**Basic Linux Tutorial**

This tutorial is designed to familiarize you with some basic Linux commands which are necessary to do the exercises in this course.

In Linux we will navigate the file system using the *command prompt* at which we will type commands. To get a *Terminal* window in which to type commands, right click and then click Open Terminal.

First, to find out which directory you are currently in (the *present working directory*) enter `pwd`. Subdirectories are separated by slashes. We call the first / directory the *root* of the directory tree.

To see the contents of the current directory, enter `ls`. This lists all files in the directory.

You have the ability to make new directories with the command `mkdir`. For example, make a directory by entering:

```
mkdir BasicLinuxTutorial
```

Now if you enter `ls` again you will see that your new directory is included in the present directory.

You can change the present working directory to this new directory by entering

```
cd BasicLinuxTutorial
```

Now the command `pwd` will show that you are in this new directory and `ls` will show no files, because this directory you created is initially empty.

To create a new text file named firstfile.txt, use the command

```
gedit firstfile.txt
```

Type several lines of text into the window, then click file->save followed by file->quit.

If the graphical text editor such as `gedit` is not available, the new file can also be created using the command:

```
cat > firstfile.txt
```

Type several lines of text into the window.
Press 'ctrl + d' to return to Terminal window.

We can easily display the contents of the file at the command prompt with the command

```
more firstfile.txt
```

or by using `cat` command

```
cat firstfile.txt
```

Alternatively, it might be more convenient to display only the last line of the file. This can be useful when we are tracking a calculation in progress by looking at the last several lines of the output file. To do this, use the command

```
tail -n 1 firstfile.txt
```

You can see any number of lines by changing "1" to a different number. Now we can make a copy of firstfile.txt named secondfile.txt using the command

```
cp firstfile.txt secondfile.txt
```

Entering `ls` will show that two files are now present in the directory. To edit the new file, enter

```
gedit secondfile.txt
```

modify one line of text, leaving the rest the same, and then save and exit as you did before. To see a list of differences between the two files use

```
diff firstfile.txt secondfile.txt
```

Now create a new subdirectory called new with the command

```
mkdir new
```

and move the new file to that directory by entering

```
mv secondfile.txt new/
```

To confirm that the file is there without changing directories, use

```
ls new
```

to list that directory. Note that the *path* "new" is *relative* to the present working directory. We can list this directory in other ways as well. For example, we can use an *absolute path* with

```
ls /home/<your username>/BasicLinuxTutorial/new
```

where the first "/" indicates that the path is absolute in that it is relative to the root of the file tree. Alternatively, you can use "~" to indicate that a path is relative to your home directory

```
ls ~/BasicLinuxTutorial/new
```

or ".." to indicate the directory immediately above the present working directory

```
ls ../BasicLinuxTutorial/new
```

The "/", "..", and "~" symbols are not unique to `ls`. They work anytime you are specifying a path in Linux. For example, you can copy secondfile.txt from the new directory to your home directory with.

```
cp new/secondfile.txt ~/
```

and then edit it with

```
gedit ../secondfile.txt
```

Now remove the file you just created with

```
rm ../secondfile.txt
```

Finally, sometimes it is convenient to copy a directory and all of its contents. For example, copy the entire directory "new" to a directory called "newer" with the command

```
cp -R new newer
```

**Two Useful Tricks**

One trick which can save you a lot of typing and help you to remember your directory names is this: Start typing "`ls ~/Basic`" and then rather than finishing typing the directory name (BasicLinuxTutorial), hit the Tab button. Tab automatically finishes the directory or file name if there is only one directory or file which starts with those characters. If there is more than one directory starting with these characters, hitting Tab once will fill in as many characters as possible, and then hitting it two times will show a list of all directories and files which start with this string of characters. (Note that the behavior when you hit the Tab button is specific to certain Linux *shells* and will not be the same on all systems.)

Another useful trick is that you can scroll back through the most recent commands you've entered at the command prompt by pressing the up arrow. You can then run the exact same command again by hitting Enter, or edit the line and hit Enter to run a similar command.

**Summary**

| | |
|---|---|
| `ls` | list the contents of a directory |
| `cd` | change directory |
| `cp` | copy a file or directory (must use -R to copy a directory and all of its contents) |
| `mv` | move or rename a file or directory |
| `rm` | remove a file (must use -R to remove a directory and all of its contents) |
| `mkdir` | make a directory |
| `pwd` | display the present working directory |
| `gedit` | edit a text file (other popular options included `emacs` and `vi`) |
| `more` | display a text file |
| `tail` | display the last several lines of a text file |
| `diff` | display the differences between two text files |

| | |
|---|---|
| `~` | the home directory |
| `/` | the root of the file system |
| `..` | one directory above the present |