The 12 Unix Commands Everyone Should Know

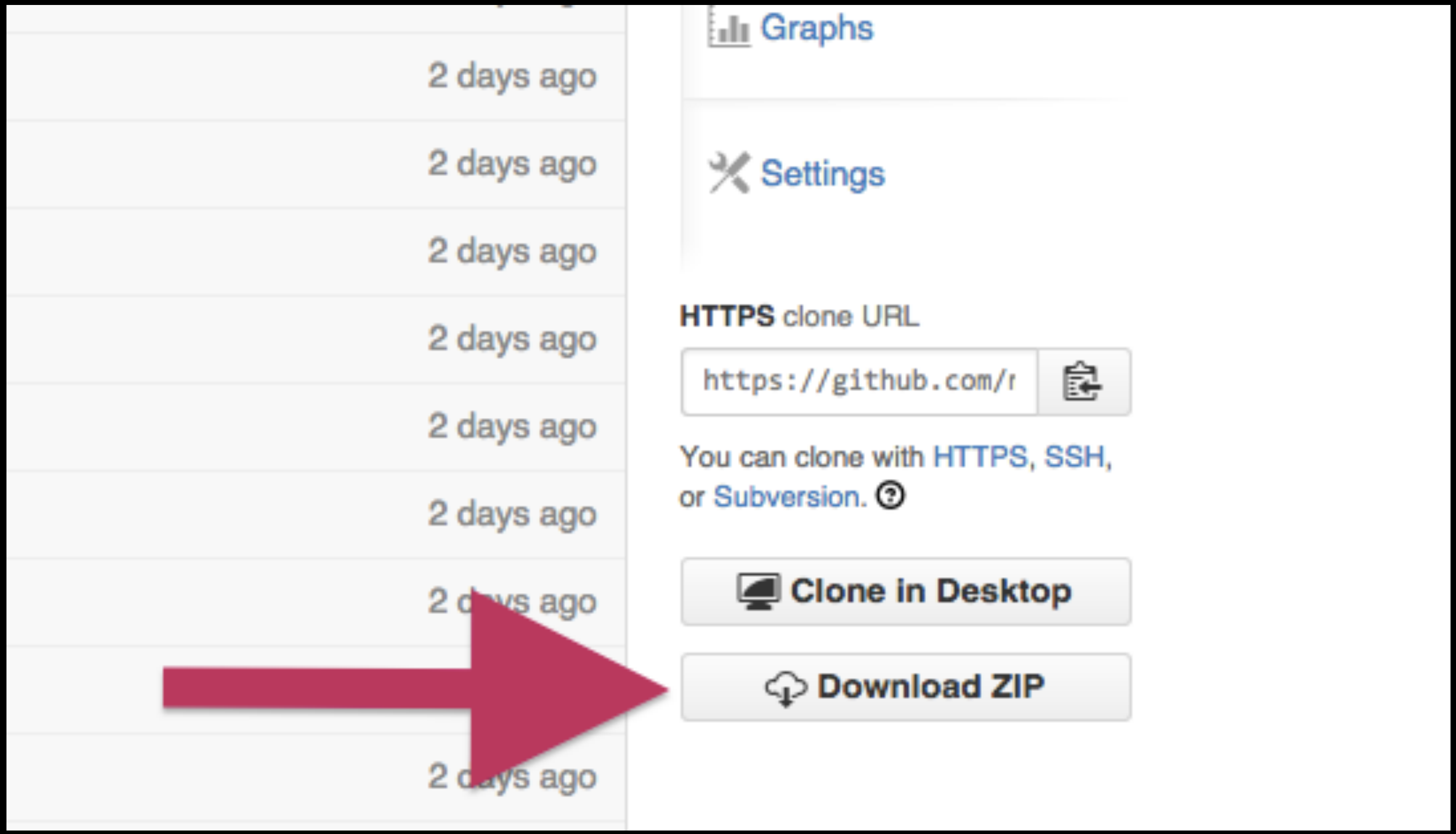
Matt Schnittker
@mschnitt

What We'll Cover

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

Code samples

https://github.com/mschnitt/psu_2015



HTTPS clone URL

https://github.com/r

You can clone with HTTPS, SSH, or Subversion.

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
     open [-e] [-t] [-f] [-F] [-W] [-R] [-n] [-g] [-h] [-b bundle_identifier]
          [-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

diskutil

```
$ man diskutil
DISKUTIL(8)              BSD System Manager's Manual             DISKUTIL(8)

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

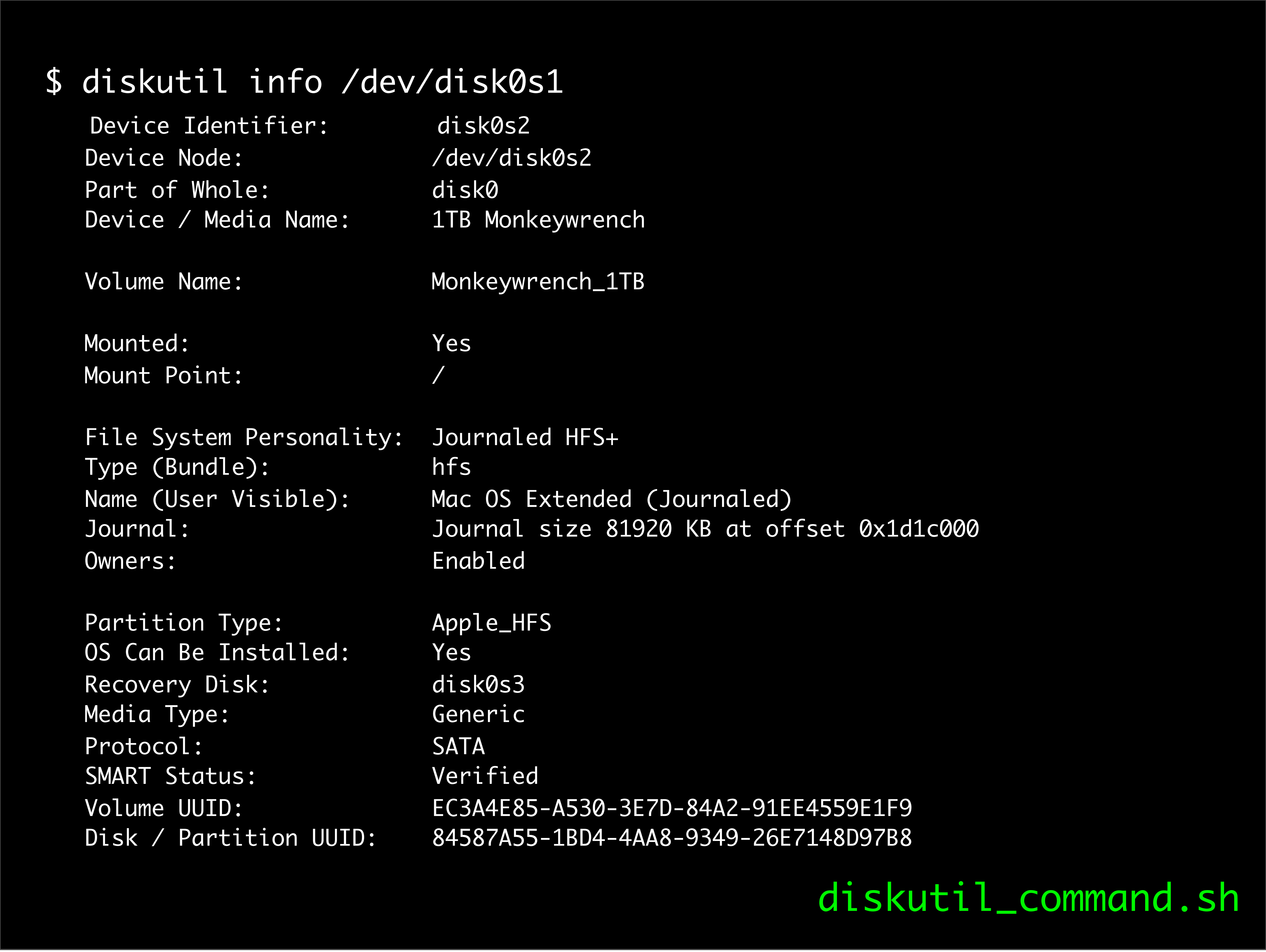
- Command line version of disk utility
- Handy tool for quick disk operations or scripting up disk partitioning.

```
$ 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_command.sh

```
$ 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_command.sh



```
$ diskutil info /dev/disk0s1
    Device Identifier:        disk0s2
    Device Node:              /dev/disk0s2
    Part of Whole:            disk0
    Device / Media Name:      1TB Monkeywrench

    Volume Name:              Monkeywrench_1TB

    Mounted:                  Yes
    Mount Point:              /

    File System Personality:  Journaled HFS+
    Type (Bundle):            hfs
    Name (User Visible):      Mac OS Extended (Journaled)
    Journal:                  Journal size 81920 KB at offset 0x1d1c000
    Owners:                   Enabled

    Partition Type:           Apple_HFS
    OS Can Be Installed:      Yes
    Recovery Disk:            disk0s3
    Media Type:               Generic
    Protocol:                 SATA
    SMART Status:             Verified
    Volume UUID:              EC3A4E85-A530-3E7D-84A2-91EE4559E1F9
    Disk / Partition UUID:    84587A55-1BD4-4AA8-9349-26E7148D97B8
```

diskutil_command.sh

```
$ diskutil
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 …
```

diskutil_command.sh

ls

```
$ man ls
LS(1)                    BSD General Commands Manual                   LS(1)

NAME
     ls -- list directory contents

SYNOPSIS
     ls [-ABCFGHLOPRSTUW@abcdefghiklmnopqrstuwx1] [file ...]

DESCRIPTION
     For each operand that names a file of a type other than directory, ls dis-
     plays 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 infor-
     mation.

     If no operands are given, the contents of the current directory are dis-
     played.  If more than one operand is given, non-directory operands are dis-
     played 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*
```

ls_command.sh

```
#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]
```

ls_command.sh

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.

chmod

```
$ man chmod
CHMOD(1)                 BSD General Commands Manual                 CHMOD(1)

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 speci-
     fied 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

	Read 4	Write 2	Execute 1	Total
Execute Only	0	0	1	1
Write Only	0	1	0	2
Read Only	1	0	0	4
Read + Execute	1	0	1	5
Everything	1	1	1	7

```
#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
```

chmod_chown_commands.sh

```
#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
```

chmod_chown_commands.sh

chown

```
$ man chown
CHOWN(8)                  BSD System Manager's Manual                 CHOWN(8)

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
```

chmod_chown_commands.sh

du

```
$ man du
DU(1)                     BSD General Commands Manual                    DU(1)

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 argu-
     ment.  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/
```

du_df_commands.sh

df

```
$ man df
DF(1)                     BSD General Commands Manual                    DF(1)

NAME
     df -- display free disk space

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

LEGACY SYNOPSIS
     df [-b | -h | -H | -k | -m | -P] [-ailn] [-t type] [-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 filesys-
     tem operand is specified, statistics for all mounted filesystems are dis-
     played (subject to the -t option below).
```

df

- df <options> <path>
- Gives free space on disk
- Usually runs quickly.

```
#$ 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
```

du_df_commands.sh

find

```
$ 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>
```

find_command.sh

```
#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
```

find_command.sh

```
#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

grep

```
$ man grep
GREP(1)                   BSD General Commands Manual                  GREP(1)

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

SYNOPSIS
     grep [-abcdDEFGHhIiJLlmnOopqRSsUVvwxZ] [-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-
     terns 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>

#Grep through multiple things at once
$ grep -i version /Applications/*.app/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
```

grep_command.sh

top

```
$ man top
op(1)                                                                     top(1)

NAME
       top - display and update sorted information about processes

DESCRIPTION
       The  top  program  periodically displays a sorted list of system processes.  The de-
       fault sorting key is pid, but other keys can be used instead.   Various  output  op-
       tions 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  specifica-
       tions  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.
Networks: packets: 9901086/13G in, 2549058/198M out. Disks: 1014451/24G read, 662307/35G
written.

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

top_command.sh

mv

```
$ man mv
MV(1)                     BSD General Commands Manual                    MV(1)

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 oper-
     and 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
```

mv_command.sh

rsync

```
$ man rsync
 sync(1)                                                        rsync(1)

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 differ-
       ences 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
```

rsync_command.sh

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!