BASH Primer

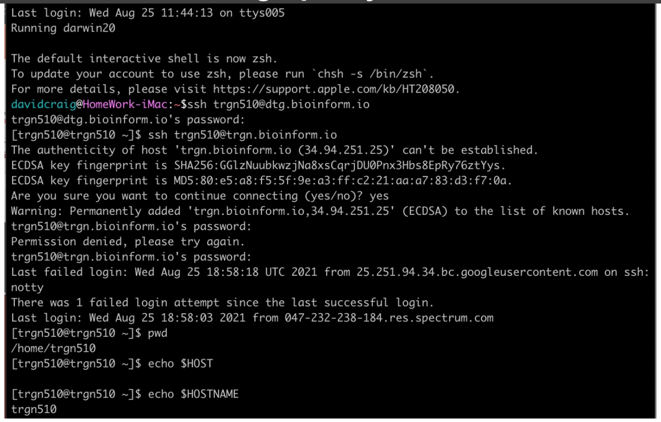
David W. Craig

USC University of Southern California

USC Institute Of Translational Genomics
Keck Medicine of USC

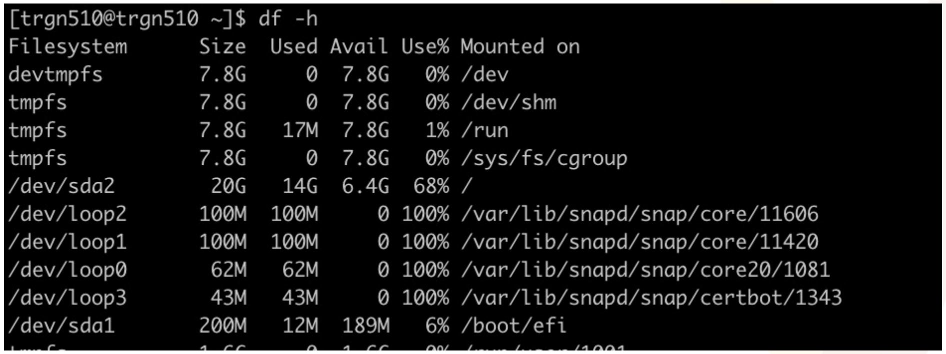
Logging in

Logging into server, through proxy server



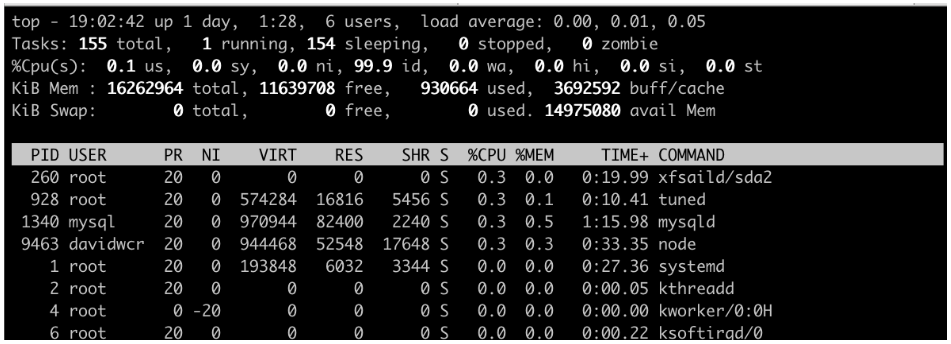
```
Last login: Wed Aug 25 11:44:13 on ttys005
Running darwin20

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
davidcraig@HomeWork-iMac:~$ssh trgn510@dtg.bioinform.io
trgn510@dtg.bioinform.io's password:
[trgn510@trgn510 ~]$ ssh trgn510@trgn.bioinform.io
The authenticity of host 'trgn.bioinform.io (34.94.251.25)' can't be established.
ECDSA key fingerprint is SHA256:GGlzNuubkwzjNa8xsCqrjDU0Pnx3Hbs8EpRy76ztYys.
ECDSA key fingerprint is MD5:80:e5:a8:f5:5f:9e:a3:ff:c2:21:aa:a7:83:d3:f7:0a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'trgn.bioinform.io,34.94.251.25' (ECDSA) to the list of known hosts.
trgn510@trgn.bioinform.io's password:
Permission denied, please try again.
trgn510@trgn.bioinform.io's password:
Last failed login: Wed Aug 25 18:58:18 UTC 2021 from 25.251.94.34.bc.googleusercontent.com on ssh:
notty
There was 1 failed login attempt since the last successful login.
Last login: Wed Aug 25 18:58:03 2021 from 047-232-238-184.res.spectrum.com
[trgn510@trgn510 ~]$ pwd
/home/trgn510
[trgn510@trgn510 ~]$ echo $HOST

[trgn510@trgn510 ~]$ echo $HOSTNAME
trgn510
```

Disk storage



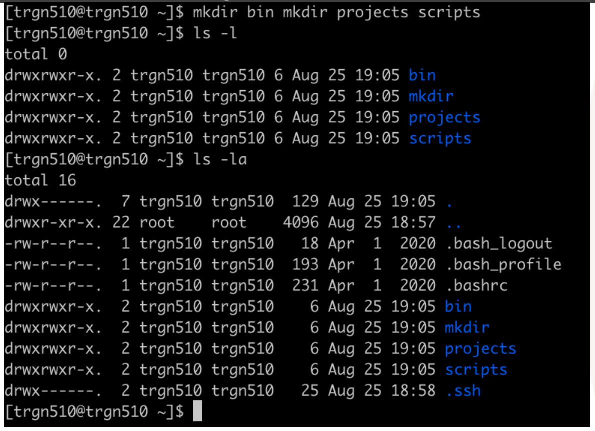
```
[trgn510@trgn510 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        7.8G     0  7.8G   0% /dev
tmpfs           7.8G     0  7.8G   0% /dev/shm
tmpfs           7.8G   17M  7.8G   1% /run
tmpfs           7.8G     0  7.8G   0% /sys/fs/cgroup
/dev/sda2        20G   14G  6.4G  68% /
/dev/loop2      100M  100M     0 100% /var/lib/snapd/snap/core/11606
/dev/loop1      100M  100M     0 100% /var/lib/snapd/snap/core/11420
/dev/loop0       62M   62M     0 100% /var/lib/snapd/snap/core20/1081
/dev/loop3       43M   43M     0 100% /var/lib/snapd/snap/certbot/1343
/dev/sda1       200M   12M  189M   6% /boot/efi
tmpfs           1.6G     0  1.6G   0% /run/user/1001
```

top



```
top - 19:02:42 up 1 day,  1:28,  6 users,  load average: 0.00, 0.01, 0.05
Tasks: 155 total,   1 running, 154 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.0 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 16262964 total, 11639708 free,   930664 used,  3692592 buff/cache
KiB Swap:        0 total,        0 free,        0 used. 14975080 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
  260 root      20   0       0      0      0 S   0.3  0.0   0:19.99 xfsaild/sda2
  928 root      20   0  574284  16816   5456 S   0.3  0.1   0:10.41 tuned
 1340 mysql     20   0  970944  82400   2240 S   0.3  0.5   1:15.98 mysqld
 9463 davidwcr  20   0  944468  52548  17648 S   0.3  0.3   0:33.35 node
    1 root      20   0  193848   6032   3344 S   0.0  0.0   0:27.36 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.05 kthreadd
    4 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0       0      0      0 S   0.0  0.0   0:00.22 ksoftirqd/0
```

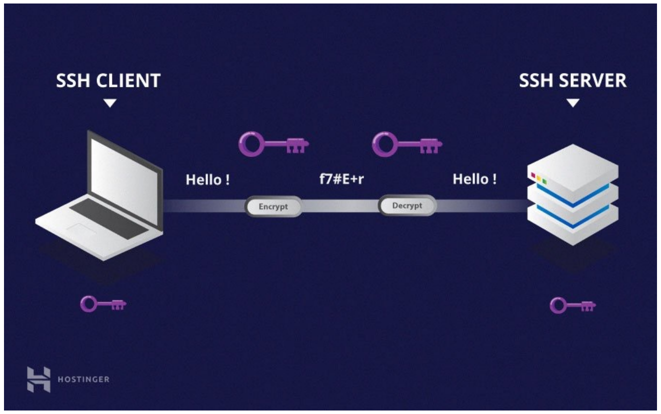
Making Directories, Listing



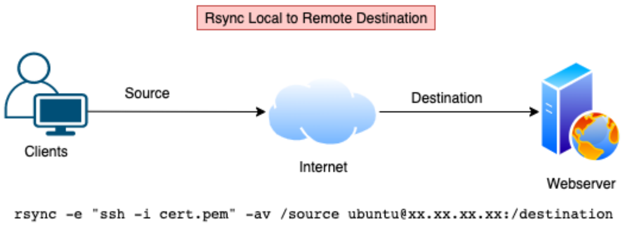
```
[trgn510@trgn510 ~]$ mkdir bin mkdir projects scripts
[trgn510@trgn510 ~]$ ls -l
total 0
drwxrwxr-x. 2 trgn510 trgn510 6 Aug 25 19:05 bin
drwxrwxr-x. 2 trgn510 trgn510 6 Aug 25 19:05 mkdir
drwxrwxr-x. 2 trgn510 trgn510 6 Aug 25 19:05 projects
drwxrwxr-x. 2 trgn510 trgn510 6 Aug 25 19:05 scripts
[trgn510@trgn510 ~]$ ls -la
total 16
drwx------.  7 trgn510 trgn510  129 Aug 25 19:05 .
drwxr-xr-x. 22 root    root    4096 Aug 25 18:57 ..
-rw-r--r--.  1 trgn510 trgn510   18 Apr  1  2020 .bash_logout
-rw-r--r--.  1 trgn510 trgn510  193 Apr  1  2020 .bash_profile
-rw-r--r--.  1 trgn510 trgn510  231 Apr  1  2020 .bashrc
drwxrwxr-x.  2 trgn510 trgn510    6 Aug 25 19:05 bin
drwxrwxr-x.  2 trgn510 trgn510    6 Aug 25 19:05 mkdir
drwxrwxr-x.  2 trgn510 trgn510    6 Aug 25 19:05 projects
drwxrwxr-x.  2 trgn510 trgn510    6 Aug 25 19:05 scripts
drwx------.  2 trgn510 trgn510   25 Aug 25 18:58 .ssh
[trgn510@trgn510 ~]$
```

Servers: ssh and rsync

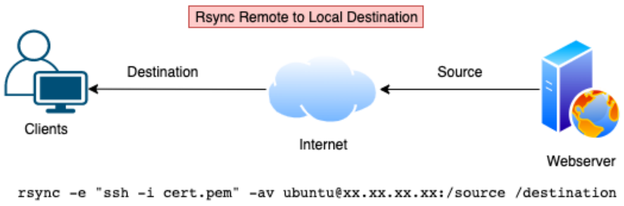
SSH Login to a Server



Rsync: Transfer files



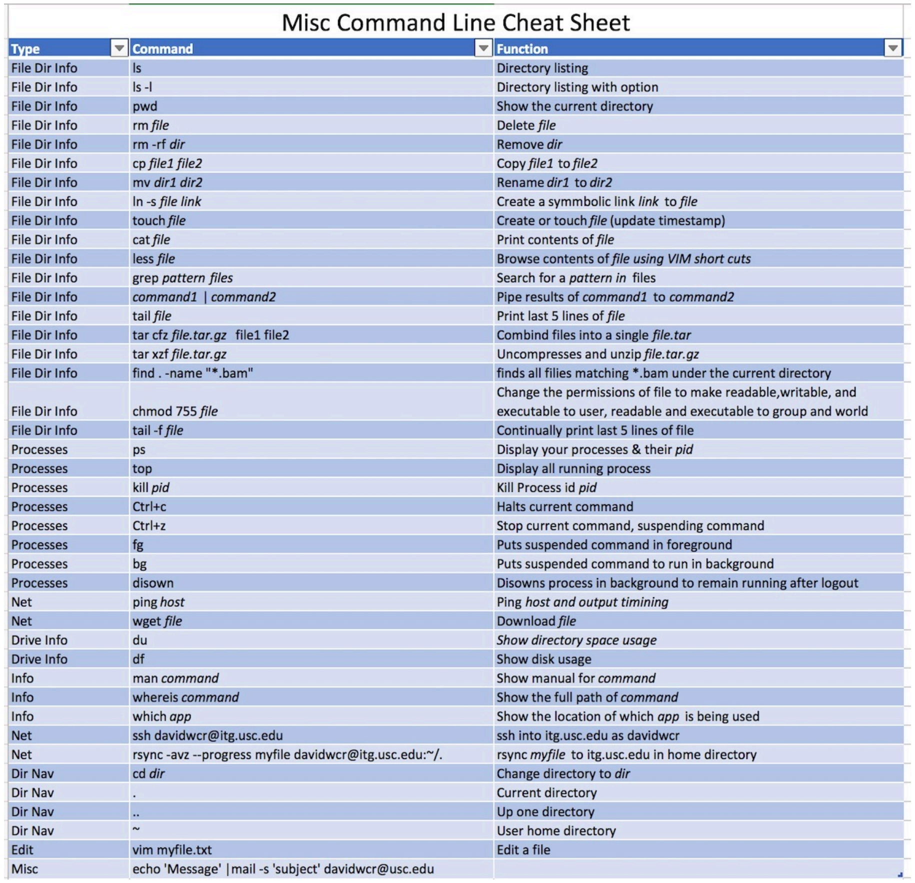
```
rsync -e "ssh -i cert.pem" -av /source ubuntu@xx.xx.xx.xx:/destination
```



```
rsync -e "ssh -i cert.pem" -av ubuntu@xx.xx.xx.xx:/source /destination
```

Credit: linux beast

Bash commands



Misc Command Line Cheat Sheet

Type	Command	Function
File Dir Info	ls	Directory listing
File Dir Info	ls -l	Directory listing with option
File Dir Info	pwd	Show the current directory
File Dir Info	rm *file*	Delete *file*
File Dir Info	rm -rf *dir*	Remove *dir*
File Dir Info	cp *file1 file2*	Copy *file1* to *file2*
File Dir Info	mv *dir1 dir2*	Rename *dir1* to *dir2*
File Dir Info	ln -s *file link*	Create a symmbolic link *link* to *file*
File Dir Info	touch *file*	Create or touch *file* (update timestamp)
File Dir Info	cat *file*	Print contents of *file*
File Dir Info	less *file*	Browse contents of *file using VIM short cuts*
File Dir Info	grep *pattern files*	Search for a *pattern in* files
File Dir Info	*command1* | *command2*	Pipe results of *command1* to *command2*
File Dir Info	tail *file*	Print last 5 lines of *file*
File Dir Info	tar cfz *file.tar.gz* file1 file2	Combind files into a single *file.tar*
File Dir Info	tar xzf *file.tar.gz*	Uncompresses and unzip *file.tar.gz*
File Dir Info	find . -name "*.bam"	finds all filies matching *.bam under the current directory
File Dir Info	chmod 755 *file*	Change the permissions of file to make readable,writable, and executable to user, readable and executable to group and world
File Dir Info	tail -f *file*	Continually print last 5 lines of file
Processes	ps	Display your processes & their *pid*
Processes	top	Display all running process
Processes	kill *pid*	Kill Process id *pid*
Processes	Ctrl+c	Halts current command
Processes	Ctrl+z	Stop current command, suspending command
Processes	fg	Puts suspended command in foreground
Processes	bg	Puts suspended command to run in background
Processes	disown	Disowns process in background to remain running after logout
Net	ping *host*	Ping *host and output timining*
Net	wget *file*	Download *file*
Drive Info	du	*Show directory space usage*
Drive Info	df	Show disk usage
Info	man *command*	Show manual for *command*
Info	whereis *command*	Show the full path of *command*
Info	which *app*	Show the location of which *app* is being used
Net	ssh davidwcr@itg.usc.edu	ssh into itg.usc.edu as davidwcr
Net	rsync -avz --progress myfile davidwcr@itg.usc.edu:~/.	rsync *myfile* to itg.usc.edu in home directory
Dir Nav	cd *dir*	Change directory to *dir*
Dir Nav	.	Current directory
Dir Nav	..	Up one directory
Dir Nav	~	User home directory
Edit	vim myfile.txt	Edit a file
Misc	echo 'Message' |mail -s 'subject' davidwcr@usc.edu	

Some keys

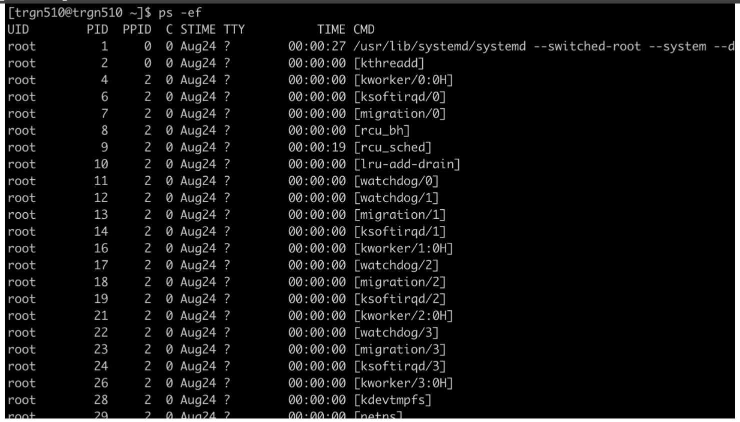
- Tilda = “~”
- Escape = “esc” key
- String = “$”
- Dot or period = “.”
- Amperstamp = “&”
- Caret = “^”
- Colon = “:”
- Dash = “-”
- Curly brackets = “{}”
- Square brackets = “[]”
- Parenthesis = “()”
- Redirect = “>”
- Pipe = “|”
- Up arrow = (↑ ↓ → ←) on your keyboard

Combo Bash commands

- Stop command = control C
- Previous command = ↑
- Finish path = “tab” key

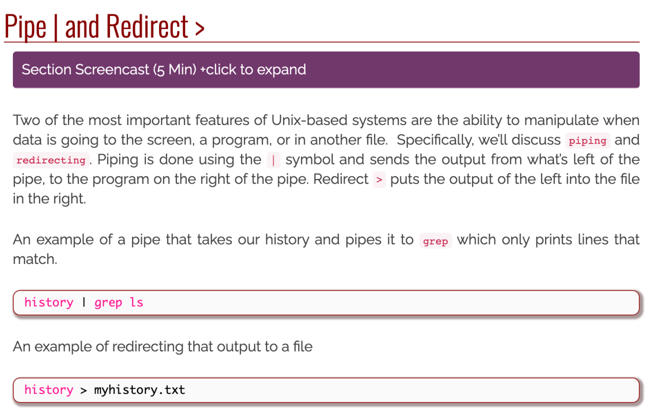
Pipes and Redirects

Example: ps



```
[trgn510@trgn510 ~]$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Aug24 ?        00:00:27 /usr/lib/systemd/systemd --switched-root --system --d
root         2     0  0 Aug24 ?        00:00:00 [kthreadd]
root         4     2  0 Aug24 ?        00:00:00 [kworker/0:0H]
root         6     2  0 Aug24 ?        00:00:00 [ksoftirqd/0]
root         7     2  0 Aug24 ?        00:00:00 [migration/0]
root         8     2  0 Aug24 ?        00:00:00 [rcu_bh]
root         9     2  0 Aug24 ?        00:00:19 [rcu_sched]
root        10     2  0 Aug24 ?        00:00:00 [lru-add-drain]
root        11     2  0 Aug24 ?        00:00:00 [watchdog/0]
root        12     2  0 Aug24 ?        00:00:00 [watchdog/1]
root        13     2  0 Aug24 ?        00:00:00 [migration/1]
root        14     2  0 Aug24 ?        00:00:00 [ksoftirqd/1]
root        16     2  0 Aug24 ?        00:00:00 [kworker/1:0H]
root        17     2  0 Aug24 ?        00:00:00 [watchdog/2]
root        18     2  0 Aug24 ?        00:00:00 [migration/2]
root        19     2  0 Aug24 ?        00:00:00 [ksoftirqd/2]
root        21     2  0 Aug24 ?        00:00:00 [kworker/2:0H]
root        22     2  0 Aug24 ?        00:00:00 [watchdog/3]
root        23     2  0 Aug24 ?        00:00:00 [migration/3]
root        24     2  0 Aug24 ?        00:00:00 [ksoftirqd/3]
root        26     2  0 Aug24 ?        00:00:00 [kworker/3:0H]
root        28     2  0 Aug24 ?        00:00:00 [kdevtmpfs]
root        29     2  0 Aug24 ?        00:00:00 [netns]
```

Pipes and Redirects



Pipe | and Redirect >

Two of the most important features of Unix-based systems are the ability to manipulate when data is going to the screen, a program, or in another file. Specifically, we'll discuss `piping` and `redirecting`. Piping is done using the `|` symbol and sends the output from what's left of the pipe, to the program on the right of the pipe. Redirect `>` puts the output of the left into the file in the right.

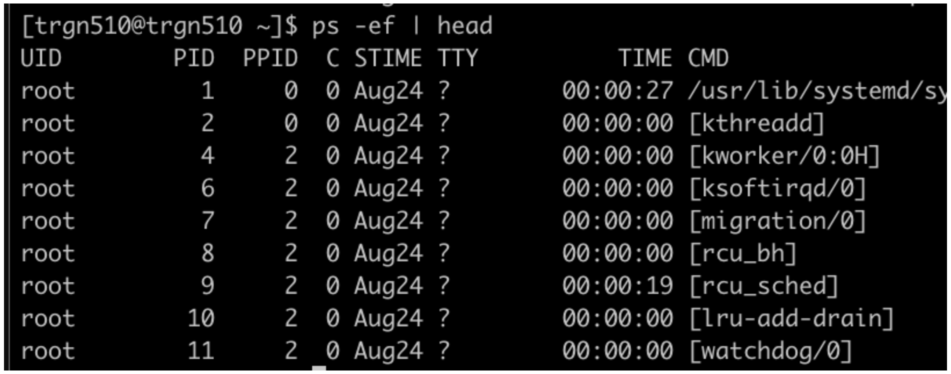
An example of a pipe that takes our history and pipes it to `grep` which only prints lines that match.

```
history | grep ls
```

An example of redirecting that output to a file

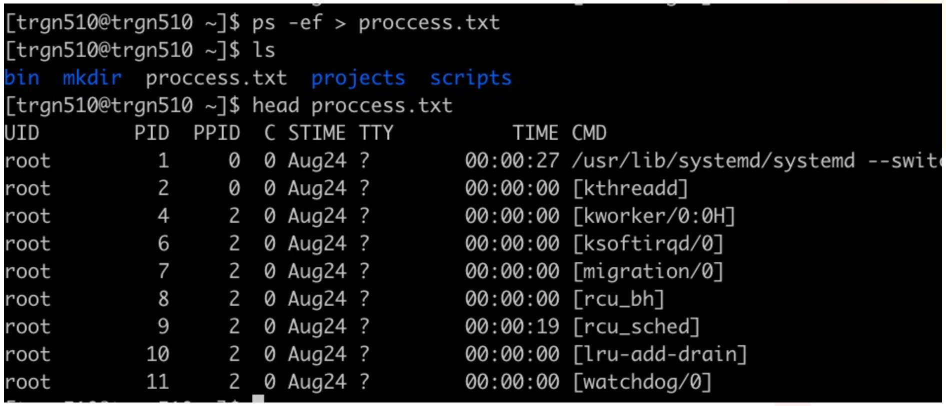
```
history > myhistory.txt
```

Example pipe



```
[trgn510@trgn510 ~]$ ps -ef | head
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Aug24 ?        00:00:27 /usr/lib/systemd/sy
root         2     0  0 Aug24 ?        00:00:00 [kthreadd]
root         4     2  0 Aug24 ?        00:00:00 [kworker/0:0H]
root         6     2  0 Aug24 ?        00:00:00 [ksoftirqd/0]
root         7     2  0 Aug24 ?        00:00:00 [migration/0]
root         8     2  0 Aug24 ?        00:00:00 [rcu_bh]
root         9     2  0 Aug24 ?        00:00:19 [rcu_sched]
root        10     2  0 Aug24 ?        00:00:00 [lru-add-drain]
root        11     2  0 Aug24 ?        00:00:00 [watchdog/0]
```

Example Redirect



```
[trgn510@trgn510 ~]$ ps -ef > proccess.txt
[trgn510@trgn510 ~]$ ls
bin  mkdir  proccess.txt  projects  scripts
[trgn510@trgn510 ~]$ head proccess.txt
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Aug24 ?        00:00:27 /usr/lib/systemd/systemd --swit
root         2     0  0 Aug24 ?        00:00:00 [kthreadd]
root         4     2  0 Aug24 ?        00:00:00 [kworker/0:0H]
root         6     2  0 Aug24 ?        00:00:00 [ksoftirqd/0]
root         7     2  0 Aug24 ?        00:00:00 [migration/0]
root         8     2  0 Aug24 ?        00:00:00 [rcu_bh]
root         9     2  0 Aug24 ?        00:00:19 [rcu_sched]
root        10     2  0 Aug24 ?        00:00:00 [lru-add-drain]
root        11     2  0 Aug24 ?        00:00:00 [watchdog/0]
```

Settings Files & Permissions

.bash_profile .bashrc

At login:

- The .bashrc and .bash_profile commands are run at login
- A file beginning with a period is hidden, to see it type `ls -la`
- Prompt is set; alias's are set, and your $PATH variable is set.
  - export PATH=$PATH:/place/with/the/file

'ls' gives files, directories and links for a path

- Files contain data, folders contain files and links, links look like a file but are actually paths to a file.

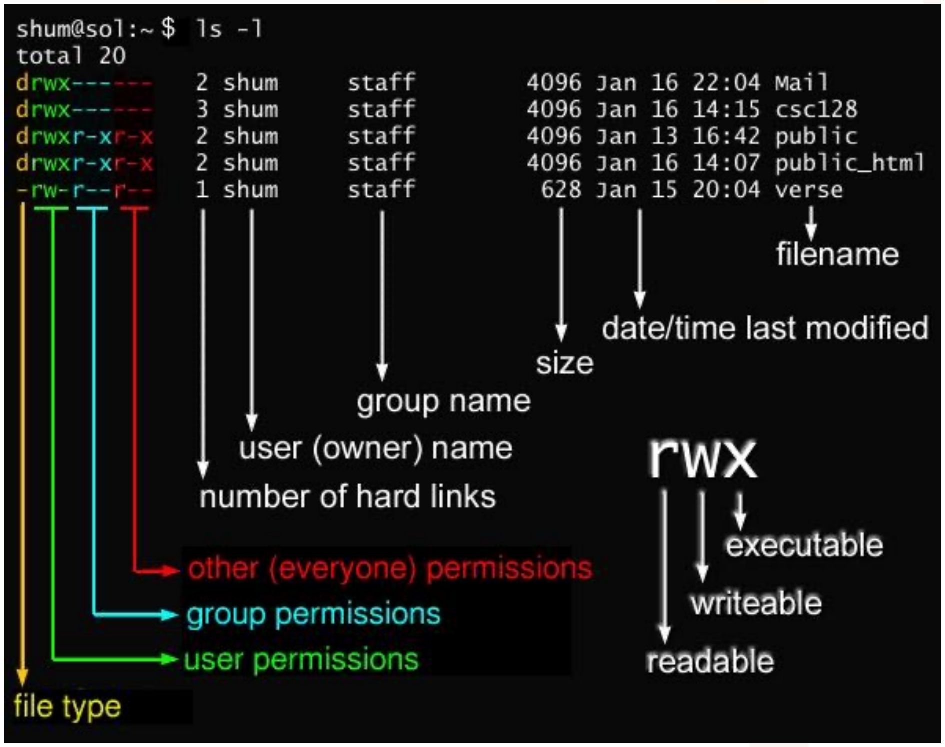
Typing a command runs a program

- For example, you can see the full path by typing `whereis` ls
- You do not need to type the full path because ls exists in a directory within the $PATH variable
- Commands typically have options indicated by a "-"
- For Example : `ls -la ~`

Permissions

- A file can be readable (+4), writable (+2) and executable (+1) to you the user, members in you group, or anyone with access to a machine.

Permissions



```
shum@sol:~$ ls -l
total 20
drwx------  2 shum   staff      4096 Jan 16 22:04 Mail
drwx------  3 shum   staff      4096 Jan 16 14:15 csc128
drwxr-xr-x  2 shum   staff      4096 Jan 13 16:42 public
drwxr-xr-x  2 shum   staff      4096 Jan 16 14:07 public_html
-rw-r--r--  1 shum   staff       628 Jan 15 20:04 verse
```

Pipe/Redirect

Pipe “|”

Pipe |

- Send the results of one program directly into another.

```
ps -ef | grep $USER
```

Redirect >

- Send the results into a file.

```
history > myhistory.txt
```

Append >>

- Append to the end of a file.

```
echo “By David” >> myhistory.txt
```

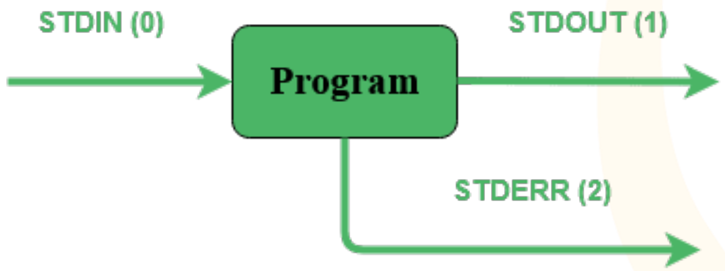
Stream STERR >&

- ONLY STDOUT

```
find . -name “*.sh” > found.txt
```

- Include STDERR
- `find . -name “*.sh” >& found.txt`

STDIN



Key Concepts: PATH

PATH : The path to a directory using '/'

The term "PATH"

- The "path" to a directory
- Your user path (or `~`): `/home/userid`
- To run an executable file, you must include the path, or the "path" must be in $PATH

The unix command 'pwd'

- Your current location of the BASH instance

Relative path.

- How to get to a directory, from your current location
- `.` : Current directory (equiv to `pwd`)
- `..` : Up a directory (`cd ..`)
- `~` : user home directory: `cd ~/bin`

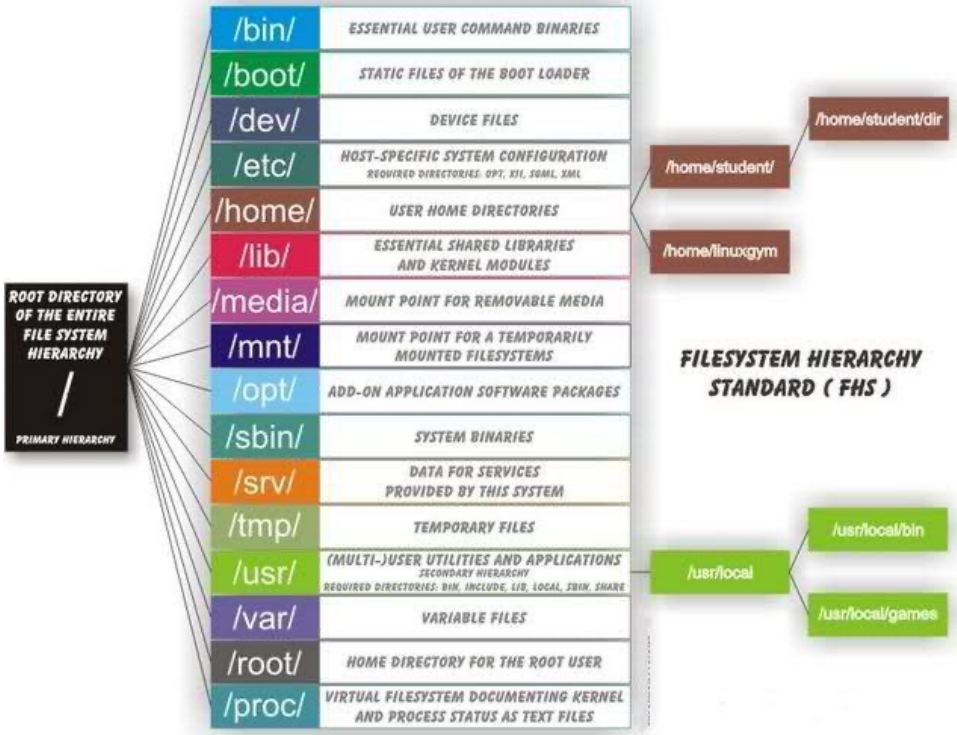
Absolute Path

- Starting from root: `cd /home/userid/projects/`

The environmental variable $PATH

- Directories checked for the program: `echo $PATH`
- export: `export PATH=$PATH:/place/with/the/file`

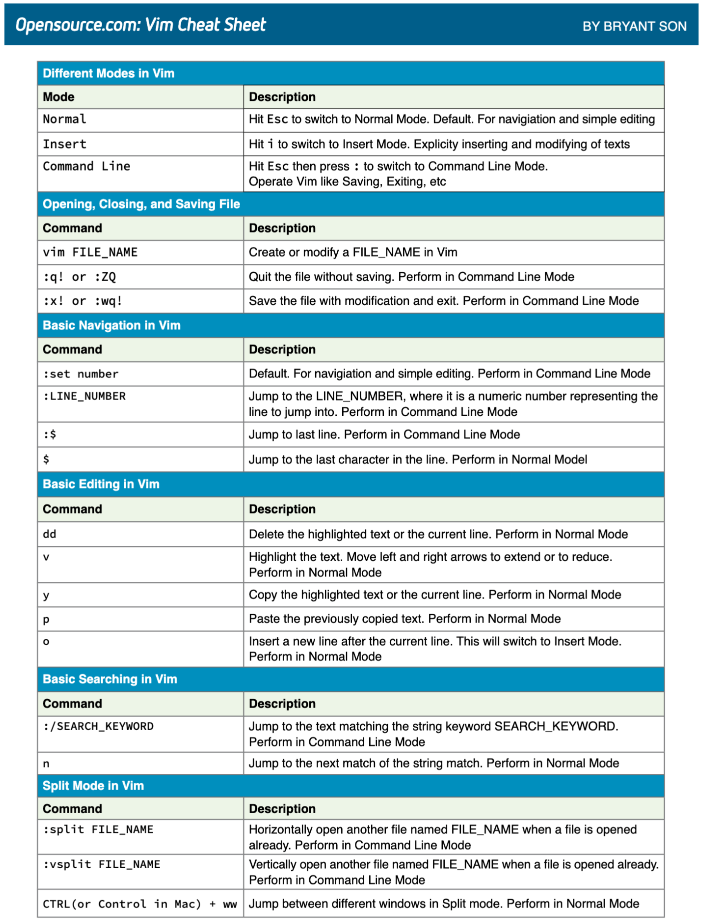
Typical Directory Tree



/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

Credit: tecmint

VIM: Creating & Editing textfiles on remote servers



Opensource.com: Vim Cheat Sheet BY BRYANT SON

Different Modes in Vim

Mode	Description
Normal	Hit Esc to switch to Normal Mode. Default. For navigiation and simple editing
Insert	Hit i to switch to Insert Mode. Explicity inserting and modifying of texts
Command Line	Hit Esc then press : to switch to Command Line Mode. Operate Vim like Saving, Exiting, etc

Opening, Closing, and Saving File

Command	Description
vim FILE_NAME	Create or modify a FILE_NAME in Vim
:q! or :ZQ	Quit the file without saving. Perform in Command Line Mode
:x! or :wq!	Save the file with modification and exit. Perform in Command Line Mode

Basic Navigation in Vim

Command	Description
:set number	Default. For navigiation and simple editing. Perform in Command Line Mode
:LINE_NUMBER	Jump to the LINE_NUMBER, where it is a numeric number representing the line to jump into. Perform in Command Line Mode
:$	Jump to last line. Perform in Command Line Mode
$	Jump to the last character in the line. Perform in Normal Model

Basic Editing in Vim

Command	Description
dd	Delete the highlighted text or the current line. Perform in Normal Mode
v	Highlight the text. Move left and right arrows to extend or to reduce. Perform in Normal Mode
y	Copy the highlighted text or the current line. Perform in Normal Mode
p	Paste the previously copied text. Perform in Normal Mode
o	Insert a new line after the current line. This will switch to Insert Mode. Perform in Normal Mode

Basic Searching in Vim

Command	Description
:/SEARCH_KEYWORD	Jump to the text matching the string keyword SEARCH_KEYWORD. Perform in Command Line Mode
n	Jump to the next match of the string match. Perform in Normal Mode

Split Mode in Vim

Command	Description
:split FILE_NAME	Horizontally open another file named FILE_NAME when a file is opened already. Perform in Command Line Mode
:vsplit FILE_NAME	Vertically open another file named FILE_NAME when a file is opened already. Perform in Command Line Mode
CTRL(or Control in Mac) + ww	Jump between different windows in Split mode. Perform in Normal Mode

- **Two modes**
  - Edit/insert mode (e.g. type ‘i’)
  - Normal/Navigate (Press `esc` key)
- **Save from Normal mode (:wq)**