UNIX C Shell Cheat Sheet

Setup

Terminal Setup	`% stty erase '^?' kill '^U' intr '^C'` `% set term=vt100`
Setting Path Variable	`% set path=(directory ... directory)`
Password Changing	`% passwd`
Changing Default Shell	`% chsh`
Help	`% man command` `% man -k topic`
Logging Out	`% logout` `% <CTRL-D>`

File Manipulation

Listing Files	`% ls`

Displaying Files	`% cat filename`
	`% more filename`
	   `<space>` next page
	   `<return>` next line
	   `q` quit

Copying Files	`% cp file1 file2`
	`% cp file1 [file2] ... directory`
	`% cp -r directory1 directory2`
	
	Use the `-i` option to prompt before overwriting files.

Moving Files	`% mv file1 file2`
	`% mv file1 [file2] ... directory`
	`% mv directory1 directory2`
	
	Use the `-i` option to prompt before overwriting files.

Removing Files	`% rm filename`
	`% rm -r directory`
	
	Use the `-i` option to prompt before deleting files.

Printing Files	`% lpr filename`

Directory Manipulation

Creating Directories	`% mkdir directory`
Changing Directories	`% cd directory` `% cd`
Displaying Current Directory	`% pwd`
Removing Directories	`% rmdir directory`

Miscellaneous Commands

Finding Files	`% find directory -name filename -print`
Searching Files	`% grep string filename`
Sorting Files	`% sort [-o output-file] filename`
Comparing Files	`% diff filename1 filename2` `% cmp filename1 filename2`

File Protection

User and Permission Types	`u` file owner (`u` is short for user) `g` group members `o` all other users `r` read permission `w` write permission `x` execute permission (for directories, allows users to use the directory name in a pathname)
Displaying File Protection	`% ls -l filename`

Changing File Protection	`% chmod [ugo][+-=][rwx] filename`
	`% chmod nnn filename`
	
	    numeric protection mode table
	
	         user group other
	`r` 400 40 4
	`w` 200 20 2
	`x` 100 10 1

Default Protection	`% umask nnn` The `umask` value is subtracted from `666` for new files and from `777` for new directories.

I/O Redirection

Input Redirection	`% command < filename`
Output Redirection (overwrite)	`% command > filename`
Output Redirection (append)	`% command >> filename`
Error Redirection (overwrite)	`% command >& filename`
Error Redirection (append)	`% command >>& filename`
Noclobber Variable	`% set noclobber` Setting the `noclobber` variable protects files from being accidentally overwritten due to output redirection (but not `cp`, `mv`, etc.). "`!`" following any redirection operator overrides `noclobber`.
Piping Operator	`% command1	command2`

C Shell Specific Commands

Enable Tracking of Commands	`% set history=n`
Displaying Command History	`% history`
Re-executing Previous Commands	`!!`　　　　previous command `!n`　　　　*n*th command `!string`　　last command beginning with *string* `!-n`　　　　*n*th most recent command `!?string?`　last command containing *string*
Saving Command History	`% set savehist=n`
Setting Aliases	`% alias alias-string command-string`
Displaying Aliases	`% alias`
Deleting Aliases	`% unalias alias-string`

Special Files

`.cshrc`	This file, if it exists in your home directory, is automatically executed at login. Must begin with "#" to indicate C shell script. `.cshrc` typically includes C shell specific commands such as the `set noclobber`, `set history`, `set savehist`, and `alias` commands.
`.login`	This file, if it exists in your home directory, is automatically executed at login. Must begin with "#" to indicate C shell script. `.login` typically contains generic UNIX commands such as the `stty`, `set path`, and `umask` commands.
`.logout`	This file, if it exists in your home directory, is automatically executed at logout. Must begin with "#" to indicate C shell script. `.logout` typically contains commands to perform cleanup.

Job Control

Running Jobs in the Background	`% command &`

Stopping Foreground Jobs	`% command`
	　　　.
	　　　.
	　　　.
	`<CTRL-Z>`
	`%`

Job Status	`% jobs`
Killing Jobs	`% kill %job-number` `% kill process-id`
Bringing Jobs to the Foreground	`% fg %job-number`
Moving Jobs to the Background	`% bg %job-number`
Process Status	`% ps`　　　lists all your processes `% ps -a`　lists all running processes `% ps -x`　full listing for given option
Displaying Users	`% who` `% whoami`