

CS 235 - Useful UNIX commands for CS 235

beginning notes:

- **remember:** UNIX is **case-sensitive**!
- `nrs-projects.humboldt.edu` is the host name of a (virtual) HSU computer that has Java 8 installed on it; it also has the course submission tool that we will be using for at least parts of the semester.
- you will use `ssh` to **connect** to `nrs-projects` to do some of the early course work for this course, and to submit much of your course work for this course, using a "home-grown" Perl script, `~st10/235submit`. You run this from within the `nrs-projects` directory in which your files to be submitted are stored.
- once you have logged onto `nrs-projects.humboldt.edu`, here is a collection of UNIX commands that you might find useful:

help-related commands:

| | |
|---|---|
| <code>man <i>desired_command</i></code> | display the UNIX manual page for <i>desired_command</i> , if it exists |
| <code>apropos <i>string</i></code> | display names of UNIX commands followed by 1-line descriptions for commands whose 1-line descriptions contain <i>string</i> |

directory-related commands:

| | |
|--|---|
| <code>cd</code> | change directory; make the home directory the current working directory |
| <code>cd <i>directory_name</i></code> | change the current working directory to <i>directory_name</i> |
| <code>.</code> | a nickname for the current directory |
| <code>..</code> | a nickname for the parent of the current directory |
| <code>~<i>username</i></code> | a nickname for <i>username</i> 's current directory |
| <code>~</code> | a nickname for the current user's home directory |
| <code>pwd</code> | give the name of the current (present) working directory |
| <code>mkdir <i>directory_name</i></code> | make a new directory named <i>directory_name</i> within/under the current working directory |
| <code>rmdir <i>directory_name</i></code> | remove the directory <i>directory_name</i> within/under the current working directory; note that it must be empty for this to work |
| <code>ls</code> | list the contents of the current working directory |
| <code>ls -l</code> | ...in "long" format, including file permissions |
| <code>ls -ld</code> | ...including permissions and information for subdirectories instead of their contents |
| <code>ls <i>directory_name</i></code> | list the contents of the directory <i>directory_name</i> |

| | |
|--|--|
| <code>chmod 700 <i>directory_name</i></code> | protect the directory <i>directory_name</i> so that only you can read, write, or execute its contents. This should be used for homework directories. |
|--|--|

file-related commands:

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

commands and tips for stopping a UNIX process:

| | |
|---|--|
| <code>^C</code> | (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) |
| <code>ps x</code> | 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 <code>ps</code> vary on different flavors of UNIX/Linux!) |
| <code>kill <i>process_id</i></code> <code>kill -9 <i>process_id</i></code> | stop, or kill, the process with process id <i>process_id</i> . 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.:

| | |
|---|--|
| <code>javac <i>ClassName</i>.java</code> | compile the Java source code in the file <i>ClassName</i> .java into Java bytecode; results in at least the Java bytecode file <i>ClassName</i> .class, and possibly additional Java bytecode files (one per non-public class also contained within <i>ClassName</i> .java's source code) |
| <code>java <i>ApplicClassName</i></code> | use the Java Virtual Machine (JVM) to interpret and run the Java application class <i>ApplicClassName</i> (whose Java bytecode is in <i>ApplicClassName</i> .class); starts execution at its <code>main</code> method |
| <code>javadoc <i>ClassName</i>.java</code> | generate HTML documentation for <i>ClassName</i> based on its javadoc-style comments |
| * | UNIX wildcard character that matches any 0 or more characters. E.g., <code>ha*s</code> matches <code>has</code> , <code>ha3s</code> , <code>happiness</code> , etc. |
| ? | UNIX wildcard character that matches any single character. E.g., <code>ha?s</code> matches <code>hams</code> , <code>ha3s</code> but does not match <code>has</code> , <code>haas</code> |
| tab key | in several UNIX shells (including <code>nrs-projects</code> ' default shell, <code>bash</code>), 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 . |
| <code>grep <i>pattern</i> *</code> | look for files in the current working directory that contain inside of them the pattern or letters <i>pattern</i> |
| <code>diff <i>file1</i> <i>file2</i></code> | compare the contents of <i>file1</i> and <i>file2</i> , and show any differences. If the two files are identical, nothing is returned. |
| <code>history</code> | show a list of the most recently-executed commands in this UNIX session |
| <code>!!</code> | redo the last UNIX command executed |
| <code>!com</code> | redo the most recent UNIX command executed starting with the letters <code>com</code> |
| <code>!-num</code> | redo the UNIX command executed <i>num</i> commands ago |
| <code>!num</code> | redo the UNIX command numbered <i>num</i> in the history list |
| up-arrow key | lets you scroll through the commands in the history list |
| <code>quota</code> | On many UNIX/Linux systems, this lets you know how much of you disk space quota you are using. This does not seem to be set up on <code>nrs-projects</code> at this time, but just in case it is set up at some point, I'm still including it in this list. |