ATM 350
A Condensed UNIX Guide (refer to UNIX Tutorial link for more thorough information!)

Logging on

Windows: Xming* + PuTTy or SecureSSH (on Desktop) → Hostname: ash.atmos.albany.edu or reed.atmos.albany.edu

Mac: XQuartz* + Open terminal (Go-->Utilities-->terminal) and type

```
ssh username@ash.atmos.albany.edu -Y
```

Linux: Open terminal (right click on desktop)

Should you wish to log in to ash, type `ssh username@ash.atmos.albany.edu -Y`

(* necessary if you will be running any Linux programs that open a window (e.g., GEMPAK, gedit)

Manual pages

```
man command_name   (lists a manual of how to use given command name)
      example:  man ls   (shows manual page of how to use the "ls" command)
                         Note:  Typing "q" exits any manual page
```

Listing contents of a directory

```
ls   (basic command which lists files/directories in current directory)
ls -a   (as above, but also lists "hidden" files)
ls -la   (as above, but displays more information about files)
ls -lath   (displays more info about files, in order of their creation)
ls -lath | more   (uses "more" command to view file list a page at a time)
```

What directory am I in?

```
pwd   (prints your current "working directory")
```

Moving into a different directory (like a "folder" in a desktop environment)

```
cd directory_name   (moves you into a directory of given name)
cd /directory_name   (moves you to the top (root) level given directory name)
cd   (moves you into your home directory:  /home/username)
cd ..   (moves you one level up a directory)
cd -   (moves you into directory you were previously in)
cd dir1/dir2   (moves you into a subdirectory of directory 1)
```

Creating a new directory

```
mkdir directory_name
mkdir ../../dir_name   (creates a new directory two levels "up" in file system)
```

Renaming a file

```
mv old_filename new_filename
```

Moving a file

```
mv filename ../   (moves file up one directory level)
mv filename /home/username/my_files   (moves file into specified directory)
```

Copying a file

```
cp filename ../../dir1   (copies file two directories up, and into dir1)
cp filename new_filename   (makes a copy of a file with a new name)
```

Removing a file

```
rm filename   (will prompt you if you really want to remove the file)
rm -rf filename   (removes file/directory...will not prompt you...be careful!!!)
```

Viewing contents of a text file

```
more file_name   (using the space key will scroll through file)
```

Creating/appending a text file

```
cat > new_file_name   (Key in data in new file (or overwrite), Ctrl-D to end)
cat >> file_name   (Appends keyed in data in file, Ctrl-D to end)
cat old_file > new_file   (copies data from old_file into new_file)
cat file1.txt file2.txt > both.txt   (takes data from two files and combines
      them into a new file
```

Searching for a string in a file

```
grep string_of_text filename
grep -i string_of_text filename   (not case-sensitive search)
grep -c string_of_text filename   (gives count of number of lines that match)
```

Changing write/read protection of a file

```
      chmod ### filename   (Change file protection for filename:
                   first digit:  User who owns the file
                   second digit:  Users in same group as file owner
                   third digit:  Everyone else)
      Digits are as follows:
      0:  No access
      1:  Execute permission only (as in a directory or other executable file)
      2:  Write permission only
      3:  Write and execute permissions only
      4:  Read permission only
      5:  Read and execute permissions only
      6:  Read and write permissions only
      7:  Read, write, and execute permissions
      Example:  chmod 640 weather.dat
      Makes "weather.dat" readable and writable for user, readable by group, but no
access to anyone else.
```

Logging off

`logout` or `exit`

UNIX tricks

Using a recently typed command

Simply type an "up arrow" on the keyboard to access your last used command. Continue typing "up" to see the next most recent command, and so on.

Viewing your command history and re-typing past commands

Simply type `history`

You'll notice that each command in your "history" has a number associated with it. For example, if you wanted to re-type command line #108 in your history, you can simply type: `!108`

Additionally, if you wanted to re-type the last time you used the more command, but didn't want to re-type the entire line, you can simply type: `!more`

Removing/copying/moving/listing multiple files at once

You can use the * key to remove multiple files that have a portion of their name in common. Some examples:

```
      rm albany.*   (removes all files beginning with the string "albany.")

      cp *.gem /home/ktyle/gem_files/   (copies all .gem files to a given directory)

      rm -rf *   (removes all files in your current directory without prompting.
                   This is very dangerous!!)

      mv *kmsp* /home/ralazear/mn/   (moves all files with the string "kmsp"
embedded somewhere in the file name to a given directory)

      ls *.dat   (in current directory, lists only the files ending in ".dat")
      ls r*   (in current directory, lists all files starting with "r")
```

Redirecting file output

```
      weather -c flatmetar alb 12 > alb_metar.dat   (creates new file/overwrites
existing file with METAR data from weather program)
      weather -c flatmetar alb 12 >> alb_metar.dat   (creates new file/appends to
existing file with METAR data from weather program)
```

Text editors

These three text editor programs use a window/graphic interface:

```
gedit
nedit
emacs
```

You can also use `emacs` or `vi` for a terminal (non-window graphic) interface.