useful UNIX commands for CIS 130

beginning notes:

- **remember:** UNIX is **case-sensitive**!

- nrs-labs is an HSU computer that can has the official course GNU C++ compiler, `g++`, installed upon it; it also has the course-specific C++ tools we will be using to start out with C++. So, you will use `ssh` to **connect** to nrs-labs to do the C++ course work for this course.

- once you have logged onto nrs-labs, here is a collection of UNIX commmands that you might find useful:

help-related commands:

`man` *desired_command*	print the UNIX manual page for *desired_command*, if it exists
`apropos` *string*	print names of UNIX 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 -a`	...including the so-called "invisible" files (those whose names begin with a period (`.`))
`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. **This should be used for homework directories.**

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** *normally* **cannot be undone! Although** *sometimes* **the file seems to be copied into your** .recycle **directory...)**
rm -i *filename*	...but make me verify that I really want to delete it, beforehand (-i here stands for interactive)
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
pico *filename* nano *filename* vi *filename* emacs *filename*	edit file *filename* (these are different **text editors** available on nrs-labs -- although pico and nano appear to be different names for the same editor...!)

commands and tips for stopping a UNIX process:

^C	(typing ctrl key and letter c at the same time) This can often be used to stop or kill a running UNIX 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 UNIX 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 UNIX/Linux!)
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".

other commands and etc.:

command > filename	run the *command*, but send its output (if any) to *filename* instead of to the screen; this is called **output redirection**.
*	UNIX wildcard character that matches **any** 0 or more characters. E.g., ha*s matches has, ha3s, happiness, etc.
?	UNIX wildcard character that matches **any single** character. E.g., ha?s matches hams, ha3s but does **not** match has, haaas

ESC key	in several UNIX shells (including nrs-labs' default shell), typing this key twice 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 UNIX session
!!	redo the last UNIX command executed
!com	redo the most recent UNIX command executed starting with the letters com
!-*num*	redo the UNIX command executed *num* commands ago
!*num*	redo the UNIX command numbered *num* in the history list
up-arrow key	lets you scroll through the commands in the history list
quota	lets you know how much of your disk space quota you are using. You can request more quota via the HSU **Account Center**, via the **Account Tools** tab, via the **Network Folders Quota and Usage** link. I'm told that you can request up to 50 MB of additional storage before you need additional approval; I can fill out a form to increase your nrs-labs quota more (for a good, course-related reason) -- the form is usually processed within 48 hours.