**Mac OSX**

Moving Files and Directories

MV Options

|  |  |
| --- | --- |
| Option | Description |
| -n | No overwriting |
| -f | Force overwriting |
| -i | Interactive overwriting, “ask me” |
| -v | verbose |

Copying Files and Directories

Cp options

|  |  |
| --- | --- |
| Option | Description |
| -n | No overwriting |
| -f | Force overwriting |
| -i | Interactive overwriting, “ask me” |
| -v | Verbose |

Deleting files and directories

* *rm*
  + Only removes files
* *rmdir*
  + Only removes empty directories
* *rm –R*
  + Removes files and directories recursively
  + By default, doesn’t care if directories are empty

Hard Links

* *ln filetolink hardlink*
* Reference a file in the filesystem
* Do not break if file is moved
* Do not break if file is deleted

Symbolic Links

* “sym link”
* *ln –s filetolink symlink*
* Reference a file path or a directory path
* Break if a file is moved
* Break if a file is deleted

Searching for Files and Directories

*find path expression*

*find ~/Documents –name “someimage.jpg”*

Wildcard Characters

|  |  |
| --- | --- |
| Wildcard | Meaning |
| \* | Zero or more characters (glob) |
| ? | Any one character |
| [ ] | Any character in the brackets |

File Organization

Typical Unix Organization

|  |  |
| --- | --- |
| Directory/Folder | Contents |
| / | Root |
| /bin | Binaries, programs |
| /sbin | System binaries, system programs |
| /dev | Devices: hard drives, keyboard, mouse, etc |
| /etc | System Configurations |
| /home | User home directories |
| /lib | Libraries of code |
| /tmp | Temporary files |
| /var | Various, mostly files the system uses |
| /usr  /usr/bin  /usr/etc  /usr/lib  /usr/local | User programs, tools and libraries (not files) |

Mac-only Files and Directories

|  |  |
| --- | --- |
| Directory/Folder | Contents |
| /Applications | Mac Programs |
| /Library | Mac libraries of code |
| /Network | Networked devices |
| /System | Mac OS X |
| /Users | User home directories |
| /Volumes | Mounted volumes (hard drive, DVD, iPod) |
| .DS\_Store | Holds folder view options icon positions |
| ~/.MacOSX | Directory for Mac OS X to store options |
| ~/.Trash | User trash can |
| ~/.hotfiles.btree | Track commonly-used files for optimization |
| ~/.Spotlight-V100 | User for spotlight indexing |

Unix File Naming

* Maximum of 255 characters
* Avoid / \ \* & % ? $ | ^ ~ < > and most other symbols
* Use A-Z, a-z, 0-9, period, underscore, hyphen
* Typically lowercase
  + “MyFile” and “myfile” would be different
* Underscores are better than spaces
  + Escape spaces with \
  + Use quotes around names with spaces
* File endings (.txt, .png, .html, etc.) not required but helpful
  + Differentiates files from commands and directories

Creating Files

* Primary techniques to create files
  + Unix text editors
  + Direct output to file
  + *touch*

Unix Text Editors

* **ed** (Edit text)
  + **Earliest Unix editor, not user-friendly**
* **vi** (visual editing mode), **vim** (vi improved)
  + **Modal, fingers rarely leave keyboard home row**
* **GNU Emacs** (editor macros)
  + **Macros to automate work, swiss army knife**
* **pico** (pine composer), **nano (1000x larger than pico)**
  + **Basic features, easy to use**

Reading Files

* *cat*
  + Concatenate
* *more*
  + Paginated output
* less
  + Backward scrolling
  + Better memory use
  + Less > more

Reading Portions of Files

* head
  + Display lines from beginning of a file
* tail
  + Display lines from end of a file
* tail –f
  + “Follow” the tail of a file

File Permissions: Alpha Notation

|  |  |  |  |
| --- | --- | --- | --- |
|  | user | group | other |
| Read (r) | Yes | Yes | Yes |
| Write (w) | Yes | Yes | No |
| Execute (x) | Yes | No | No |

File Permissions: Alpha Examples

*chmod ugo=rwx filename*

*chmod u=rwx,g=rw,o=r filename*

*chmod ug+w filename*

*chmod o-w filename*

*chmod ugo+rw filename*

*chmod a+rw filename*

File Permissions: Octal Notation

|  |  |  |  |
| --- | --- | --- | --- |
|  | User | Group | Other |
| Read (r) | 4 | 4 | 4 |
| Write (w) | 2 | 2 | 0 |
| Execute (x) | 1 | 0 | 0 |
| Total | **7** | **6** | **4** |

File Permissions: Alpha Vs. Octal

rwxrw-r-- = 764

File Permissions: Octal Examples

*chmod 777 filename*

*chmod 764 filename*

*chmod 755 filename*

*chmod 000 filename*

File Permissions: Alpha to Octal Conversion

r = 4

w = 2

x = 1

The Root User

* Superuser account that can do anything on the system
* Root user is disabled by default in Mac OS X
* Why talk about it?
  + Important Unix concept
  + May read or hear references to “root”
  + Remote Unix servers usually have the root user enabled
  + Important when discussing sudo

Command Basics

* Command options: -v, --version, --help
* Exit: q, x, ctrl + q, ctrl + x, or ESC
* Force quit: Control + c
  + Or close window (but process may keep running)
* Semicolons between commands
* *whereis, which, whatis*

Text File Helpers

* *wc*
  + word count
* *sort*
  + sort lines
* *uniq*
  + filter in/out repeated lines

Utility Programs

* *cal / ncal*
  + calendar
* *bc*
  + calculator
* *expr*
  + expression evaluator
* *units*
  + unit conversion

Command History Shortcuts

|  |  |
| --- | --- |
| **Command** | **Description** |
| !3 | References history command #3 |
| !-2 | References command which was 2 commands-back |
| !dog | References most recent command beginning with “dog” |
| !! | References previous command |
| !$ | References previous command’s arguments |

Command Prompt Formatting Codes

PS1 = Prompt string 1

|  |  |
| --- | --- |
| Command | Command Prompt Output |
| \u | username |
| \s | Current shell |
| \w | Current working directory |
| \W | Basename of current working directory |
| \d | Date in “weekday month date” format (“Mon Jan 16”) |
| \D{format} | Date in strftime format (“%Y-%m-%d”) |
| \A | Time in 24-hour HH:MM format |
| \t | Time in 24-hour HH:MM:SS format |
| \@ | Time in 12-hour HH:MM am/pm format |
| \T | Time in 12-hour HH:MM:SS format |
| \H | Hostname |
| \h | Hostname up to first “.” |
| \! | History number of this command |
| \$ | When UID is 0 (root), a “#”, otherwise “$” |
| \\ | A literal backslash |

GREP: Searching with regular expressions

* grep
  + Output all lines matching a regular expression
  + Global Regular Expression Print
    - In Unix ed editor: g/re/p
* Regular Expression
  + Expression for describing a pattern in text strings
  + “regex” for short
* Useful in other contexts
  + Most programming languages
  + Many text editors

Standard Input and Standard Output

* Standard input
  + Stdin
  + Keyboard
  + /dev/stdin
* Standard output
  + Stdout
  + Text terminal
  + /dev/stdout

Supressing Output

* /dev/null
  + “null device”, “bit bucket”, “black hole”
  + Similar to special files /dev/stdin and /dev/stdout
  + Unix discards any data sent there

Profile, Login, and, resource files

* Upon login to a bash shell
  + /etc/profile
  + **~/.bash\_profile,** ~/.bash\_login, ~/.profile, ~/.login
* Upon starting a new bash subshell
  + ~/.bashrc
* Upon logging out of bash shell
  + **~/.bash\_logout**
    - Useful if you want some scripts to run
* The load orders for other shells are on wiki

Bash Shells: Login Vs. Non-Login

Add to ~/.bash\_profile:

*if [ -f ~/.bashrc ]; then*

*source ~/.bashrx*

*fi*

Then put all shell configuration in ~/.bashrc

Screen Capture

Screencapture options

|  |  |
| --- | --- |
| Option | Description |
| -i | Interactive capture (‘esc’ to cancel) |
| -m | Main monitor only |
| -C | Show cursor |
| -t | Format (png, pdf, jpg, tiff) |
| -T | Delay in seconds |
| -P | Open file with preview |
| -M | Open file in Mail message |
| -c | Capture to clipboard |

Shutdown, Reboot, and Sleep

* *shutdown*
  + Requires sudo
  + Options
    - h: halt at specified time
    - r: reboot at specified time
    - s: sleep at specified time
  + Time format
    - Now
    - +minutes
    - yymmddhhmm

Shutdown now (Same as Finder)

*sudo shutdown –h now*

Shutdown in 45 minutes

*sudo shutdown –h +45*

Shutdown on June 1, 2011 at midnight

*sudo shutdown –h 1106010000*

Reboot now (same as Finder)

*sudo shutdown –r now*

Reboot in 45 minutes

*sudo shutdown –r +45*

*Reboot on June 1,2011 at midnight*

*sudo shutdown –r 110610000*

Sleep now (same as Finder)

*sudo shutdown –s now*

Text to Speech

Read a file:

say –f file\_to\_read.txt

Output audio as AIFF file:

say –f file\_to\_read,txt –o audio\_file.aiff

Alert when actions complete:

cp –R dir1 dir1; say ‘Directory copy done’

For fun:

sleep 300; say “Hello”; sleep 10; say “I said Hello. Is anyone there?”

Changing System Configurations

*defaults read <domain> <key>*

*defaults write <domain> <key> <value>*

Domain is typically in the format com.companyname.appname

Show “.” Files in Finder:

*defaults write com.apple.finder AppleShowAllFiles –bool TRUE*

Display Unix path in Finder windows

*defaults write com.apple.finder \_FXShowPosixPathInTitle –bool TRUE*

Modify screen capture type:

*defaults write com.apple.screencapture type PNG*

Formats: PNG, BMP, GIF, JPEG, PDF, PICT, PSD, SGI, TGA, TIFF

Modify screen capture location:

*defaults write com.apple.screencapture location “/Users/kyle/Desktop”*

Change login background:

*sudo defaults write com.apple.loginwindow*

*DesktopPicture “/Library/Desktop Pictures/Aqua Blue.jpg”*

Regular Expression Basics

|  |  |  |
| --- | --- | --- |
| Regex | Meaning | Example |
| . | Wild card, any one character except line breaks | gre.t |
| [ ] | Character set, any one character listed inside [ ] | gr[ea]ly |
| [^ ] | Negative character set, any one character not listed inside [ ] | [^aeiou] |
| - | Range indicator (when inside a character set) | [A-Za-z0-9] |
| \* | Preceding element can occur zero or more times | file\_\*name |
| + | Preceding element can occur one or more times | gro+ve |
| ? | Preceding element can occur zero or one time | colou?r |
| | | Alternation. Or operator | (jpg|gif|png) |
| ^ | Start of line anchor | ^Hello |
| $ | End of line anchor | World$ |
| \ | Escape the next character (“\+” is literal “+” character) | image\.jpg |
| \d | Any digit | 20\d\d-06-09 |
| \D | Anything not a digit | ^\D+ |
| \w | Any word character (alphanumeric + underscore) | \w+\_export\.sql |
| \s | Whitespace (space, tab, line break) | \w+\s\w+ |
| \S | Anything not whitespace | \S+\s\S+ |

Regular Expression Character Classes

|  |  |
| --- | --- |
| Class | Represents |
| [:alpha:] | Alphabetic character |
| [:digit:] | Numeric characters |
| [:alnum:] | Alphanumeric characters |
| [:lower:] | Lower-case alphabetic characters |
| [:upper:] | Upper-case alphabetic characters |
| [:punct:] | Punctuation characters |
| [:space:] | Space characters (space, tab, new line) |
| [:blank:] | Whitespace characters |
| [:print:] | Printable characters, including space |
| [:graph:] | Printable characters, not including space |
| [:cntrl:] | Control characters (non-printing) |
| [:xdigit:] | Hexadecimal characters (0-9, A-F, a-f) |

TR: Deleting and Squeezing Characters

tr options

|  |  |
| --- | --- |
| Option | Description |
| -d | Delete characters in listed set |
| -s | Squeeze repeats in listed set |
| -c | Use complementary set |
| -dc | Delete characters not in listed set |
| -sc | Squeeze characters not in listed set |

*echo “abc1233deee567f” | tr –d [:digit:] # “abcdeef”*

*echo “abc1233deee567f” | tr –dc [:digit:] # “1233567”*

*echo “abc1233deee567f” | tr –s [:digit:] # “abc123deee567f”*

*echo “abc1233deee567f” | tr –sc [:digit:] # “abc1233de567f”*

*echo “abc1233deee567f” | tr –ds [:digit:] [:alpha:] # “abcdef”*

*echo “abc1233deee567f” | tr –dsc [:digit:] [:digit:] # “123567”*

Remove non-printable chracters from file1:

*tr –dc [:print:] < file1 > file2*

Remove surplus carriage return and end of file character:

*tr –d ‘\015\032’ < windows\_file > unix\_file*

Remove double spaces from file1

*tr –s ‘ ’ < file1 > file2*

SED: Stream Editor

sed ‘s/a/b/’

s: substitution

a: search string

b: replacement

DIFf: Comparing Files

Diff comparison options

|  |  |
| --- | --- |
| Option | Description |
| -i | Case insensitive |
| -b | Ignore changes to blank characters |
| -w | Ignore all whitespace |
| -B | Ignore blank lines |
| -r | Recursively compare directories |
| -s | Show identical files |

Diff output formats

|  |  |
| --- | --- |
| Option | Description |
| -c | Copied context |
| -u | Unified context |
| -y | Side-by-side |
| -q | Only whether files differ |

Notes

Kernel and Shells

* Kernel
  + Core of OS
  + Allocates time and memory to programs
  + Mac OS X uses the Mach Kernel
* Shell
  + Outer layer OS
  + What we see when we open terminal
  + Interacts with user
  + Sends requests to kernel
  + Mac OS X uses the bash shell, but includes other choices

Viewing Process

* Whenever a command is executed in the shell (Bash), a file executes and it communicates with the kernel.
  + The kernel then sets aside some space and starts a process running in it.
  + Output from the process is finished it is then returned to the shell.
  + Then that memory space is closed out.
  + The kernel manages process.