The 12 Unix Commands Everyone Should Know

Matt Schnittker
@mschnitt
What We’ll Cover

- open
- diskutil
- ls
- chmod
- chown
- du
- df
- find
- grep
- top
- mv
- rsync

Thursday, July 9, 15
Code samples

https://github.com/mschnitt/psu_2015

git clone https://github.com/mschnitt/psu_2015.git
Overview

• Could easily have been “50 Commands Everyone Should Know”

• Picked commands that help accomplish common tasks, or help illustrate deeper concepts

• More about exposure to concepts than depth

• Sorry if your favorite 12 are not here
The Terminal

- Actually a Unix shell (bash)
- Similar environment to Linux
- Powerful abilities without leaving your keyboard
open

$ man open
OPEN(1)                   BSD General Commands Manual                  OPEN(1)

NAME
   open -- open files and directories

SYNOPSIS
       [-a application] file ... [--args arg1 ...]

DESCRIPTION
   The open command opens a file (or a directory or URL), just as if you had
double-clicked the file's icon. If no application name is specified, the
default application as determined via LaunchServices is used to open the
specified files.
   If the file is in the form of a URL, the file will be opened as a URL.
Opened applications inherit environment variables just as if you had
launched the application directly through its full path. This behavior
was also present in Tiger.
open

- open <item to open>
- Will open a path, file, or URL.
- Uses default application to handle it.
- Great for opening pictures and documents or getting a finder window.
$ open /Applications

$ open /usr/bin

$ open /Applications/Utilities/Console.app

$ open http://www.disneyanimation.com

$ open /Library/Desktop\ Pictures/Frog.jpg

$ open “/Library/Desktop Pictures/Yosemite.jpg”

$ open -a safari http://www.google.com

$ open -a google chrome http://disneycareers.com/en/default/

open_command.sh

Thursday, July 9, 15
$ man diskutil

NAME

diskutil -- modify, verify and repair local disks

SYNOPSIS

diskutil [quiet] verb [options]

DESCRIPTION

diskutil manipulates the structure of local disks. It provides information about, and allows the administration of, the partitioning schemes, layouts, and formats of disks. This includes hard disks, solid state disks, optical discs, CoreStorage volumes, and AppleRAID sets. It generally manipulates whole volumes instead of individual files and directories.
diskutil

- Command line version of disk utility
- Handy tool for quick disk operations or scripting Disk Partitioning.
```
$ diskutil list
/dev/disk0
 #: TYPE NAME                     SIZE IDENTIFIER
 0: GUID_partition_scheme         *1.0 TB  disk0
 1: EFI EFI                       209.7 MB  disk0s1
 2: Apple_HFS Monkeywrench_1TB    999.3 GB  disk0s2
 3: Apple_Boot Recovery HD       650.0 MB  disk0s3
/dev/disk1
 #: TYPE NAME                     SIZE IDENTIFIER
 0: GUID_partition_scheme         *2.0 TB  disk1
 1: EFI EFI                       209.7 MB  disk1s1
 2: Apple_HFS Seagate_2TB         2.0 TB   disk1s2
/dev/disk2
 #: TYPE NAME                     SIZE IDENTIFIER
 0: FDisk_partition_scheme        *2.0 TB  disk2
 1: Apple_HFS Elements_2TB        2.0 TB   disk2s1
```
$ diskutil info /dev/disk0

Device Identifier:       disk0
Device Node:             /dev/disk0
Part of Whole:           disk0
Device / Media Name:     ST1000LM024 HN-M101MBB Media
Volume Name:             Not applicable (no file system)
Mounted:                 Not applicable (no file system)
File System:             None
Content (IOContent):     GUID_partition_scheme
OS Can Be Installed:     No
Media Type:              Generic
Protocol:                SATA
SMART Status:            Verified
Total Size:              1.0 TB (1000204886016 Bytes) (exactly 1953525168 512-Byte-Units)
Volume Free Space:       Not applicable (no file system)
Device Block Size:       512 Bytes
Read-Only Media:         No
Read-Only Volume:        Not applicable (no file system)
Ejectable:               No
Whole:                   Yes
Internal:                Yes
Solid State:             No
OS 9 Drivers:            No
Low Level Format:        Not supported
Device Location:         "Lower"
$ diskutil info /dev/disk0s1

<table>
<thead>
<tr>
<th>Device Identifier:</th>
<th>disk0s2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Node:</td>
<td>/dev/disk0s2</td>
</tr>
<tr>
<td>Part of Whole:</td>
<td>disk0</td>
</tr>
<tr>
<td>Device / Media Name:</td>
<td>1TB Monkeywrench</td>
</tr>
<tr>
<td>Volume Name:</td>
<td>Monkeywrench_1TB</td>
</tr>
<tr>
<td>Mounted:</td>
<td>Yes</td>
</tr>
<tr>
<td>Mount Point:</td>
<td>/</td>
</tr>
<tr>
<td>File System Personality:</td>
<td>Journaled HFS+</td>
</tr>
<tr>
<td>Type (Bundle):</td>
<td>hfs</td>
</tr>
<tr>
<td>Name (User Visible):</td>
<td>Mac OS Extended (Journaled)</td>
</tr>
<tr>
<td>Journal:</td>
<td>Journal size 81920 KB at offset 0x1d1c000</td>
</tr>
<tr>
<td>Owners:</td>
<td>Enabled</td>
</tr>
<tr>
<td>Partition Type:</td>
<td>Apple_HFS</td>
</tr>
<tr>
<td>OS Can Be Installed:</td>
<td>Yes</td>
</tr>
<tr>
<td>Recovery Disk:</td>
<td>disk0s3</td>
</tr>
<tr>
<td>Media Type:</td>
<td>Generic</td>
</tr>
<tr>
<td>Protocol:</td>
<td>SATA</td>
</tr>
<tr>
<td>SMART Status:</td>
<td>Verified</td>
</tr>
<tr>
<td>Volume UUID:</td>
<td>EC3A4E85-A530-3E7D-84A2-91EE4559E1F9</td>
</tr>
<tr>
<td>Disk / Partition UUID:</td>
<td>84587A55-1BD4-4AA8-9349-26E7148D97B8</td>
</tr>
</tbody>
</table>
Usage: `diskutil [quiet] <verb> <options>`, where `<verb>` is as follows:

- `list` (List the partitions of a disk)
- `info[rmation]` (Get information on a specific disk or partition)
- `listFilesystems` (List file systems available for formatting)
- `activity` (Continuous log of system-wide disk arbitration)
- `u[n]mount` (Unmount a single volume)
- `unmountDisk` (Unmount an entire disk (all volumes))
- `eject` (Eject a disk)
- `mount` (Mount a single volume)
- `mountDisk` (Mount an entire disk (all mountable volumes))
- `enableJournal` (Enable HFS+ journaling on a mounted HFS+ volume)
- `disableJournal` (Disable HFS+ journaling on a mounted HFS+ volume)
- `moveJournal` (Move the HFS+ journal onto another volume)
- `enableOwnership` (Treat as exact User/Group IDs for a mounted volume)
- `disableOwnership` (Ignore on-disk User/Group IDs for a mounted volume)
- `rename[Volume]` (Rename a volume)
- `verifyVolume` (Verify the file system data structures of a volume)
- `repairVolume` (Repair the file system data structures of a volume)
- `verifyDisk` (Verify the components of a partition map of a disk)
- `repairDisk` (Repair the components of a partition map of a disk)
- `verifyPermissions` (Verify the permissions of a Mac OS X volume)
- `repairPermissions` (Repair the permissions of a Mac OS X volume)
- `eraseDisk` (Erase an existing disk, removing all volumes)
- `eraseVolume` (Erase an existing volume)
- `reformat` (Erase an existing volume with same name and type)

and more …
NAME
ls -- list directory contents

SYNOPSIS
ls [-ABCFGHLOPRSTUW@abcdefghiklmnopqrstuvwxyz] [file ...]

DESCRIPTION
For each operand that names a file of a type other than directory, ls displays its name as well as any requested, associated information. For each operand that names a file of type directory, ls displays the names of files contained within that directory, as well as any requested, associated information.

If no operands are given, the contents of the current directory are displayed. If more than one operand is given, non-directory operands are displayed first; directory and non-directory operands are sorted separately and in lexicographical order.
ls

- `ls <options> <path>`
- List contents of a directory or file.
- If path is omitted, current directory is used.
- Very powerful when used with expressions
# List contents of the /Volumes directory
$ ls /volumes

# List /usr/bin
$ ls /usr/bin

# Multiple paths
$ ls /usr/bin /usr/sbin/ /bin

# What if you know some of the path? Use a wildcard.
$ ls /usr/sbin/ds*
$ ls /usr/*bin/ds*

# One File Per Line
$ ls -1 /usr/sbin/ds*

# Long File Listing
$ ls -l /usr/sbin/ds*
#Hidden Files In My Home Directory
$ ls -a ~

#Long Listing
$ ls -la ~

#Multiple paths
$ ls /usr/bin /usr/sbin/ /bin

#Three characters, d, s, and anything else.
$ ls /usr/bin/ds?

#Three characters
$ ls /usr/bin/???

#Any three lowercase letters.
$ ls /usr/sbin/[a-z][a-z][a-z]
POSIX Permissions

- User/Group/World
- Read/Write/Execute
- Traditionally in /etc/passwd and /etc/group
inodes

• POSIX representation of File, Directory, link

• Not used in HFS, but has corresponding components

• Much of what you see and can do is based on this older structure.

• Try “stat” on a file.
NAME
chmod -- change file modes or Access Control Lists

SYNOPSIS
chmod [-fv] [-R [-H | -L | -P]] mode file ...
chmod [-fv] [-R [-H | -L | -P]] [-a | +a | =a] ACE file ...
chmod [-fhv] [-R [-H | -L | -P]] [-E] file ...
chmod [-fhv] [-R [-H | -L | -P]] [-C] file ...
chmod [-fhv] [-R [-H | -L | -P]] [-N] file ...

DESCRIPTION
The chmod utility modifies the file mode bits of the listed files as specified by the mode operand. It may also be used to modify the Access Control Lists (ACLs) associated with the listed files.
chmod

- chmod <mode> <item>
- Works on Files and Directories
- Read / Write / Execute = 4 / 2 / 1
- User / Group / World
### chmod bits

<table>
<thead>
<tr>
<th></th>
<th>Read</th>
<th>Write</th>
<th>Execute</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute Only</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Write Only</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Read Only</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Read + Execute</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Everything</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>
#Make the file readable by everyone
$ chmod 444 testfile
# or
$ chmod a+r testfile
$ ls -l testfile
-r--r--r-- 1 mschnitt wheel 0 Jul 5 15:44 testfile

#Make the file readable and executable by everyone
$ chmod 555 testfile
# or
$ chmod a+rx testfile
$ ls -l testfile
-r-xr-xr-x 1 mschnitt wheel 0 Jul 5 15:44 testfile

#Give all permissions to file owner, but none to others
$ chmod 700 testfile
# or
$ chmod a-rwx
$ chmod u+rwx testfile
$ ls -l testfile
-rwx--- 1 mschnitt wheel 0 Jul 5 15:44 testfile
# Give all permission to owner. Read Only to everyone else
$ chmod 744 testfile
$ ls -l testfile
-rwxr--r--  1 mschnitt wheel  0 Jul  5 15:44 testfile

# or
$ chmod a-rwx testfile
$ chmod u+rwx testfile
$ chmod g+r testfile
$ chmod a+r testfile
$ ls -l testfile
-rwxr--r--  1 mschnitt wheel  0 Jul  5 15:44 testfile
NAME
chown -- change file owner and group

SYNOPSIS
chown [-fhv] [-R [-H | -L | -P]] owner[:group] file ...
chown [-fhv] [-R [-H | -L | -P]] :group file ...

DESCRIPTION
The chown utility changes the user ID and/or the group ID of the specified files. Symbolic links named by arguments are silently left unchanged unless -h is used.
chown

- chown <user> <item>
- Change ownership of a directory or file
#Create file and change ownership
$ touch testfile
$ chown guest testfile
$ ls -l testfile
-rw-r--r--  1 Guest  wheel  0 Jul  5 15:44 testfile

#Change it back to me.
$ chown $USER testfile
$ ls -l testfile
-rw-r--r--  1 mschnitt  wheel  0 Jul  5 15:44 testfile
NAME
   du -- display disk usage statistics

SYNOPSIS
   du [-H | -L | -P] [-a | -s | -d depth] [-c] [-h | -k | -m | -g] [-x]
      [-I mask] [file ...]

DESCRIPTION
   The du utility displays the file system block usage for each file argument
   and for each directory in the file hierarchy rooted in each directory argument. If no file is specified, the block usage of the hierarchy rooted in
   the current directory is displayed.
du

- `du <options> <path>`
- Gives disk usage.
- Can take a long time to run since it has to tally up the usage.
How much space am I using in /Applications?
$ du -ks /Applications/Safari.app

Space taken by users folders?
$ sudo du -ks /Users

Personal Library Folder and list all files?
$ du -k ~/Library/Application\ Support/
NAME

df -- display free disk space

SYNOPSIS

df [-b | -h | -H | -k | -m | -g | -P] [-ailn] [-t] [-T type]
    [file | filesystem ...]

DESCRIPTION

The df utility displays statistics about the amount of free disk space on the specified filesystem or on the filesystem of which file is a part. Values are displayed in 512-byte per block counts. If neither a file or a filesystem operand is specified, statistics for all mounted filesystems are displayed (subject to the -t option below).
df

• df <options> <path>

• Gives free space on disk

• Usually runs quickly.
```bash
#$ df -k
filesystem  1024-blocks   Used   Available  Capacity   iused ifree  %iused   Mounted on
/dev/disk0s2  975922976  283462440  692204536    30%  70929608 173051134  29% /

$ df -h
Filesystem Size Used Avail Capacity iused ifree %iused Mounted on
/dev/disk0s2  931Gi 270Gi 660Gi 30% 70929686 173051056 29% /
devfs 189Ki 189Ki 0Bi 100% 652 0 100% /dev
map -hosts 0Bi 0Bi 0Bi 100% 0 0 100% /net
map auto_home 0Bi 0Bi 0Bi 100% 0 0 100% /home
/dev/disk1s2 1.8Ti 702Gi 1.1Ti 38% 183978721 304315945 38% /Volumes/Sgt_2TB
/dev/disk2s1 1.8Ti 876Gi 987Gi 48% 229615733 258762121 47% /Volumes/Elem_2TB
/dev/disk3s0 32Mi 32Mi 0Bi 100% 0 0 100% /Volumes/SCHNITTK

$ df -h /Volumes/*
Filesystem Size Used Avail Capacity iused ifree %iused Mounted on
/dev/disk3s0 32Mi 32Mi 0Bi 100% 0 0 100% /Volumes/SCHNITTK
/dev/disk2s1 1.8Ti 876Gi 987Gi 48% 229615733 258762121 47% /Volumes/Elem_2TB
/dev/disk0s2 931Gi 270Gi 660Gi 30% 70928560 173052182 29% /
/dev/disk1s2 1.8Ti 702Gi 1.1Ti 38% 183978721 304315945 38% /Volumes/Sgt_2TB
```
$ man find

FIND(1)                   BSD General Commands Manual                  FIND(1)

NAME
    find -- walk a file hierarchy

SYNOPSIS
    find [-H | -L | -P] [-EXdsx] [-f path] path ... [expression]
    find [-H | -L | -P] [-EXdsx] -f path [path ...] [expression]

DESCRIPTION
    The find utility recursively descends the directory tree for each path listed, evaluating an expression (composed of the ``primaries'' and ``operands'' listed below) in terms of each file in the tree.
find

• find <path> <options>

• Searches through folder hierarchy

• Commonly used to find things by name, modification date, type or size.

• Can also run commands
#Find the path to the Terminal App
$ find /Applications -name "Terminal.app"
/Applications/Utilities/Terminal.app

#Find applications changed in the last 5 days
$ find /Applications -mtime -5 -name "*.app"

#Search your hard drive.
$ find / -name ls

#Directories owned by root in the current directory
$ find . -type d -uid root
<output not shown>

#Symbolic Links in my home directory
$ find ~ -type l
<output not shown>
Sometimes, permissions are a problem, use sudo!

$ find /Volumes -type d -depth 2
find: /Volumes/Elements_2TB/.Trashes: Permission denied

#sudo to the rescue!
$ sudo find /Volumes -type d -depth 2

#Find Large Files in your users directory
$ sudo find /Users -size +1000k

# ...and modified in the last 2 days
$ sudo find /Users -size +1000k -mtime -2
#Send output to ls
$ find ~ -type f -size +50000k -exec ls -l {} \;
<output not shown>

#Size listing of directories
$ find ~ -type d -exec du -sk {} \;

#Also useful for removing files.. but let’s not run this one.
#Remove everything over 100 days old.
$ find . -mtime +100 -exec rm {} \;

#Remove .DS_Store files
$ find . -name .DS_Store -exec rm {} \;

find_command.sh

Thursday, July 9, 15
$ man grep
GREP(1)                    BSD General Commands Manual                  GREP(1)

NAME
     grep, egrep, fgrep, zgrep, zegrep, zfgrep -- file pattern searcher

SYNOPSIS
     grep [-abcdDEFGHhiJLlmnOopqRSsUVvwxZ] [-A num] [-B num] [-C[num]] [-e pattern]
          [-f file] [--binary-files=value] [--color[=when]] [--colour[=when]]
          [--context[=num]] [--label] [--line-buffered] [--null] [pattern] [file ...]

DESCRIPTION
The grep utility searches any given input files, selecting lines that match one or
more patterns. By default, a pattern matches an input line if the regular expression
(RE) in the pattern matches the input line without its trailing newline. An empty
expression matches every line. Each input line that matches at least one of the pat-
tterns is written to the standard output.

grep is used for simple patterns and basic regular expressions (BREs); egrep can han-
dle extended regular expressions (EREs). See re_format(7) for more information on
regular expressions. fgrep is quicker than both grep and egrep, but can only handle
fixed patterns (i.e. it does not interpret regular expressions).
grep

- grep <options> <pattern> <file>
- Looks for patterns within files.
- Supports regular expressions
#Find a string in a file
$ grep root /etc/passwd
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false

#Case Insensitive
$ grep -i version /Applications/Safari.app/Contents/Info.plist
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
  <key>CFBundleInfoDictionaryVersion</key>
</plist>

#Grep through multiple things at once
$ grep -i version /Applications/*/Contents/Info.plist

#Grep with Regular Expressions (Advanced)
grep "[0-9+\.[0-9+].[0-9+]]" /Applications/*/Contents/Info.plist

grep_command.sh
#Pipe into grep
$ find ~ -type d | grep Mozilla

#Match can be a single letter
$ ls /Applications | grep i

#Or, something more complex
$ ls /Applications | grep Safari
$ man top

DESCRIPTION
The top program periodically displays a sorted list of system processes. The default sorting key is pid, but other keys can be used instead. Various output options are available.

OPTIONS
Command line option specifications are processed from left to right. Options can be specified more than once. If conflicting options are specified, later specifications override earlier ones. This makes it viable to create a shell alias for top with preferred defaults specified, then override those preferred defaults as desired.
#top

Load Avg: 2.45, 2.45, 2.80  CPU usage: 19.81% user, 15.56% sys, 64.62% idle
SharedLibs: 12M resident, 19M data, 0B linkedit.
MemRegions: 58698 total, 2848M resident, 124M private, 1047M shared.
PhysMem: 7849M used (1332M wired), 343M unused.
VM: 611G vsize, 1065M framework vsize, 0(0) swapins, 0(0) swapouts.

<table>
<thead>
<tr>
<th>PID</th>
<th>COMMAND</th>
<th>%CPU</th>
<th>TIME</th>
<th>#TH</th>
<th>#WQ</th>
<th>#PORT</th>
<th>MEM</th>
<th>PURG</th>
<th>CMPRS</th>
<th>PGRP</th>
<th>PPID</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>80099</td>
<td>top</td>
<td>4.2</td>
<td>00:00.46</td>
<td>1/1</td>
<td>0</td>
<td>19</td>
<td>2752K</td>
<td>0B</td>
<td>0B</td>
<td>80099</td>
<td>78189</td>
<td>running</td>
</tr>
<tr>
<td>80067</td>
<td>QuickLookSat</td>
<td>0.0</td>
<td>00:00.29</td>
<td>3</td>
<td>1</td>
<td>42</td>
<td>11M</td>
<td>0B</td>
<td>0B</td>
<td>80067</td>
<td>1</td>
<td>stuck</td>
</tr>
<tr>
<td>80066</td>
<td>quicklookd</td>
<td>0.0</td>
<td>00:00.22</td>
<td>4</td>
<td>0</td>
<td>86</td>
<td>8404K</td>
<td>0B</td>
<td>0B</td>
<td>80066</td>
<td>1</td>
<td>sleeping</td>
</tr>
<tr>
<td>79965</td>
<td>mdworker</td>
<td>0.0</td>
<td>00:00.62</td>
<td>25</td>
<td>22</td>
<td>93</td>
<td>8548K</td>
<td>0B</td>
<td>0B</td>
<td>79965</td>
<td>1</td>
<td>sleeping</td>
</tr>
<tr>
<td>79960</td>
<td>mdworker</td>
<td>0.0</td>
<td>00:00.04</td>
<td>3</td>
<td>0</td>
<td>50</td>
<td>1520K</td>
<td>0B</td>
<td>0B</td>
<td>79960</td>
<td>1</td>
<td>sleeping</td>
</tr>
<tr>
<td>79953</td>
<td>com.apple.iC</td>
<td>0.0</td>
<td>00:00.13</td>
<td>2</td>
<td>0</td>
<td>45</td>
<td>1724K</td>
<td>0B</td>
<td>0B</td>
<td>79953</td>
<td>1</td>
<td>sleeping</td>
</tr>
<tr>
<td>78189</td>
<td>bash</td>
<td>0.0</td>
<td>00:00.12</td>
<td>1</td>
<td>0</td>
<td>15</td>
<td>620K</td>
<td>0B</td>
<td>0B</td>
<td>78189</td>
<td>77854</td>
<td>sleeping</td>
</tr>
<tr>
<td>77897</td>
<td>printtool</td>
<td>0.0</td>
<td>00:00.02</td>
<td>2</td>
<td>1</td>
<td>29</td>
<td>984K</td>
<td>0B</td>
<td>0B</td>
<td>77897</td>
<td>1</td>
<td>sleeping</td>
</tr>
<tr>
<td>77854</td>
<td>login</td>
<td>0.0</td>
<td>00:00.04</td>
<td>2</td>
<td>0</td>
<td>26</td>
<td>932K</td>
<td>0B</td>
<td>0B</td>
<td>77854</td>
<td>419</td>
<td>sleeping</td>
</tr>
<tr>
<td>4564</td>
<td>Google Chrom</td>
<td>0.0</td>
<td>00:03.28</td>
<td>8</td>
<td>0</td>
<td>98</td>
<td>32M</td>
<td>0B</td>
<td>0B</td>
<td>3501</td>
<td>3501</td>
<td>sleeping</td>
</tr>
<tr>
<td>4532</td>
<td>Google Chrom</td>
<td>7.9</td>
<td>14:00.95</td>
<td>25</td>
<td>0</td>
<td>72</td>
<td>76M</td>
<td>0B</td>
<td>0B</td>
<td>3501</td>
<td>3501</td>
<td>sleeping</td>
</tr>
<tr>
<td>4529</td>
<td>Google Chrom</td>
<td>3.0</td>
<td>03:56.60</td>
<td>10</td>
<td>0</td>
<td>100</td>
<td>101M</td>
<td>0B</td>
<td>0B</td>
<td>3501</td>
<td>3501</td>
<td>sleeping</td>
</tr>
<tr>
<td>4502-</td>
<td>crash-catche</td>
<td>0.0</td>
<td>00:00.02</td>
<td>2</td>
<td>0</td>
<td>34</td>
<td>1160K</td>
<td>0B</td>
<td>0B</td>
<td>4502</td>
<td>4499</td>
<td>sleeping</td>
</tr>
<tr>
<td>4499-</td>
<td>TextWrangler</td>
<td>0.0</td>
<td>00:50.42</td>
<td>3</td>
<td>0</td>
<td>291</td>
<td>44M</td>
<td>92K</td>
<td>0B</td>
<td>4499</td>
<td>1</td>
<td>sleeping</td>
</tr>
<tr>
<td>4442</td>
<td>DashboardCli</td>
<td>0.0</td>
<td>00:10.15</td>
<td>8</td>
<td>0</td>
<td>277</td>
<td>100M</td>
<td>3608K</td>
<td>0B</td>
<td>420</td>
<td>420</td>
<td>sleeping</td>
</tr>
</tbody>
</table>
NAME
mv -- move files

SYNOPSIS
mv [-f | -i | -n] [-v] source target
mv [-f | -i | -n] [-v] source ... directory

DESCRIPTION
In its first form, the mv utility renames the file named by the source operand to the destination path named by the target operand. This form is assumed when the last operand does not name an already existing directory.

In its second form, mv moves each file named by a source operand to a destination file in the existing directory named by the directory operand. The destination path for each operand is the pathname produced by the concatenation of the last operand, a slash, and the final pathname component of the named file.
mv

- `mv <source> <destination>`
- Move a file to a new location or rename
#Create a test file
$ touch myfile
$ ls myfile

#Simple file move
$ mv myfile yourfile
$ ls myfile
$ ls yourfile
yourfile

#Move From One Directory To Another
$ mkdir testdir1
$ mkdir testdir2
$ touch testdir1/myfile
$ mv testdir1/myfile testdir2/myfile
NAME
rsync - faster, flexible replacement for rcp

DESCRIPTION
rsync is a program that behaves in much the same way that rcp does, but has many more options and uses the rsync remote-update protocol to greatly speed up file transfers when the destination file is being updated.

The rsync remote-update protocol allows rsync to transfer just the differences between two sets of files across the network connection, using an efficient checksum-search algorithm described in the technical report that accompanies this package.
rsync

- `rsync <options> <source> <destination>`
- Used to copy a directory tree from one location to another.
- Can be run multiple times to pick up changes
- Great for migrating user data or filesystems.
# Copy your downloads folder to /tmp
$ rsync -av ~/Downloads/ /tmp/

# Backup your home folder to /tmp
$ rsync -av $HOME /tmp
What We Covered

- open
- diskutil
- ls
- chmod
- chown
- du
- df
- find
- grep
- top
- mv
- rsync
We Also Talked About

- man
- . / .. / ~
- rm
- sort
- touch
- mkdir
- $HOME
- $USER
- Regular Expressions
- Pipe "|"
- git
### So Much More

- `cd / pwd`
- `env`
- `sed / awk`
- `mkdir / rmdir / ln`
- `who / finger`
- `shutdown / uptime`
- `more / less / cat / head`
- `kill`
- `uname`
- `pkgutil`
- `system_profiler`
- `mount`
- `ifconfig / netstat`
- `wc`
Thanks!