

R. Yoshii for cs111 (211, 311)
Unix Help File: (Cheatsheet for the Unix Operating System)

The main server is called **empress.csusm.edu** (located elsewhere on campus; it is the machine you should log onto.

You should **use putty** from any PC on campus to logon to empress.

If you encounter system failures on empress, email empressissues@csusm.edu

This cheat sheet lists essential UNIX commands.

******* IMPORTANT *******

- If you don't know what went wrong,
 type control-C until you get a Unix prompt.
 If that does not work, type control-D's to logout.
- You must always logout fully before going home.

Things surrounded by < > should be replaced by the actual argument.

Each command must be terminated with Return.

Manual:

man <command> - for manual pages on the <command>
man -k <word> - for commands related to the <word>

e.g. **man -k print**

Users:

users - to see who else is logged on your machine
finger <address> - to see the status of another account

e.g. **finger ryoshii**

Files:

ls - list files and directories you have in your current directory
ls -l - for more info on files and directories
ls -F - can tell which are files and which are directories
ls <name> - if <name> is a file, then list just that file.
 - if it's a directory, list the files and directories it contains.

rm <file> - to delete <file> -- permanently!

more <file> - to read files one screen at a time (press space bar)

cp <file> <to-file> - to copy files

mv <file> <to-file> - to rename files
 or move them from one directory to another
 (or both at once)

DIRECTORIES: (like Folders on a Mac or under Windows)

(Don't play with these until you know what you are doing)

Special names and terms for directories:

"current directory": The "current" directory (or "working" directory) is whichever one you are "in" at any given moment as long as you're in UNIX. It is the one whose contents are immediately available to you. In contrast, to reach a name in any other directory, you have to qualify that name to show how to reach that directory from the current one. We talk about typical ways of qualifying below.

"parent directory": Except for one single topmost directory, every directory in UNIX is actually inside another directory. The directory that contains directory <x> is called the "parent" or "parent directory" of <x>.

"child directory": Opposite of the "parent" relationship. One directory can hold (therefore be parent to) many child directories.

"sibling directory": Another directory within the same parent directory.

"directory tree": The heirarchical organisation of the file system under UNIX. Like a family tree, it starts at the top from a single directory, "the root", which holds files and some directories, and each (or most) of these directories includes more directories, expanding on downward like an upsidedown tree. All our login directories are (typically) at the third level of this tree.

"up", "upward": Toward the root, from anywhere in the directory tree. So a parent directory is "upward" from any of its child directories.

"down", "downward": The opposite: toward directories lower in the tree than the current one (if any).

Special symbols for directories:

. (Yes, just a single period): this is a nickname for whatever directory you're in at the time. Most of the time we don't need it, but sometimes it has its uses.

.. (Two periods) : this is the nickname for the parent of your current directory.

~ (The tilde sign): this is a nickname for your login directory, no matter what directory you are in at the time.

/ : Punctuation following a directory name, usually when followed by the name of a file or directory inside it.

Your home directory is "~<your login name>" (without the quotes): for example, mine is ~ryoshii. When you first login, this is your current directory.

Essential commands for directories:

mkdir <dir> - to create a new directory in your current directory

- cd <dir>** - to go to <dir>, making it become the current directory (actually, "cd" means "change directory", but "go to" is probably more descriptive.)
- cd ..** - to go up a level.
DON'T CONFUSE THIS with going back to your home directory. That will result only when your home directory happens to be the one immediately above your current directory.
- cd** - (all by itself) to change back to your top-level directory, no matter which directory is current.
- rmdir <dir>** - to delete an empty directory; if anything is still in <dir>, rmdir won't remove it.

<file>s in any command can be preceded by directory names to refer to files in other directories:

- e.g.** assignments/hw2
the file called "hw2" in the directory called "assignments", which is in the current directory.
- e.g.** assignments/part1/hw2 (a complete path down to hw2 must be given)
- e.g.** ~/spanish/assignments/term-paper
the file called "term-paper" in the directory "assignments" in the directory "spanish" in your home directory.
NOTE: ~spanish would NOT be the same: it would refer to a login directory (somebody else's) called "spanish".

TIP: If you start with "~", it's a shorthand for your top-level directory, no matter which of your directories is current.

TIP: When copying or moving files between directories, always cd to the destination directory first, and issue the command after that.
This way you know exactly where the destination is.

Examples of use and results:


-
- Action: first you login and create a new terminal window.
Result: you are in your home directory, ~<your-login>
- Action: mkdir CS111
Result: ~<your-login> now contains a child directory called "CS111"
(Question: what directory is now CS111's parent?)
- Action: create a file MyFile in the home directory (with emacs or any other available way). Use "ls".
Result: entries for MyFile and CS111/ will be shown.
- Action: cd CS111
Result: your current directory is now CS111. Nothing visible happens!
- Action: cp ../MyFile . (YES, the third word is a period ".")
Result: a copy of MyFile from the parent directory is now in the current directory. Nothing visible happens!
- Action: ls
Result: an entry for MyFile will be shown.
- Action: /cs/cs311/lab2.out
Result: the program called "lab2.out" in my /cs/cs311 is executed.

- Action: `cp /cs/cs311/hw1 .` (YES, the third word is a period ".")
Result: a copy of hw1 from my /cs/311 is now in your current directory.
- Action: `ls`
Result: entries for MyFile and hw1 will be shown.
- Action: `cd ..`
Result: current directory becomes CS111's parent, ~<login-name>.
- Action: `mv CS111/hw1 lab1`
Result: the file that was hw1 in CS111 is now the file lab1 in the current directory. NOTE: this is moving, not copying, the file.
- Action: `cd CS111`
Result: current directory is once again CS111
- Action: `mv ../lab1 hw1`
Result: the file lab1 from the parent directory is now the file hw1 in the current directory. Note this happens to be undoing the previous mv command (anybody can make a mistake!)

GOING HOME:

logout - to log off (don't forget to do this each time!!!!)
This is logging off from empress, not from the PC.

Email: (see help-pine)

pine  for reading and sending messages and files;

- you must provide your account name and password to use it.
- **do NOT use the mouse at all while you are in pine.**
 - I for reading messages
(then S or E to save messages into a file)
 - C for sending messages
(you can use ^R to enclose a file in your message)
 - A for setting up nick names for addresses
 - Q to quit pine to go back to unix

(always look at the bottom of the window for various commands)
Note: Pine and Web Mail share files. Delete in one and it is deleted in another.

Editors: (see help-emacs)

vi and **emacs** are useful editors.
vi has two modes - typing and editing
emacs has one mode - multiple control characters for commands

C++ Files: (see help-gcc-compiler)

`man g++` for details.

g++ <list .C files here> to compile (**never list .h files**)
./a.out to execute

script <filename> to record test results

run your program here

exit -- everything from script to exit will be recorded in the file.
You should download the script file to your PC and print it.