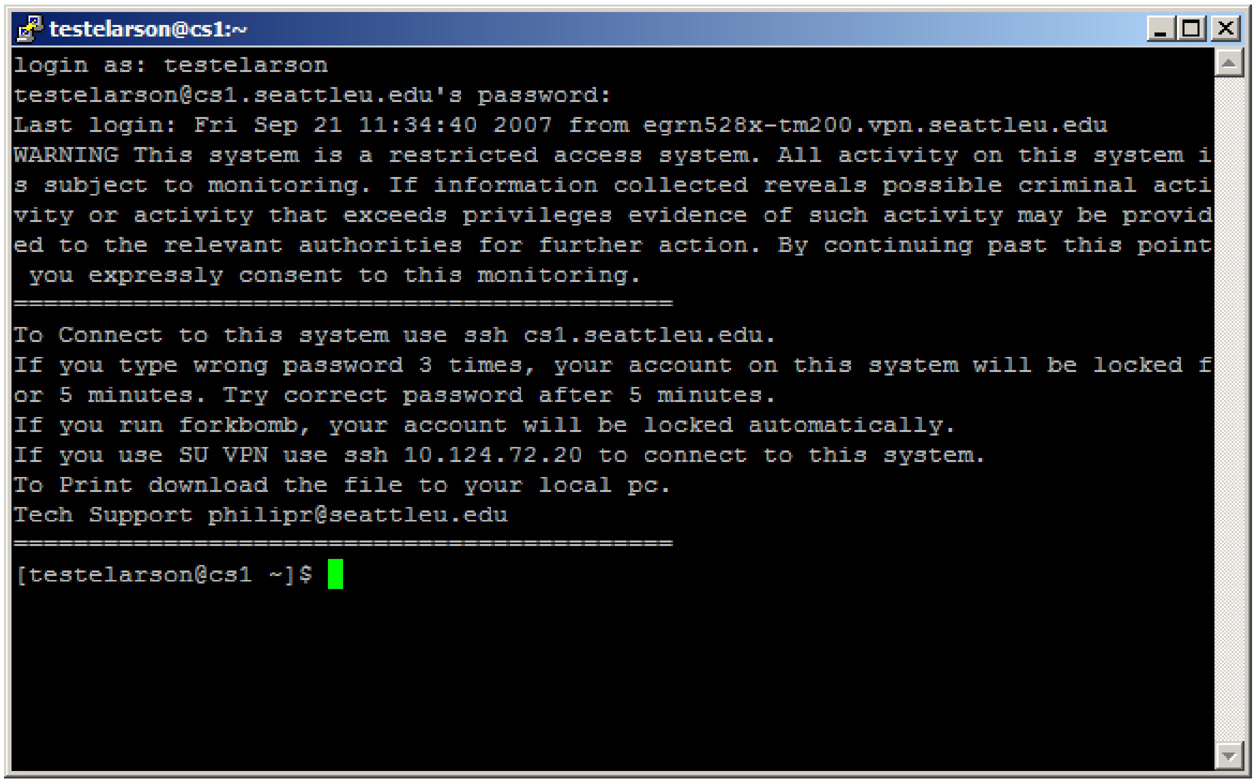


Host Name (or IP address): cs1.seattleu.edu
Port: 22
Connection type: Raw Telnet Rlogin [✓ SSH] Serial
Close window on exit: Always Never [✓ Only on clean exit]

Then you will see a prompt stating '`login as:`'. The Linux machine is asking for your username. The username is the same as the username for your Seattle University email account (the part before the '@' sign).

Then it will prompt for a password. The password is the same as your email password. Please be aware that nothing will appear on the screen when typing your password (not even asterisks or circles you typically see when typing passwords on Windows).

If you successfully log on, you will see an information followed by a prompt:



testelarson@cs1:~

```
login as: testelarson
testelarson@cs1.seattleu.edu's password:
Last login: Fri Sep 21 11:34:40 2007 from egrn528x-tm200.vpn.seattleu.edu
WARNING This system is a restricted access system. All activity on this system i
s subject to monitoring. If information collected reveals possible criminal acti
vity or activity that exceeds privileges evidence of such activity may be provid
ed to the relevant authorities for further action. By continuing past this point
 you expressly consent to this monitoring.
============================================
To Connect to this system use ssh cs1.seattleu.edu.
If you type wrong password 3 times, your account on this system will be locked f
or 5 minutes. Try correct password after 5 minutes.
If you run forkbomb, your account will be locked automatically.
If you use SU VPN use ssh 10.124.72.20 to connect to this system.
To Print download the file to your local pc.
Tech Support philipr@seattleu.edu
============================================
[testelarson@cs1 ~]$
```

At the prompt, you can either direct Linux to execute a program or one of its internal commands. We'll learn a variety of commands during this tutorial.

File and directory operations

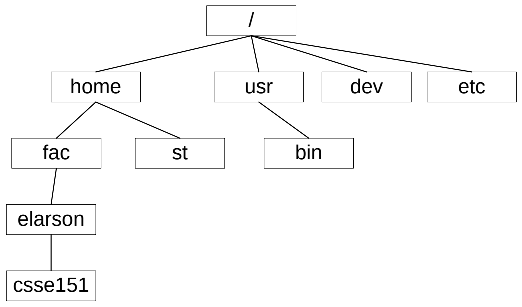
The first thing to understand that the Linux file system is a hierarchy of directories. A directory is identical to a folder in Windows. When you start, you are in your home directory. It is similar to the 'My Documents' folder in Windows XP. Each user has their own home directory. You can create subdirectories and files within your home directory. To see where your home directory is located type the command and press Enter:

pwd

It will show you an output like this (except that your username will appear instead of testelarson).

```
/home/st/testelarson
```

The top level directory notated **/** is the root directory. It is similar to accessing the top-level of the C-drive on a Windows computer. Here is a picture of the directory structure:



It is not critical that you understand or memorize this hierarchy. All you need to focus is on your home directory. How you organize your home directory is up to you. Your home directory also has a shortcut character **~**. You can see the **~** in both the prompt and on the title bar for putty. Both the prompt and title bar keep track of which directory you are in.

To list the contents of a directory, type the following command at the prompt (and press enter):

```
ls
```

Initially the directory is empty (contains no files or directories) so nothing is printed out.

Now we are going to create a directory. The command is `mkdir` which stands for make directory.

```
mkdir csse151
```

Again, nothing appears to happen. In many cases in Linux, no news is good news. Many commands do not display any feedback to the user unless the command failed for some reason.

Now, list the contents of the directory again:

```
ls
```

This time, we see the csse151 directory.

Now we are going to traverse into the csse151 directory. The command is `cd` which stands for change directory:

```
cd csse151
```

Note how the prompt and title bar changed. The prompt only displays the name of the current directory while the title bar gives the full path relative to the home directory (`~`). You can also use the pwd command to the see the full path relative to the root directory (`/`).

There are a few variations on the cd command that are important:

`cd pa1`	(local change, there must be a pa1 directory in the current directory)
`cd ..`	(go up one level to the parent directory)
`cd`	(go to your home directory)
`cd /usr/bin`	(absolute path)

In general, `".."` means the directory one level above and `"."` means the current directory. This is useful in other commands in addition to `cd`.

Using Linux for computer programming

The minimal steps for creating a program on Linux:

1. Typing the program using an editor installed on Linux called emacs. In doing so, you create a source code file. C++ source code files, by convention, have a suffix of `.cpp`.
2. Compiling the program using g++. This will create an executable file that you can run within Linux.
3. Running the program. This is done directly in Linux.

Keep in mind the process of creating a program is much more complicated than what is presented here, especially for larger programs. The purpose of this tutorial is how to learn Linux and its tools – it is not designed for teaching the entire programming process.

Editing a file in emacs

Now we are going to create a file that contains a C++ source code program using emacs. First, make sure you are in the csse151 directory we created earlier. Then, we must invoke emacs using the command:

```
emacs hello.cpp
```

where `hello.cpp` is the name of the source code file you want to create and will vary from program to program. Since `hello.cpp` does not exist, the screen will be mostly blank.

Now, start typing the program that follows. There are keyboard commands for most text editing functions. Here are the three essential keyboard commands:

Save a file:	Press Ctrl-x followed by Ctrl-s
Exit emacs:	Press Ctrl-x followed by Ctrl-c
Cancel out of a command:	Press Ctrl-g

The last command is critical if you improperly type a command as some request more input. In rare you may do something that opens up another file or splits the screen. The best bet in these cases is probably to exit emacs (saving your file in the process) and restart it.

Another common problem worth mentioning is if you accidentally press Ctrl-z (likely problem since z is so close to x on the keyboard). When this occurs, emacs exits back to Linux with a message saying it has stopped. This means emacs is suspended. The best way to explain this is that it is similar to minimizing a window on Windows. To get emacs back, you need to maximize emacs using the `fg` command (`fg` stands for foreground; suspending emacs puts emacs in the background). Simple type and press enter:

```
fg
```

You will simply resume emacs where you left off. The file is not saved or modified during this process.

There are also a myriad of other Emacs commands are available. An emacs quick reference card (available on the "Introduction to Linux" website) shows a more complete list of commands.

On the blank screen, type the following programming – maintain all formatting. Note that emacs will automatically color different parts of the code and keeps track of indenting[1].

```
// Your Name
// hello.cpp
// Prints hello world to the screen

#include <iostream>
using namespace std;

int main()
{
  cout << "Hello world" << endl;
  return 0;
}
```

When you are done, save the file and exit emacs.

Once the file has been created, you can make changes to the file using the same command line:

```
emacs hello.cpp
```

Note that you must be in the directory where hello.cpp is located in. Otherwise, you will create a new file called `hello.cpp` in whatever directory you happen to be in.

Compiling and executing a program

To *compile* the sample program, enter at your UNIX prompt:

[1] If syntax coloring does not work, execute the following command in your home directory: `cp ~elarson/.emacs ~/.`

```
g++ -Wall hello.cpp
```

There are three parts to the command line:

- `g++` is the name of the program – it is the GNU C++ compiler
- `-Wall` is a switch that tells g++ to turn on all warnings. The compiler will inform the user of likely errors in the code. It is highly recommend for beginning programmers to use this switch.
- `hello.cpp` is the name of the source code file you want to compile.

As before, no news is good news. If all you see is the prompt again, that mean your program compiled successfully without any errors or warnings. If warning or errors occur during the compile process (they will be listed on the screen), you will need to *edit* your program and make corrections.

Fixing compilation errors

Errors are indicated by line numbers. In emacs, you can jump to a line by pressing Ctrl-x followed by Ctrl-j. Then you will be given a prompt asking which line you want to go to.[2]

A few tips:

- You can run Putty multiple times. It is helpful to have two windows. One for emacs and one for compiling. That way you can see the error message and the source code at the same time.
- Sometimes the error occurs on the previous line of code. This is especially common when there is a missing semicolon on the previous line.
- If you get multiple error messages, start with the first one (smallest line number) and work your way down. Sometimes one error will cause several errors down the road.
- Error messages can be cryptic at times – especially to beginning programmers. If you do not understand an error message, please send your instructor email with the error message and the piece of code that is failing right away.

Executing a program

The executable that is created from g++ is called `a.out`. You can verify that a file of this name has been created using ls (you may also notice some backup files that emacs leaves around). To run the program, you need to specify the following command:

```
./a.out
```

The initial `./` indicates the program is in the current directory.

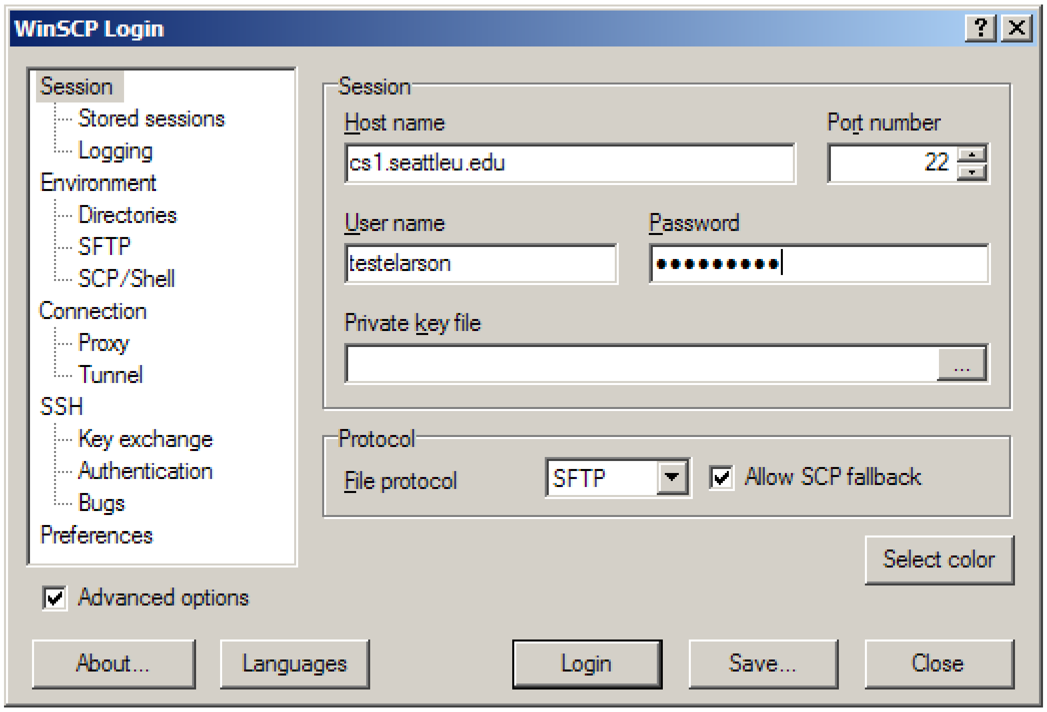
[2] This command is not a standard Emacs command. If it does not work, try executing the following command in your home directory: `cp ~elarson/.emacs ~/.`

Playing around is encouraged! Try editing the file to change the output message. Then repeat the compilation and execution steps. More complicated programs may take several rounds of editing and compilation.

Backup your programs

It is a good idea to backup your files on a thumb drive or other media occasionally as the hard drive on cs1 could fail at anytime or you may simply accidentally delete them. Your instructor, the computer science and software engineering department, and Seattle University are not responsible for any damages occurred due to lost files.

To do this, use the program WinSCP. When starting the program, click on New. You will see a form where you can enter the hostname 'cs1.seattleu.edu'. Unlike Putty, you also enter your username and password using this form. When the form is filled in, click Login.



Session

Host name: cs1.seattleu.edu

Port number: 22

User name: testelarson

Password: ●●●●●●●●●

Private key file:

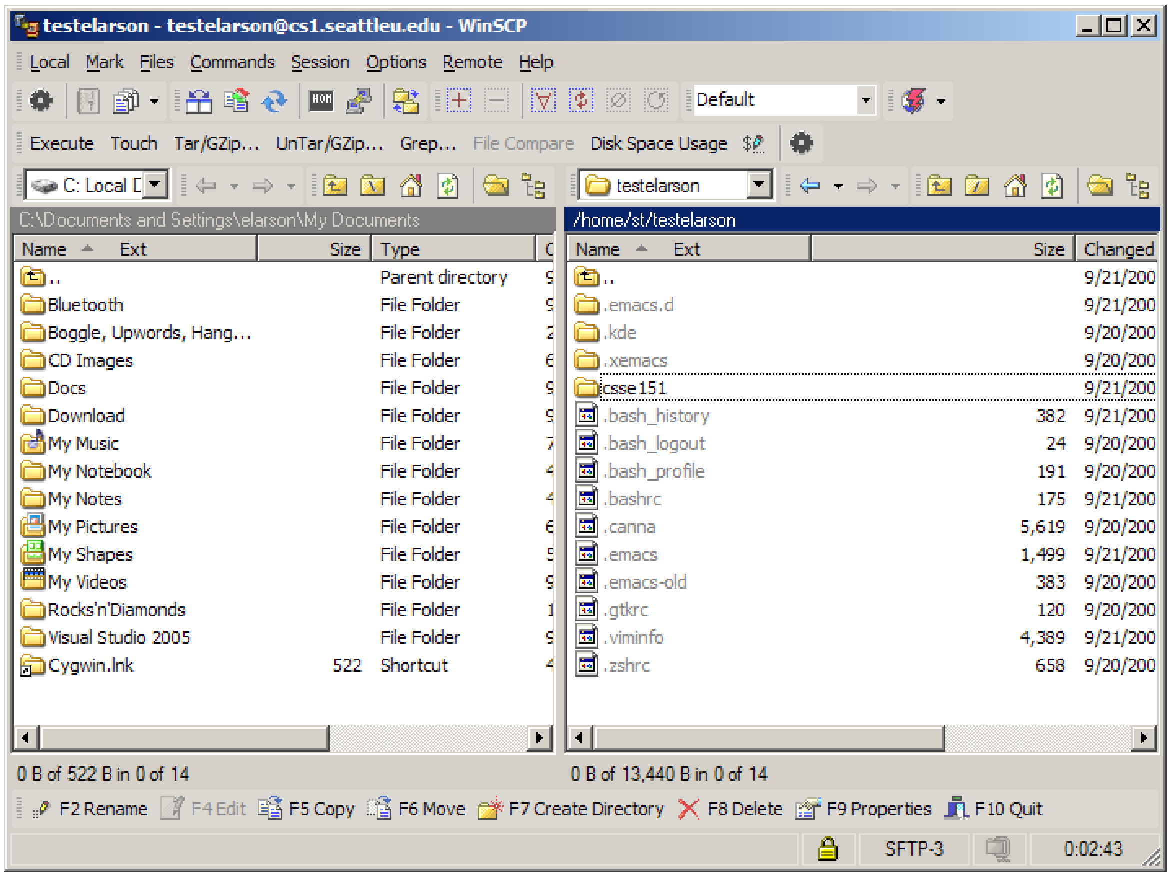
Protocol

File protocol: SFTP

[✓] Allow SCP fallback

[✓] Advanced options

Then you should see a screen like this with two panes:


testelarson - testelarson@cs1.seattleu.edu - WinSCP
Local Mark Files Commands Session Options Remote Help
Default
Execute Touch Tar/GZip... UnTar/GZip... Grep... File Compare Disk Space Usage
C: Local D
testelarson
F2 Rename F4 Edit F5 Copy F6 Move F7 Create Directory F8 Delete F9 Properties F10 Quit
SFTP-3
0:02:43

C:\Documents and Settings\elarson\My Documents

Name	Ext	Size	Type	C
..			Parent directory	9
Bluetooth			File Folder	9
Boggle, Upwords, Hang...			File Folder	2
CD Images			File Folder	6
Docs			File Folder	9
Download			File Folder	9
My Music			File Folder	7
My Notebook			File Folder	4
My Notes			File Folder	4
My Pictures			File Folder	6
My Shapes			File Folder	5
My Videos			File Folder	9
Rocks'n'Diamonds			File Folder	1
Visual Studio 2005			File Folder	9
Cygwin.lnk		522	Shortcut	4

0 B of 522 B in 0 of 14

/home/st/testelarson

Name	Ext	Size	Changed
..			9/21/200
.emacs.d			9/21/200
.kde			9/20/200
.xemacs			9/20/200
csse151			9/21/200
.bash_history		382	9/21/200
.bash_logout		24	9/20/200
.bash_profile		191	9/20/200
.bashrc		175	9/21/200
.canna		5,619	9/20/200
.emacs		1,499	9/21/200
.emacs-old		383	9/20/200
.gtkrc		120	9/20/200
.viminfo		4,389	9/21/200
.zshrc		658	9/20/200

0 B of 13,440 B in 0 of 14

The left pane refers to the directory structure of the Windows computer you are on. The right pane refers to your Linux home directory. One thing you may notice is the additional files that start with a period and are grayed out. These are hidden files that don't appear when you use `ls` (unless you use `ls -a`). For the purposes of this discussion, you can ignore these files.

On the left pane, navigate to the drive and/or directory you want to place the files. Then you can drag the file(s) that you want to copy from the right pane to the left. You can also copy directories. The easiest way to backup everything is to simply drag the csse151 directory. It will copy that directory and its contents.

Macs

For Macs, no additional software is necessary. Run Terminal under Utilities. This brings up a prompt. Here, type: "`ssh cs1.seattleu.edu`" and press Enter. You'll be asked for your username and password.

Working Off Campus

The machine cs1.seattleu.edu can be accessed from any machine with an Internet connection and an SSH client. Putty and WinSCP are free programs that can be downloaded at these sites:

Putty: http://www.chiark.greenend.org.uk/~sgtatham/putty/
WinSCP: http://winscp.net/eng/index.php

Linux commands for copying, moving, and removing files

To copy a file, use the `cp` (copy) command:

`cp` *current-file-name* *new-file-name-or-path*

The *new-file-name-or-path* can either be a file name in which the file will be copied in the current directory with the new file name or it can be a directory in which the file will be copied with the same name in the new directory or you can do both. Examples:

```
cp /home/fac/elarson/csse151/p1/test1.txt .      (copies test1.txt to the current directory)
cp test1.txt words.txt                     (copies puzzles.txt to the file words.txt)
cp test1.txt /home/elarson/words.txt       (copies puzzles.txt into new directory with new name)
```

To move or rename files, use the `mv` (move) command:

`mv` *current-file-name new-file-name-or-path*

The move command works in the same way as the copy command except that the original file is deleted. This commands allows you to move files to different directories or change the name of a file. Examples:

```
mv test1.txt ..                 (move puzzles.txt to the parent directory)
mv test1.txt words.txt          (renames puzzles.txt to words.txt)
```

To remove a directory, use the `rmdir` (remove directory) command:

`rmdir` *directory-name*

This will delete the directory. This command will only work if the directory is empty. Files can be removed from a directory using the `rm` command below.

To remove a file, use the `rm` (remove) command:

`rm` *file-name*

This will permanently delete the file. WARNING: There is no recycling bin in Linux so once you delete a file, it is gone.

Occasionally, you will want to copy, move, or delete an entire directory at once. This can be accomplished using the "`-r`" switch:

`cp -r /home/elarson/public .`	(copy the entire public directory to the current directory)
`rm -r csse152`	(delete the csse152 directory and all of its contents)

In other situations, you will want to list, copy, move, or delete a group of files with a similar name. This can be done using a wildcard '`*`'. The '`*`' will match any group of letters:

`ls *.cpp`	(list all files that end with .cpp)
`ls a*r`	(list all files that begin with a and end with r)
`cp * junk`	(copy all files into a directory called junk)
`rm *.o`	(remove all object code files)

Other useful Linux commands

`less` *filename*
Displays the contents of a text file. `more` will pause after filling up one screen.

`clear`
Clears the screen.

`diff` *file1 file2*
Compares the contents of the two text files and reports any changes.

`spell` *filename*
Check the spelling of a text file. Probably would not want to run this on a program source file.

`man` *command*
Get help on the entered command.

Note: Errors, suggestions, or improvements to this tutorial? Please send email to Dr. Larson elarson@seattleu.edu