Unix/Linux Cheat Sheet

I want to …		Example
Get Help	Display manual	**man** *CommandName* *CommandName* **--help** or **-h**
	Get help	
	When the computer tells you "no" make it say "yes"	**sudo** *MakeMeASandwich*
Navigate and Manipulate Files and Directories	Dude, where am I?	**pwd**
	Show files & directories with details	**ls -l** *path/to/dir*
	Specify parent or present directory in a path	**../** **./**
	Change directory	**cd** *path/to/dir*
	Show contents of a file named *FileName*	**less** *path/to/file*
	Chng user permissions on files and directories read=4 write=2 execute=1	**chmod 777** *FileName*
	Create a directory	**mkdir** *NewDirectoryName*
	Remove a directory or file	**rm -rf** *path/to/dir* **rm** *path/to/file*
	Copy a file	**cp** *path/to/orig_file* *path/to/duplicate_file*
	Rename or move files & directories	**mv** *path/to/file new/path/to/file*
Bash Short Cuts	Populate command line with previous command	↑
	Go to the start or end of a command line	**ctrl-a** **ctrl-e**
	Autocomplete at the command line	**TAB**
Edit Text	Open file in text editor	**nano** *path/to/file*
Filter file contents	Search a file for a specific string of characters, such as *1168*, and return the lines with the search pattern . See redirecting below.	**grep** ***'1168'*** *path/to/file*
Extract Files	Extract from tarball (`*.tar.gz`) Extract from most compression formats Extract from `*.zip`	**tar -xf** *path/to/file.tar.gz* **gunzip** *path/to/file.gz* **unzip** *path/to/file.zip*
Scripting	Run a script with bash	**bash** *path/to/ScriptName.sh*
Pipelining and Redirecting	Use the output from one command as input for the next command	***command1*** **|** ***command2***
	Store the output of a command into a file	**command >** *path/to/newfile.txt*
	Append the output of a command into an existing file	**command >>** *path/to/file.txt*
	Merge files together	**cat** *File1 File2 > NewFile*

Cancel command running now: **ctrl-c** Exit text viewer: **q** Exit nano text editor: **ctrl-x** Send cmd to background: **ctrl-z** **bg**

C. Bird, Texas A&M University– Corpus Christi

Package Installation

Standard Application Installation:

Install your desired application.

```
/usr/bin $ sudo apt-get install ApplicationToBeInstalled
```

Update Existing Applications:

Upgrade all of the installed applications with updates in the APT repository.

```
usr/bin$ sudo apt-get upgrade
```

Install Applications from a Website:

There are 9 basic steps to install an application in Linux. With some applications you may need to complete all of the steps below, whereas with others you may need to only complete a few of these steps.

0. In your GUI web browser navigate to the distribution webpage of the application you want to install.

1. Find the user manual.

Things to look for include:

- System requirements, Dependencies, Instructions for installation

By reading the user manual you will be able to determine which of the remaining steps you need to complete to install the application.

2. Install any dependencies.

3. Download the application package.

4. Unpack the application package.

a. Unzip the gunzipped file

```
gunzip ApplicationPackage.tar.gz
```

b. Untar the tar file

```
tar xf ApplicationPackage.tar
```

5. Make a configuration file if one is not provided.

6. Run the configuration file to check for dependencies and create the files required to build the application from the source code.

```
~Path/to/package$ ./configure
```

7. Run the `Makefile` that was created (this file may already be provided). This will compile the source code and build the application.

```
~Path/to/package$ make
```

8. Install the application binaries.

```
~Path/to/package$ sudo make install
```

Typically you want the program and its associated files in the `/usr/local/bin`. You can confirm the location of the binaries by typing **`locate SpecificFileName`** in the terminal. You may also have to move the binaries manually using the cp or mv command.

9. (Optional) Add directories to the PATH variable as necessary.

Goto your home directory. Make a copy of the `.profile` file. Then open the file named `.profile` in nano and add the following line at the end of the file if the new directory to add to the PATH is `/usr/local/bin/vcftools`:

```
export PATH="$PATH:/usr/local/bin/vcftools"
```