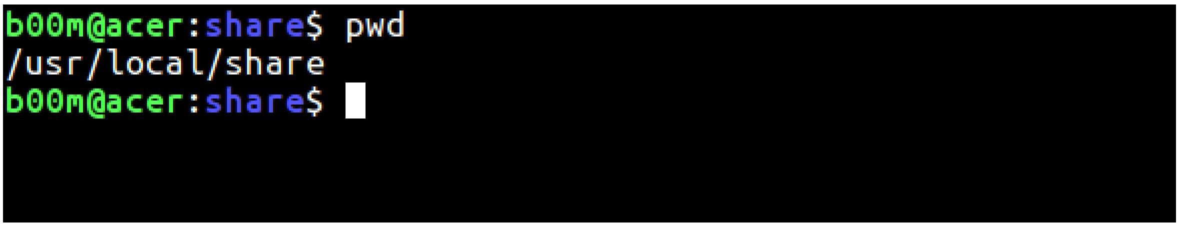
Basic Linux Commands – Part I

1. pwd command

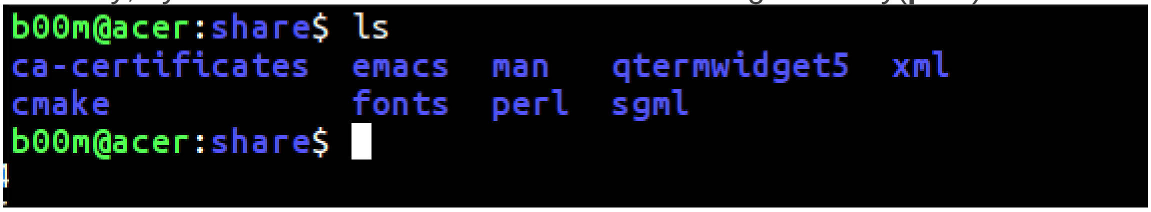
This command prints the location of your current working directory. It's important to know actually where you're before going to a parent or sub directories.



```
b00m@acer:share$ pwd
/usr/local/share
b00m@acer:share$ 
```

2. ls command

ls is one of the most used basic linux commands, used to **print** contents of a directory, by default it lists contents of current working directory(**pwd**).

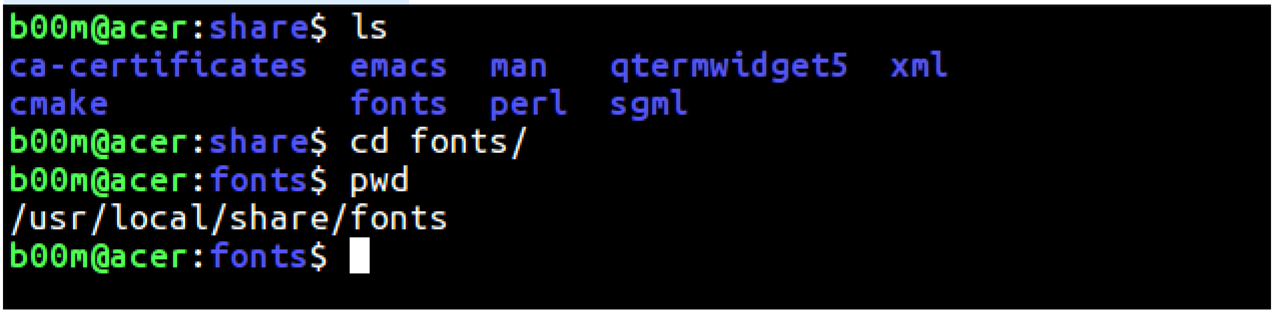


```
b00m@acer:share$ ls
ca-certificates  emacs  man   qtermwidget5  xml
cmake            fonts  perl  sgml
b00m@acer:share$ 
```

Example, use `ls /usr/bin` to list contents of the **/usr/bin** folder.

3. cd command

After knowing your **pwd** and getting an overview with the **ls**, it's time to move around with **cd** command. Clarification, assume you're on your **Home** directory, you need to go to the **/usr/local/share/fonts** directory, use `cd /usr/local/share/fonts`.



```
b00m@acer:share$ ls
ca-certificates  emacs  man   qtermwidget5  xml
cmake            fonts  perl  sgml
b00m@acer:share$ cd fonts/
b00m@acer:fonts$ pwd
/usr/local/share/fonts
b00m@acer:fonts$ 
```

There's three shortcut, if you need to move one directory up, use `cd ..` and go straight to your Home folder with `cd`, and use `cd -` to go back to your last working directory.

4. cat command

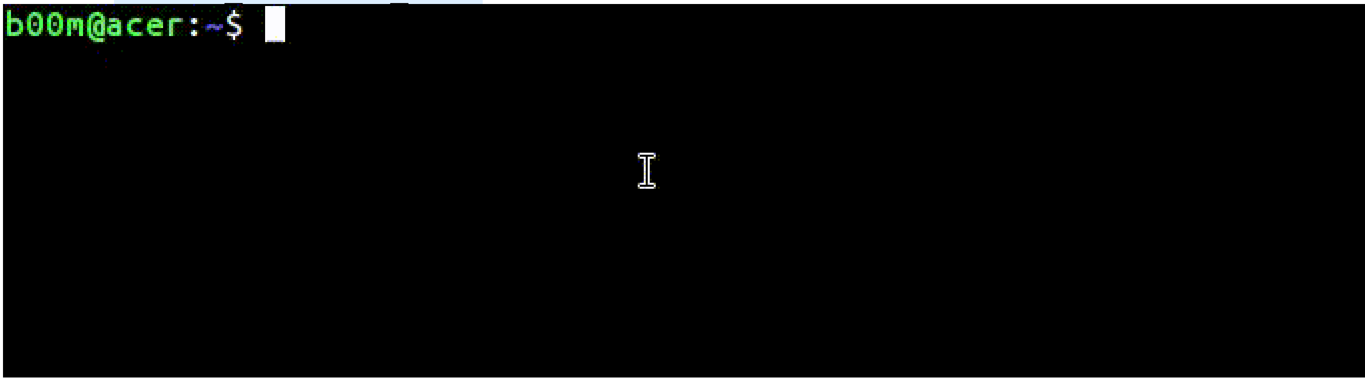
It's used to print the contents of a file to the screen(**stdout** more precisely), really useful when you want to have a quick look on contents of a file. As example, use `cat a_text_file` to get the inside contents of that file in your screen.

5. cp command

cp , You can copy files and directories with this command. Typical usage is like `cp file_a file_1_copy` or `cp directory_a dir_a_copy` Also don't forget to use proper path when you're coping something to different location.

6. mv command

The `mv` command is used to **move** or **rename** directories and files. To rename a file use `mv old_name new_name`, more details about mv **here** and **here**.



```
b00m@acer:~$
```

7. rm command

The `rm` command is used to remove directory or files. Like use `rm -r /tmp/backup` to remove everything that folder. Of course you've to be careful before removing anything.

8. mkdir command

mkdir, it's used to make a new directory in linux. Example, use `mkdir my_new_dir` to make a new directory named my_new_directory. The `-p` argument is useful, when you don't want to make parent directories manually.

9. rmdir command

rmdir, if you need to remove a directory, use this command. As example, use `rmdir my_dir` to remove that specific directory. More details about the rmdir command **here**.

10. touch command

touch, It's the equivalent command of mkdir for files. You can create a blank file with touch command. As example, use `touch ~/Public/index.html` to create a blank index.html file under the Public directory.

11. ln command

This command is used to make link between files and directories. As example, you need to make a symbolic link of the `/var/www` directory to the `/tmp` directory.

```
ln -s /var/www/ /tmp/
```

To un-link that symlink, use

```
unlink /tmp/www
```

You've to be extra careful with complete path and trailing slashes while linking and un-linking.

12. sudo command

sudo , that's an essential yet potentially dangerous command. Whenever you're getting a Permission denied, Authorization failed or something like that use sudo.

As example, the /var/www directory is not writable by the normal user. So to create a blank **index.html** file under the **/var/www** directory use `sudo touch /var/www/index.html`

13. head command

If you need to print first few lines of a file(any type) then you can use head command. A nice practical example w'd be

```
head -20 /var/log/syslog
```

This will print the first 20 lines of the **rsyslogd** log to the stdout. By default head command prints first 10 lines.

14. tail command

It's similar to the head command, but the function is opposite, prints last 10 lines of any file by default. Here's an example, how to print last 30 lines of the kernel log.

```
tail -30 /var/log/kern.log
```

15. chmod command

It's also a very important command, used to change file and directory permission. As the chmod command is a very long topic, so here I'll explain it in brief.

Basically there's three type of permission, read, write and execute. Each of them denoted by a number.

- 4 for **read** permission
- 2 for **write** permission
- 1 for **execute** permission

So if you need to set universal read/write permission to a file, you can use

```
chmod 666 my_file_name
```

Assume you need to make a script executable, you can use

```
chmod +x my_script_name
```

There'll be a full chmod tutorial very soon, to explain you in detail.

16. md5sum command

You may often need to check if a file tempered with or not. However md5sum is not the safest, but no doubt one of the most used.

An easy example could be finding the checksum of a ISO file

```
user@host:~$  md5sum  ~/OS/slitaz-5.0-rc3.iso
0d685551f8b0b0bd9caa3a4e66d61a3e  ~/OS/slitaz-5.0-rc3.iso
```

The long string of numbers and digits is the md5 hash of that particular file, just match first and last two characters, that's enough.

17. locate command

The basic command to find files and directories in Linux. As it's a database driven command, so for the first time you need to build the database, run `sudo updatedb` and wait for few minutes.

A typical example to locate something could be like below.

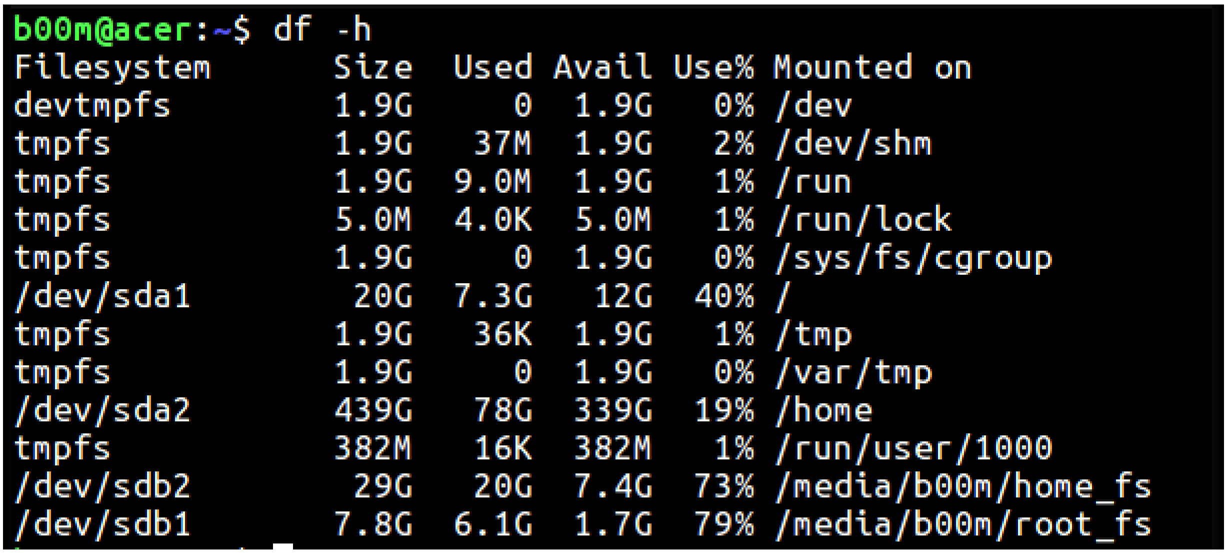
```
locate -i *chromium*
```

It supports wildcards, and use the `-i` option to ignore upper/lower case.

18. df command

This command is used to check disk space usage on a linux system. The most common usage is like below, used along with the `-h` flag.

```
df -h
```



```
b00m@acer:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        1.9G     0  1.9G   0% /dev
tmpfs           1.9G   37M  1.9G   2% /dev/shm
tmpfs           1.9G  9.0M  1.9G   1% /run
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           1.9G     0  1.9G   0% /sys/fs/cgroup
/dev/sda1        20G  7.3G   12G  40% /
tmpfs           1.9G   36K  1.9G   1% /tmp
tmpfs           1.9G     0  1.9G   0% /var/tmp
/dev/sda2       439G   78G  339G  19% /home
tmpfs           382M   16K  382M   1% /run/user/1000
/dev/sdb2        29G   20G  7.4G  73% /media/b00m/home_fs
/dev/sdb1       7.8G  6.1G  1.7G  79% /media/b00m/root_fs
```

19. du command

If you need to quickly check disk space usage of a file or directory, the `du` command is here.

For a single file, a nice example could be like below,

```
du -sh /boot/vmlinuz-4.10.10
```

Or could be like below for a entire directory and it's contents.

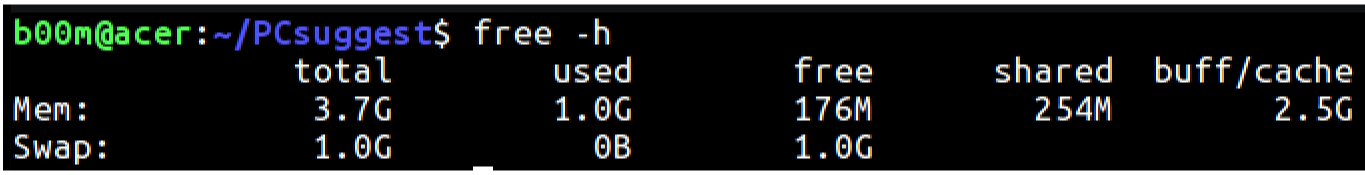
```
du -sh /opt/google/chrome/
```

The `-s` flag is used to suppress unnecessary clutter and `-h` flag is to make the output more human readable.

20. free command

The free command is used to display amount of free and used RAM in the system, also prints the swap space stats.

```
free -h
```



```
b00m@acer:~/PCsuggest$ free -h
              total        used        free      shared  buff/cache
Mem:           3.7G        1.0G        176M        254M        2.5G
Swap:          1.0G          0B        1.0G
```

Again, the `-h` flag is used to make the output easier to read by humans.You can read more here, check linux memory usage with command line tools.

21. zip command

No doubt you often need to create and extract zip archives, here's the `zip` and `unzip` commands for that.

Most probably these commands are not pre-installed, install them with apt in Ubuntu.

```
sudo apt-get install zip unzip
```

The syntax to create a zip archive,

```
zip -9r my_archive.zip  file_1 file_2 folder_1 folder_2 folder_3
```

When the `-9` option is used, zip attempts maximum compression on all files and `-r` option is for recursive archiving.

The `unzip` command extracts archives to the current working directory(**pwd**) by default. So if you need to extract the contents to a specific folder, then use

```
unzip my_archive.zip -d /path/to/my_directory
```

You might want to read about another archiving tool here, 7zip linux command examples.

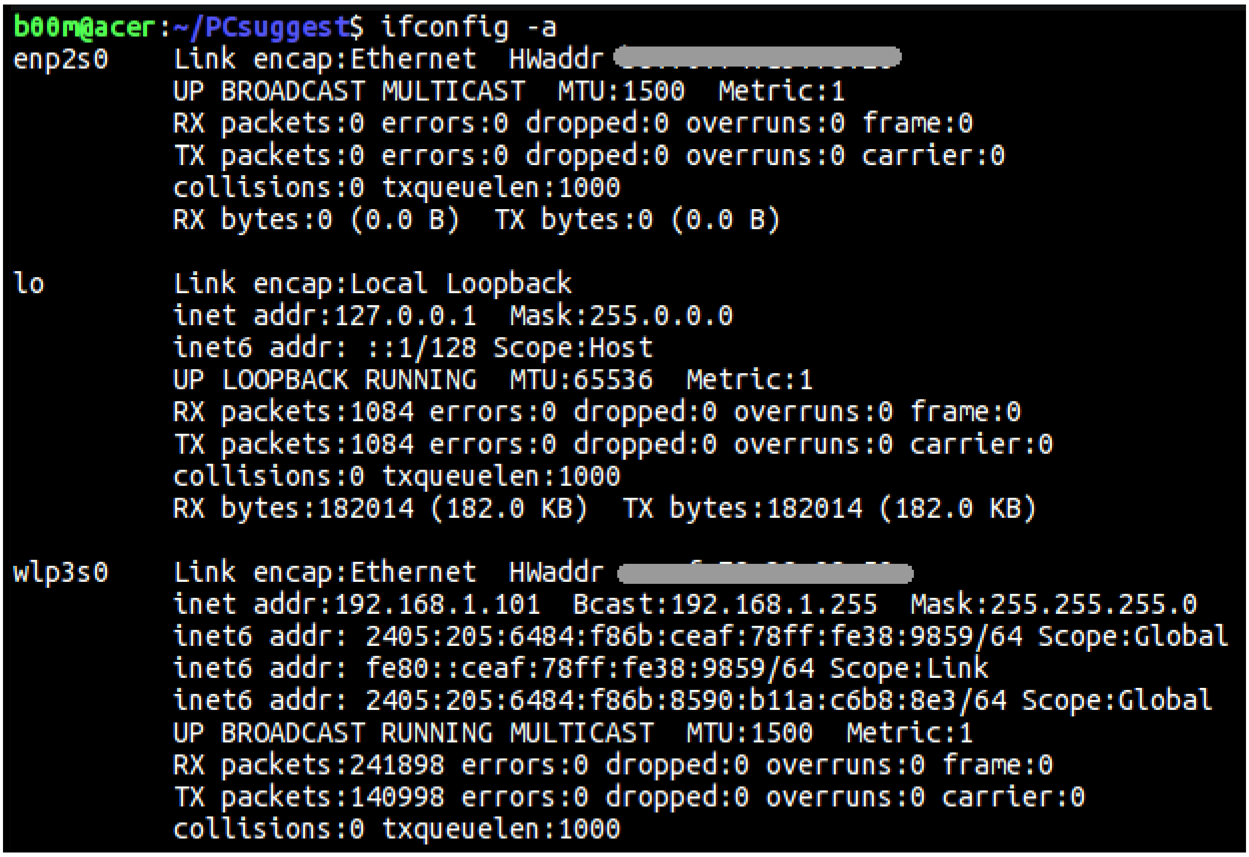
22. ifconfig commnad

`ifconfig` stands for **interface configuration**, and it can do many networking related things, literally.

Some basic use for beginners could be like checking which network interfaces are connected and their respective IP address.

Or you can find out how much data passed through a specific interface, all could be done just by running the ifconfig command.

```
ifconfig -a
```



```
b00m@acer:~/PCsuggest$ ifconfig -a
enp2s0    Link encap:Ethernet  HWaddr [redacted]
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:1084 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1084 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:182014 (182.0 KB)  TX bytes:182014 (182.0 KB)

wlp3s0    Link encap:Ethernet  HWaddr [redacted]
          inet addr:192.168.1.101  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 2405:205:6484:f86b:ceaf:78ff:fe38:9859/64 Scope:Global
          inet6 addr: fe80::ceaf:78ff:fe38:9859/64 Scope:Link
          inet6 addr: 2405:205:6484:f86b:8590:b11a:c6b8:8e3/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:241898 errors:0 dropped:0 overruns:0 frame:0
          TX packets:140998 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

23. uname command

This command prints some basic information about the system, like OS name, kernel version, host name, system time, OS architecture and so on.

```
uname -a
Linux acer 4.10.10 #2 SMP Mon Apr 24 00:48:20 IST 2017 x86_64
x86_64 x86_64 GNU/Linux
```

The above command prints everything it can, see the man page for more info.

24. history command

As the name suggests, `history` command prints a list of previously typed commands. Very useful when you're trying to find what you've done wrong before.

You can also quickly find previously typed commands by pressing the `Ctrl` + `R` key combo.

25. man command

The command to find details about other commands.

Almost every command has their respective man pages, useful to get a quick over view of an unknown command, use it like `man any_command` .

```
man ifconfig
```

Extra - Shutdown Linux machines with command line

We're not done yet! How you can shut down and reboot your system through command line?

- `shutdown -h now` to power off immediately.
- `shutdown -h +10` to shutdown after 10 minutes.
- `reboot` to reboot the machine immediately.

You may need to use `sudo` with the command above, depending on your current user id. Another thing important note is the modern `shutdown` command is a symbolic link to the **systemd** init daemon.