Useful Linux commands

beginning notes:

- **remember: Linux** is **case-sensitive**!
- `nrs-projects-ssh.humboldt.edu` is the host name of a (virtual) Humboldt computer that can access Oracle; you will be using the Oracle `student` database reachable from nrs-projects this semester.
  - You will use `ssh` to **connect** to `nrs-projects-ssh.humboldt.edu` to do much of the course work for this course.
- once you have `ssh`'d to `nrs-projects-ssh.humboldt.edu`, here is a collection of Linux commands that you might find useful:

help-related commands:

`man` *desired_command*	display the Linux manual page for *desired_command*, if it exists
`apropos` *string*	display names of Linux commands followed by 1-line descriptions for commands whose 1-line descriptions contain *string*

directory-related commands:

`cd`	change directory; make the home directory the current working directory
`cd` *directory_name*	change the current working directory to *directory_name*
`.`	a nickname for the current directory
..	a nickname for the parent of the current directory
~*username*	a nickname for *username*'s current directory
~	a nickname for the current user's home directory
`pwd`	give the name of the current (present) working directory
`mkdir` *directory_name*	make a new directory named *directory_name* within/under the current working directory
`rmdir` *directory_name*	remove the directory *directory_name* within/under the current working directory; note that it must be **empty** for this to work
`ls`	list the contents of the current working directory
`ls -l`	...in "long" format, including file permissions
`ls -ld`	...including permissions and information for subdirectories instead of their contents
`ls` *directory_name*	list the contents of the directory *directory_name*
`chmod 700` *directory_name*	protect the directory *directory_name* so that only **you** can read, write, or execute its contents. **(Note: nrs-projects' web server CANNOT access files in a directory with these permissions)**

file-related commands:

`cp` *filename newfilename*	create a copy of *filename* with the name *newfilename*
`cp` *f1 f2 f3 ... directory_name*	creates copies of files *f1*, *f2*, *f3*, ... (all that you care to list) in the directory *directory_name*
`mv` *filename newfilename*	change the name of the file *filename* to *newfilename*
`mv` *f1 f2 f3 ... directory_name*	moves files *f1*, *f2*, *f3*, ... (all that you care to list) to the directory *directory_name*
`rm` *filename*	remove the file *filename* **(be careful - this cannot be undone!)**
`rm -i` *filename*	slightly-safer way to remove a file -- asks you to confirm removal! (**BUT still cannot be undone!**)
`chmod 600` *filename*	protect the file *filename* - only **you** can read or write it
`more` *filename*	look at the contents of *filename* on-screen, one screen at a time
`cat` *filename*	look at the contents of *filename* on-screen, all at once
`nano` *filename* `vi` *filename* `emacs` *filename*	edit file *filename* (these are three different **text editors** available on `nrs-projects`)

commands and tips for stopping a Linux process:

^C	(typing ctrl key and letter c at the same time) This can often be used to stop or kill a running Linux command (a command running in the foreground). Useful if you accidentally type a command that does more than you want to see (e.g., when you don't want to see the rest of a man page)
`ps x`	gives information about currently-running processes that you own (even from other Linux sessions). The name of each process is on the far right, and the **process id** of each process is in the first column. (**Beware**: the options for `ps` vary on different flavors of Linux/UNIX!)
`kill` *process_id* `kill -9` *process_id*	stop, or kill, the process with process id *process_id*. I was always taught to try the version **without** -9 **before** trying the version with -9, because the former kills the process less "messily". **This command is very useful to kill rogue `sqlplus` sessions if you start getting error messages about tables being locked!**

other commands and etc.:

`sqlplus /`	start up the Oracle SQL*Plus program on nrs-projects
*	Linux wildcard character that matches **any** 0 or more characters. E.g., `ha*s` matches `has`, `ha3s`, `happiness`, etc.
?	Linux wildcard character that matches **any single** character. E.g., `ha?s` matches `hams`, `ha3s` but does **not** match `has`, `haaas`

tab key	in several Linux shells (including nrs-projects' default shell, `bash`), typing this key after you have started typing a file name will cause the shell to try to **complete** (fill in) the file name you have started typing, if it can. This is called **filename completion**.
`grep` *pattern* `*`	look for files in the current working directory that **contain inside** of them the pattern or letters *pattern*
`diff` *file1 file2*	compare the contents of *file1* and *file2*, and show any differences. If the two files are identical, nothing is returned.
`history`	show a list of the most recently-executed commands in this Linux session
!!	redo the last Linux command executed
!`com`	redo the most recent Linux command executed starting with the letters `com`
!-*num*	redo the Linux command executed *num* commands ago
!*num*	redo the Linux command numbered *num* in the history list
up-arrow key	lets you scroll through the commands in the history list
`quota`	On many Linux/UNIX systems, this lets you know how much of your disk space quota you are using. **Note:** how much you can store in your Oracle account is **unrelated** to how much you can store on nrs-projects! To increase how much you can store in your Oracle account (for a good, course-related reason), we have to contact Humboldt's Oracle database administrator.