

1. start BASH shell: `bash`
2. access previous commands in bash shell: up and down arrow keys
3. slogin to a UCS host from another Unix, Linux, etc. host:

`slogin path-to-host`
`slogin clid@path-to-host`

examples: `slogin d100.ucslouisiana.edu`
`slogin abc1234@d100.ucslouisiana.edu`
4. open nedit editor at the terminal or in an slogin session: `nedit file-name &`
5. Unix / Linux root directory path: `/`
6. Unix / Linux relative path to file in local directory: `directory/file`
7. Unix / Linux absolute path to file : `/directory/directory/.../directory/file`
8. UCS network and WANT lab network home directories path: `/home/clid`
9. shortcut path to home directory: `~/`
10. shortcut path parent directory: `../`
11. this directory (i.e. here): `.`
12. show path of current (working) directory: `pwd`
13. make a directory: `mkdir directory-name`
14. copy a file: `cp from-path to-path`
15. copy a file to here: `cp from-path .`
16. rename a file: `mv old-file-name new-file-name`
17. move a file: `mv current-path new-path`

18. change to a directory: `cd path-to-directory`
19. change to home directory: `cd`
20. change to root directory: `cd /`
21. list the contents of a file: `cat path`
22. list the contents of a file one screen at a time: `more path`
23. clear the terminal window / screen: `clear`
24. list normal files in a directory: `ls`
25. list normal files in a directory with information: `ls -l`
26. list all files in a directory: `ls -a`
27. list all files in a directory with information: `ls -la`
28. pipe output from one command to another command: `command | command`
29. list files in a directory one screen at a time: `ls | more`
30. delete a file: `rm path`
31. delete a directory and its subdirectories: `rm -rf path-to-directory`
32. change a DOS / Windows text file that contains extra characters at the end of each line to a Unix text file (the same name can be used as the source and target) : `dos2unix source-file-name target-file-name`
33. Compile a single file with the CC compiler to a.out: `CC prog.cpp`
34. Compile a single file with the CC compiler to a given name: `CC -o target prog.cpp`
35. Compile multiple files at once with the CC compiler to a.out: `CC prog1.cpp prog2.cpp ... progn.cpp`
36. Compile multiple files at once with the CC compiler to a given name:

`CC -o target prog1.cpp prog2.cpp ... progn.cpp`

37. Compile files only when needed with the CC compiler, then compile to a given name:

```
CC -c prog1.cpp
CC -c prog2.cpp
...
CC -c progn.cpp
CC -o target prog1.o prog2.o ... progn.cpp
```

38. The Gnu gcc compiler is available on most Unix and Linux installations. Gnu gcc can be accessed to compile a C++ file by substituting “g++” for “CC” in the above examples. (Note: Gnu gcc does not support independent compilation of class templates when multiple files are used to support information hiding.)

39. “bang!” - repeat a command beginning with a letter or letters: *!letter(s)*

40. Up and down arrows can be used to access previous commands in a bash shell.

41. You can check your UCS email from anywhere via the ULL OpenWebMail site:

<http://webmail.louisiana.edu/>

OpenWebMail also allows you to up and download files to and from your UCS account via the WebDisk utility. However, it messes up the ownership properties so that you will not be able to directly send files to the “submit” system uploaded this way. In order to send files to submit that have been uploaded via WebDisk, follow these steps:

a) upload the file to your home directory, NOT YOUR CLASS DIRECTORY

b) copy the uploaded file saved in your home directory to your class directory so that your TA or instructor can access it and you can submit it:

```
cp file-name class-link/path
```

```
example: cp myFile.cpp 2601/proj4
```

Note: *mv* will not accomplish this correctly. Neither will saving directly into your class directory via WebDisk. You must use *cp* so get the settings and ownership right.