

Day 1

Linux Commands

"whoami" command

whoami command is used to print the loggedin user name.

```
whoami
```

"uname" command

1. Kernel name

To reveal the kernel name, you can use -s parameter.

```
# uname -s
```

The output will be same with uname without parameter.

2. Kernel release

If you need to know what kernel release you're using, just use -r parameter

```
# uname -r
```

3. Kernel version

Beside kernel information, uname can also fetch the kernel version. Use -v parameter for this purpose

```
# uname -v
```

4. Nodename

Parameter -n will give you the node hostname. For example, if your hostname is "dev-machine", -n parameter will print dev-machine as the output of -n parameter

```
# uname -n
```

For RedHat and CentOS, you can also use /etc/redhat_release file :

```
# cat /etc/redhat_release  
CentOS release 5.10 (Final)
```

For non-RedHat based distro, you may use /etc/issue. Here's the example :

```
# cat /etc/issue
Linux Mint Olivia \n \l
```

5. Hardware name

If you are wondering what kind of machine you're using, you can try `-m` parameter. It will show you information about it.

```
# uname -m
```

i686 indicates that your system is 32 bit operating system. While `x86_64` means your system is a 64 bit system.

6. Hardware platform

Similar with hardware name, `-i` parameter will show you hardware platform.

```
# uname -i
```

i386 means you are running a 32 bit system. If the output is `x86_64` it means that you are running 64 bit system.

7. Processor type

To see processor type, you can use `-p` parameter. If `uname` is not able to show you that information, it will show you 'unknown' in the output.

```
# uname -p
```

8. Operating system

`Uname` can also be used to reveal what operating system you are running. Use `-o` parameter to fulfill this purpose.

```
# uname -o
```

9. All information

There is one parameter that can reveal all information. It's `-a` parameter. It will show you all information except omit `-i` and `-p` if they are unknown.

```
# uname -a
Linux dev-machine 2.6.18-371.1.2.el5 #1 SMP Tue Oct 22 12:57:43
EDT 2013 i686 i686 i386 GNU/Linux
```

"uptime" command

1. Basic Usage

uptime tells you how long the system has been running.

```
$uptime  
15:23:43 up 6 days, 35 min, 3 users, load average: 0.44, 0.41, 0.41
```

2. -p option

show uptime in pretty format

```
$uptime -p  
up 6 days, 34 minutes
```

3. -s option

system up since, in yyyy-mm-dd HH:MM:SS format

```
$uptime -s  
2015-10-09 14:48:19
```

"useradd" command

1. Add new user usr1 with default settings

```
useradd usr1
```

2. Create the new user's home dir in /home

```
useradd usr1 -d /home/usr1
```

3. From 30/4/2009 the user acc will be disabled.

```
useradd usr1 -e 2009-04-30
```

4. After passwd expires, system will allow the user to login for 6 days with a warning to change his passwd.

```
useradd usr1 -f 6
```

5. Set user1 group as staff

```
useradd usr1 -g staff
```

6. Create the usr1 with the given encrypted password. For No passwd, acc disabled.

```
useradd usr1 -p $1$d8
```

"userdel" command

1. Delete the user account

userdel loginname

2. delete the user account together with user's home directory and all files inside it

userdel -r loginname

3. for the deletion even the user is still log in, it will force to delete the user's home directory and files as well, this option is dangerous use with caution

userdel -f loginname

"usermod" command

1. Create the new home Dir for usr1 in /home2 & Move old Dir contents to this Dir.

`usermod -d /home2/usr1 usr1`

2. From 30/4/2005 the usr1 acc will be disabled.

`usermod -e 2005-04-30 usr1`

3. After passwd expires, system will allow the user to login for 6 days with a warning to change his passwd.

`usermod -f 6 usr1`

4. Set usr1's initial group as prof.

`usermod -g prof usr1`

5. Set the new passwd for the usr1

`usermod -p 1d8 usr1`

6. Set Bash as the default login shell for the usr1.

`usermod -s /bin/bash usr1`

7. Lock a user's password.

`usermod -L usr1`

8. Unlock a user's password.

```
usermod -U usr1
```

"sleep" command

You need to use the sleep command to add delay for a specified amount of time. The syntax is as follows for gnu/bash sleep command:

```
sleep NUMBER[SUFFIX]
```

Where SUFFIX may be:

1. s for seconds (the default)
2. m for minutes.
3. h for hours.
4. d for days.

To sleep for 5 seconds, use:

```
sleep 5
```

To sleep for 2 minutes, use:

```
sleep 2m
```

To sleep for 3 hours, use:

```
sleep 3h
```

To sleep for 5 days, use:

```
sleep 5d
```

"logout" command

Use the command "logout" to exit a given session. If you have logged in, then typed "su" to become a superuser or another user, you may need to type "exit" until your SHLVL environment value is 1. Then you can type "logout" to exit your session. The "exit" command will take you back to previous shell levels.

"reboot" command

1. The following command will shutdown linux.

```
# reboot -p
```

The "p" options stands for poweroff.

2. To reboot linux just call the reboot command directly without any options.

```
# reboot
```

This will perform a graceful shutdown and restart of the machine. This is what happens when you click restart from your menu.

3. Reboot linux forcibly

The following command will forcefully reboot the machine. This is similar to pressing the power button of the CPU. No shutdown takes place. The system will reset instantly.

```
# reboot -f
```

“passwd” command

1. Change Password of System Users

When you logged in as non-root user like ‘tux’ in my case and run passwd command then it will reset password of logged in user.

```
$ passwd
Changing password for user tux
Changing password for tux.
(current) UNIX password:
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

When you logged in as root user and run **passwd** command then it will reset the root password by default and if you specify the user-name after passwd command then it will change the password of that user.

Note : System user’s password is stored in an encrypted form in /etc/shadow file.

2. Display Password Status Information.

To display password status information of a user , use **-S** option in passwd command.

```
# tux PS 2015-08-04 0 99999 7 -1 (Password set, SHA512 crypt.)
```

In the above output first field shows the user name and second field shows Password status (**PS** = **Password Set** , **LK** = **Password locked** , **NP** = **No**

Password), third field shows when the password was changed and last & fourth field shows minimum age, maximum age, warning period, and inactivity period for the password

3. Display Password Status info for all the accounts

To display password status info for all the accounts use **"-aS"** option in passwd command, example is shown below :

```
# passwd -Sa
```

4. Removing Password of a User using -d option

In my case i am removing/ deleting the password of 'tux' user.

```
# passwd -d tux
Removing password for user tux.
passwd: Success
```

"-d" option will make user's password empty and will disable user's account.

5. Set Password Expiry Immediately

Use **'-e'** option in passwd command to expire user's password immediately , this will force the user to change the password in the next login.

```
# passwd -e tux
Expiring password for user tux
passwd: Success
```

6. Lock the password of System User

Use **'-l'** option in passwd command to lock a user's password, it will add **"!"** at starting of user's password. A User can't Change it's password when his/her password is locked.

```
# passwd -l tux
Locking password for user tux.
passwd: Success
```

7. Unlock User's Password using -u option

```
# passwd -u tux
Unlocking password for user tux.
passwd: Success
```

8. Setting inactive days using -i option

-i option in passwd command is used to set inactive days for a system user. This will come into the picture when password of user (in my case tux) expired and user didn't change its password in '**n**' number of days (i.e 10 days in my case) then after that user will not able to login.

```
# passwd -i 10 tux
Adjusting aging data for user tux.
passwd: Success
```

9. Set Minimum Days to Change Password using -n option.

In the below example tux user has to change the password in 90 days. A value of zero shows that user can change it's password in any time.

```
# passwd -n 90 linuxtechi
Adjusting aging data for user linuxtechi.
passwd: Success
```

10. Set Warning days before password expire using -w option

'-w' option in passwd command is used to set warning days for a user. It means a user will be warned for n number of days that his/her password is going to expire.

```
# passwd -w 12 tux
Adjusting aging data for user tux
passwd: Success
```

"sudo" command

1. Set up sudo Environment in /etc/sudoers

You can provide sudo privilege to an individual user or a group by modifying /etc/sudoers.

sudo access to an user

To provide sudo access to an individual user, add the following line to the /etc/sudoers file.

```
tux    ALL=(ALL) ALL
```

In the above example:

- tux : name of user to be allowed to use sudo
- ALL : Allow sudo access from any terminal (any machine).

- (ALL) : Allow sudo command to be executed as any user.
- ALL : Allow all commands to be executed.

sudo access to a group

To provide sudo access to a group, add the following line to the /etc/sudoers file.

```
%programmers  ALL=(ALL) ALL
```

In the above example:

- programmers : name of group to be allowed to use sudo. Group name should be preceded with percentage symbol.
 - ALL : Allow sudo access from any terminal (any machine).
 - (ALL) : Allow sudo command to be executed as any user.
 - ALL : Allow all commands to be executed.

Note: Ubuntu users are already familiar with sudo command, as you'll use sudo apt-get install to install any package. On Ubuntu, sudo is already setup for your username as shown below. i.e All users who belong to admin group has access to execute root commands using sudo.

```
$ sudo cat /etc/sudoers
%admin ALL=(ALL) ALL
```

```
$ grep admin /etc/group
admin:x:115:sathiya
```

2. Executing a command as super user

Once the sudo access is provided to your account in /etc/sudoers, you can pass any root command as an argument to the sudo command. For example, mount can only be done by root. But, a normal user can do mount as shown below using sudo.

```
$ sudo mount /dev/sda3 /mnt
```

Note: If you are executing sudo for the first time in a shell it will ask for the password (current user password) by default.

3. Forgot to Use Sudo in Vim? No Worries. Save file Trick in vim with sudo

When you have opened a file that can be saved only by root user using vim (without using the sudo command), you can do the following.

For example, if you want to edit the file `/etc/group` that can only be saved by root user, you typically do the following. When you do a `:w`, no problem, it will work, as it was opened using `sudo` command.

```
$ sudo vim /etc/group
:w
```

What if you've forgot to give `sudo` when you've opened the `/etc/group` file as shown below? In this case, instead of coming out of the file (and losing all your changes) and executing the `vim` command with `sudo`, you can do the following.

```
$ vim /etc/group

:w !sudo tee %
```

Note: `":w !sudo tee %"` will save the file as root privilege, even if you didn't use `sudo` command to open it.

4. Forgot to give `sudo` for root command? Do it again using `!!`

If you've forgot to give `sudo` for a command that requires root privilege, instead of typing the command with `sudo` again, you can simply do `sudo !!` as shown below.

```
$ head -n 4 /etc/sudoers
head: cannot open `/etc/sudoers' for reading: Permission denied
```

```
$ sudo !!
sudo head -n 4 /etc/sudoers
# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
```

5. Get Root Shell Access using Sudo

To get a root shell from your user account, do the following.

```
$ sudo bash
```

Once you get the root shell, you can execute any root command without having to enter `sudo` in front of it every time.

6. Built in commands won't work with Sudo – Command not found

`sudo` invokes an executable as the another user, so `bash` built in commands won't work. It will give "sudo command not found" error as shown below.

For example, `umask` is a bash built-in command, which will not work when used along with `sudo` as shown below.

```
$ sudo umask
sudo: umask: command not found
```

Work-around: To use bash shell built-in command in `sudo`, first get the root shell, by doing `'sudo bash'` and then execute the shell built in command.

7. View Unauthorized Sudo command executions from `auth.log`

When an user who doesn't have `sudo` permission, tries to execute `sudo` command, they'll get following error message.

```
$ sudo ls /
[sudo] password for test:
tux is not in the sudoers file. This incident will be reported.
```

Anytime this happens, it will be logged in the `/var/log/auth.log` file for `sysadmins` to view any unauthorized `sudo` access.

```
Sep 25 18:41:35 sathiya sudo: tux : user NOT in sudoers ; TTY=pts/4 ;
PWD=/home/tux ; USER=root ; COMMAND=/bin/ls /
```

"ls" commands

1. List Files using `ls` with no option

ls with no option list files and directories in bare format where we won't be able to view details like file types, size, modified date and time, permission and links etc.

```
# ls
```

2. List Files With option `-l`

Here, **ls -l** (**-l** is character not one) shows file or directory, size, modified date and time, file or folder name and owner of file and it's permission.

```
# ls -l
```

3. View Hidden Files

List all files including hidden file starting with `.'`.

```
# ls -a
```

4. List Files with Human Readable Format with option `-lh`

With combination of **-lh** option, shows sizes in human readable format.

ls -lh

5. List Files and Directories with '/' Character at the end

Using **-F** option with **ls** command, will add the '/' Character at the end each directory.

ls -F

6. List Files in Reverse Order

The following command with **ls -r** option display files and directories in reverse order.

ls -r

7. Recursively list Sub-Directories

ls -R option will list very long listing directory trees. See an example of output of the command.

ls -R

8. Reverse Output Order

With combination of **-ltr** will shows latest modification file or directory date as last.

ls -ltr

9. Sort Files by File Size

With combination of **-ls** displays file size in order, will display big in size first.

ls -ls

10. Display Inode number of File or Directory

We can see some number printed before file / directory name. With **-i** options list file / directory with inode number.

ls -i

11. Shows version of ls command

Check version of ls command.

ls --version

12. Show Help Page

List help page of ls command with their option.

ls --help

Usage: ls [OPTION]... [FILE]..

13. List Directory Information

With **ls -l** command list files under directory **/tmp**. Wherein with **-ld** parameters displays information of **/tmp** directory.

ls -l /tmp

ls -ld /tmp/

14. Display UID and GID of Files

To display **UID** and **GID** of files and directories. use option **-n** with ls command.

ls -n

"cd" command

1. Change from current directory to /usr/local.

cd /usr/local

2. Change from current directory to /usr/local/lib using absolute path.

cd /usr/local/lib

3. Change from current working directory to /usr/local/lib using relative path.

cd lib

4. (a) Move one directory back from where you are now.

cd -

4. (b) Change Current directory to parent directory.

cd ..

5. Show last working directory from where we moved (use **'-'** switch) as shown.

cd --

6. Move two directory up from where you are now.

cd ../ ../

7. Move to users home directory from anywhere.

```
cd ~
```

8. Change working directory to current working directory (seems no use of in General).

```
cd .  
or  
cd ./
```

9. Your present working Directory is `"/usr/local/lib/python3.4/dist-packages/ "`, change it to `"/home/avi/Desktop/ "`, in one line command, by moving up in the directory till `'/'` then using absolute path.

```
cd ../../../../home/avi/Desktop/
```

10. Change from current working directory to `/var/www/html` without typing in full using TAB.

```
cd /v<TAB>/w<TAB>/h<TAB>
```

11. Navigate from your current working directory to `/etc/v__ _`, Oops! You forgot the name of directory and not supposed to use TAB.

```
cd /etc/v*
```

12. You need to navigate to user `'av'` (not sure if it is avi or avt) home directory, without using TAB.

```
cd /home/av?
```

14. Change to a directory containing white spaces.

```
cd test\ tecmint/  
or  
cd 'test tecmint'  
or  
cd "test tecmint"/
```

15. Change from current working directory to Downloads and list all its settings in one go.

```
cd ~/Downloads && ls
```

"chmod" Command

1. Add single permission to a file/directory

Changing permission to a single set. + symbol means adding permission. For example, do the following to give execute permission for the user irrespective of anything else:

```
$ chmod u+x filename
```

2. Add multiple permission to a file/directory

Use comma to separate the multiple permission sets as shown below.

```
$ chmod u+r,g+x filename
```

3. Remove permission from a file/directory

Following example removes read and write permission for the user.

```
$ chmod u-rx filename
```

4. Change permission for all roles on a file/directory

Following example assigns execute privilege to user, group and others (basically anybody can execute this file).

```
$ chmod a+x filename
```

5. Make permission for a file same as another file (using reference)

If you want to change a file permission same as another file, use the reference option as shown below. In this example, file2's permission will be set exactly same as file1's permission.

```
$ chmod --reference=file1 file2
```

6. Apply the permission to all the files under a directory recursively

Use option -R to change the permission recursively as shown below.

```
$ chmod -R 755 directory-name
```

7. Change execute permission only on the directories (files are not affected)

On a particular directory if you have multiple sub-directories and files, the following command will assign execute permission only to all the sub-directories in the current directory (not the files in the current directory).

```
$ chmod u+X *
```

"chown" command

1. Change the owner of a file

```
chown OWNER FILE
```

2. Change the group of a file

Through the chown command, the group (that a file belongs to) can also be changed.

```
chown :GROUP FILE
```

3. Change both owner and the group

```
chown OWNER:GROUP FILE
```

4. Using chown command on symbolic link file

```
chown OWNER:GROUP SYMLINK_PATH
```

5. Using chown command to forcefully change the owner/group of symbolic file. Using flag '-h', you can forcefully change the owner or group of a symbolic link as shown below.

```
chown -h OWNER:GROUP SYMLINK_PATH
```

6. Change owner only if a file is owned by a particular user

Using chown "--from" flag, you can change the owner of a file, only if that file is already owned by a particular owner.

```
chown --from=OWNER OWNER FILE
```

7. Change group only if a file already belongs to a certain group

Here also the flag '-from' is used but in the following way:

```
# ls -l tmpfile
```

```
-rw-r--r-- 1 himanshu friends 0 2012-05-22 20:03 tmpfile
```

```
# chown --from=:friends :family tmpfile
```

```
# ls -l tmpfile
```

```
-rw-r--r-- 1 himanshu family 0 2012-05-22 20:03 tmpfile
```


Since the file 'tmpfile' actually belonged to group 'friends' so the condition was correct and the command was successful.

So we see that by using the flag '-from=:<conditional-group-name>' we can change the group under a particular condition.

NOTE: By following the template '-from=<conditional-owner-name>:<conditional-group-name>', condition on both the owner and group can be applied.

8. Copy the owner/group settings from one file to another

This is possible by using the '-reference' flag.

```
chown --reference=REF_FILE FILE
```

9. Change the owner/group of the files by traveling the directories recursively
This is made possible by the '-R' option.

```
chown -R OWNER:GROUP FILE
```

10. Using chown to forcefully change the owner/group of a symbolic link directory recursively

This can be achieved by using the flag -H

```
# chown -R -H OWNER:GROUP SYMLINK_FILE
```

11. List all the changes made by the chown command

Use the verbose option -v, which will display whether the ownership of the file was changed or retained as shown below.

```
# chown -v -R OWNER:GROUP FILE
```

"mkdir command"

mkdir is a command for creating directories in Linux system. This command is a built-in command.

1. You can type **mkdir** directly from your console to use it.

```
$ mkdir
```

By default, running mkdir without any parameter will create directory under the current directory.

When Linux found that the directory which suppose to be created is already exist, then Linux will telling us that Linux can't create it.

2. Create multiple directories

We can also create multiple directories at the same time. Let say we want to create directories named **ubuntu, redhat and slackware**. Then the syntax will be like this :

```
$ mkdir ubuntu redhat slackware
```

3. Add directory include its sub-directory

When you want to created a include its sub-directory, you will need to use **-p** parameter. This parameter will create parent directory first, if mkdir cannot find it. Let say we want to create directory named **letter** and directory named **important** under directory letter. Then the syntax will be like this :

```
$ mkdir -p letter/important
```

4. Set access privilege

Using **-m** parameter, we can also set the access privilege for the new directory on-the-fly. Here's an example.

```
$ mkdir -m=r-- letter
```

The above command will create a directory named letter and give access privilege **read-only** for the **directory owner, directory group owner** and **anybody**.

5. Print message a message for each created directory

If we want, we can use **-v** parameter to do this. Here's an example.

```
$ mkdir -v ubuntu redhat slackware
```

"pwd" command

1. Print your current working directory.

```
Pwd
```

2. Print working directory from environment even if it contains symlinks.

```
Pwd -L
```

3. Print actual physical current working directory by resolving all symbolic links.
Pwd -P

"rm" command

1. How to Delete Empty Directories in Unix?

rmdir command will delete the empty directories. i.e directory without any sub-directories or files.

```
rmdir DIRNAME
```

To ensure that you are deleting an empty directory you should use rmdir command. If there is any files / directories in that directory it will display the following error.

```
$ rmdir test
rmdir: failed to remove `test': Directory not empty
```

2. How to Delete Nested Empty Directories in Linux?

Use option -p, to delete nested directories as shown below.

```
$ rmdir -p dir1/dir2/dir3
```

Note: Don't get panic that how a directory can be nested and also empty. It is nested when you are invoking the command, but it deletes the inner most directory first, and makes the next level directory empty then it deletes that directory. And it continue doing so.

The rmdir -p dir1/dir2/dir3 is equivalent to

```
$ rmdir dir1/dir2/dir3 dir1/dir2 dir1
```

3. Delete Directory Which has Content (i.e Directory with Files and Sub-directories)

Some times you may want to delete directory which has contents in it. You can do it with rm command as shown below.

```
$ rm -rf DIRNAME
```

This will delete the directory including all the files and sub-directories. This is very

dangerous when you use it accidentally as you cannot recover those files easily. So it is strongly recommended that you pay attention and think twice before executing the `rm -rf` command.

4. Delete Interactively: Avoid using `-f` in `rm` at the early stages.

If you've ever accidentally executed `rm -rf` by mistake, you may want to seriously consider using `-i` option to delete the files and folders interactively as shown below (especially under root).

Deleting a directory recursively & interactively.

```
# rm -ir DIRNAME
```

Deleting a file interactively.

```
# rm -i FILENAME
```

If you are a Linux newbie, don't use `-f` option in root until get super comfortable with the command line. Instead, try to use `-i` option as shown above.

"echo" command

1. Input a line of text and display on standard output

```
echo "Hello"
```

2. Declare a variable and echo its value. For example, Declare a variable of `x` and assign its value=10.

```
$ x=256
```

```
echo its value:
```

```
echo "The value of variable x = $x "
```

Note: The ``-e`` option in Linux acts as interpretation of escaped characters that are backslashed.

3. Using option `\b` – backspace with backslash interpreter `-e` which removes all the spaces in between.

```
echo -e "Hello \bis \bWorld \b!"
```

4. Using option `\n` – New line with backspace interpreter `-e` treats new line from where it is used.

```
echo -e "Hello\nWorld \n!"
```

Hello

World

!

5. Using option `\t` – horizontal tab with backspace interpreter `-e` to have horizontal tab spaces.

```
echo -e "Hello \tWorld \t!"
```

Hello World !

6. How about using option new Line `\n` and horizontal tab `\t` simultaneously.

```
echo -e "\n\tHello \n\tWorld \n\t!"
```

Hello
World
!

7. Using option `\v` – vertical tab with backspace interpreter `-e` to have vertical tab spaces.

```
echo -e "\vHello \vWorld \v! "
```

Hello
World

!

8. Using option `\r` – carriage return with backspace interpretor `-e` to have specified carriage return in output.

```
echo -e "Hello \rWorld !"
```

World !

9. Omit echoing trailing new line using option `-n`.

```
echo -n "Hello World !"
```

Hello World !:~/Documents\$

10. Using option `\a` – alert return with backspace interpretor `-e` to have sound alert.

```
echo -e "Tecmint is a community of \aLinux Nerds"
```

Tecmint is a community of Linux Nerds

Note: Make sure to check Volume key, before firing.

11. Print all the files/folder using echo command (ls command alternative).

```
echo *
```

12. Print files of a specific kind. For example, let's assume you want to print all `.mp3` files, use the following command.

```
echo *.jpeg
```

13. The echo can be used with redirect operator to output to a file and not standard output.

```
echo "How are you?" > test
```

`id` command

1. Display your own UID and GID

```
id
```

2. How do I find a specific user's UID?

```
id -u <username>
```

3. How do I find a specific user's GID?

```
id -g <username>
```

4. How do I see the UID and all groups associated with a user name?

```
id <username>
```

5. Find out all the groups a user belongs to...

```
id -G <username>
```

6. Display a name instead of a UID/GID

By default, id command displays number for the -G, -g and -u options. You can force id command to display the name of the UID or GID instead of the number for the -G, -g and -u options by passing the -n option as follows:

```
id -ng <username>
```

```
id -nu <username>
```

```
id -nG <username>
```

7. How do I display real ID instead of the effective ID for specified user?

You can show the real ID for the -g, -G and -u options instead of the effective ID by passing the -r option:

```
id -r -g <username>
```

```
id -r -u <username>
```

```
id -r -G <username>
```

8. How do I display SELinux show security context on Linux?

To display only the security context of the current user, type:

```
id -Z
```

9. Print user name, UID and all the groups to which the user belongs

```
id -a
```

“cp” command

1. Run cp without any options

This is a very basic cp usage. To copy a file named myfile.txt from one location to another location, we can type like this :

```
$ cp myfile.txt /home/pungki/office
```

2. Copy multiple files at the same time

To copy multiple files at the same time, we can just put the files behind the copy command which are separated by space. Here's an example :

```
$ cp file_1.txt file_2.txt file_3.txt /home/pungki/office
```

3. Copy a directory

Copying a directory is a little bit tricky. You need to add **-r** or **-R** option to do it. **-r** or **-R** option means recursive. **This option is a must** whether the directory is empty or not. Here's an example :

```
$ cp -r directory_1 /home/pungki/office
```

4. Create hard links to files instead of copying them

Copying files means you must have some space on the storage to store the copied files. Sometimes for any reasons, you may want to create “shortcut” or links to the files instead of copying them. To do this, we can use **-l** option.

```
$ cp -l file_4.txt /home/pungki/office
```

5. Create symbolic links to files

There is another type of links called **softlinks** or **symbolic links**. We use **-s** option to do this. Here's a sample command.

```
$ cp -s /home/pungki/Documents/file_6.txt file_6.txt
```


6. Copy without following symbolic links in Source

To do this, we can use **-P** option. When cp command found a file with symbolic links, it will copy the as is. Take a look at the sample below.

```
$ cp -P file_6.txt ./movie
```

7. Preserve the file permission and other attributes

You need to pass the -p option to save the following file attributes of each source file as allowed by permissions:

1. File modification time
2. File access time
3. File flags
4. File mode
5. File user ID and group ID

```
cp -p file1 file2
```

8. Copy with following symbolic links in Source

Now we can do this with **-L** option. Basically, **this is an opposite** of -P option above. Here's the sample.

```
$ cp -L file_6.txt ./movie
```

9. Archive the files

When we are going to copy a directory, we will use **-r or -R** option. But we can also use **-a** option to archive file. This will create an **exact copy** of files and directories including symbolic links if any. Here's a sample :

```
$ cp -a directory_1/ /home/pungki/office
```

10. Explain what is being done

By default, when copying activity is success, we will see a command prompt again. If you want to know what happen during the copying file, we can use **-v** option.

```
$ cp -v *.txt /home/pungki/office
```

11. Copy only when the source file is newer

To do this, we can use **-u** option. Take a look this example below.

```
$ cp -vu *.txt /home/pungki/office
```

12. Use interactive mode

Interactive mode will ask if the destination folder have already the file. To activate interactive mode, use **-i** option.

```
$ cp -ir directory_1/ /home/pungki/office/
```

13. Create backup date of each copied file

When the destination folder already have the file, by default cp command will overwrite the same file in the destination directory. Using **--backup** option, cp command will make a backup of each existing destination file. *../office will refer to /home/pungki/office*. Here's a sample :

```
$ cp --backup=simple -v *.txt ../office
```

14. Copy only file attributes

Cp command also provide us with **--attributes-only** option. As we can guess from its name, this option will only copy a file name and its attributes without copying any data. Here's a sample.

```
$ cp --attributes-only file_6.txt -v ../office
```

15. Force copying

Using **-f** option will force the copying activity. If the destination files cannot be opened, then **-f** will try again.

```
$ cp -f *.txt -v ../office
```

16. Remove destination before copy

To do this, we can use **--remove-destination** option. This option is **contrast with -f option** above. If the cp command find the same file name on the destination folder, cp command will remove destination file first, the copy the new one. Here's an example.

```
$ cp --remove-destination *.txt -v ../office
```

"touch" command

1. How to Create an Empty File

The following touch command creates an empty (zero byte) new file called sheena.

```
# touch test
```

2. How to Create Multiple Files

By using touch command, you can also create more than one single file. For example the following command will create 3 files named, test test1 & test2

```
# touch test test1 test2
```

3. How to Change File Access and Modification Time

To change or update the last access and modification times of a file called test, use the -a option as follows. The following command sets the current time and date on a file. If the test file does not exist, it will create the new empty file with the name.

```
# touch -a test
```

4. How to Avoid Creating New File

Using -c option with touch command avoids creating new files. For example the following command will not create a file called test if it does not exists.

```
# touch -c test
```

5. How to Change File Modification Time

If you like to change the only modification time of a file called test, then use the -m option with touch command. Please note it will only updates the last modification times (not the access times) of the file.

```
# touch -m test
```

6. Explicitly Set the Access and Modification times

You can explicitly set the time using -c and -t option with touch command. The format would be as follows.

```
# touch -c -t YYDDHHMM test
```

For example the following command sets the access and modification date and time to a file test as 17:30 (17:30 p.m.) December 10 of the current year (2012).

```
# touch -c -t 12101730 test
```

7. How to Use the time stamp of another File

The following touch command with -r option, will update the time-stamp of file test1 with the time-stamp of test file. So, both the file holds the same time stamp.

```
# touch -r test test1
```

8. Create a File using a specified time

If you would like to create a file with specified time other than the current time, then the format should be.

```
# touch -t YYMMDDHHMM.SS test
```

"cat" commands

1. Display the contents of a file

When you pass the filename as an argument to cat, it displays the contents of the file as shown below.

```
cat FILENAME
```

You can also display contents of more than one file as shown below.

```
cat FILE1 FILE2
```

2. Create a New File

Using cat command, the lines received from stdin can be redirected to a new file using redirection symbols.

When you type simply cat command without any arguments, it just receives the stdin content and displays it in the stdout. So after typed the line once, when you press enter, the same line gets printed in the subsequent line as seen below.

```
cat
```

You can also redirect the stdout to a new file as shown below.

```
cat >cmd_usage.txt
cat cmd_usage.txt
```

Sometimes you may have to append the content to a file, use >> redirection symbol as shown below.

```
cat >>cmd_usage.txt
```

3. Copy File Content

Redirection symbols in unix plays an important role in processing the standard file descriptor contents. Using it, you can copy the contents of one file into another as shown below.

```
cat FILE1 > FILE2
```

4. Concatenate Contents of Multiple Files

Through cat command, you will be able to concatenate contents of more than one file into a new file.

For example, the codes from program.pl and program2.pl gets combined into a new file all_pgrm.pl.

```
cat program.pl program2.pl >all_pgrm.pl
```

5. Display Line numbers

To display the contents of a file with the line number in front of each line, use option -n.

```
cat -n FILE
```

6. Concatenate File Contents along with Input from Stdin

There is a possibility to read lines from stdin along with concatenation of other files. Hence the user can type his own content whenever its required.

In the following example, you can insert a few lines (from stdin) in the beginning while combining files together.

```
cat - FILE1 FILE2 >FILE3
Contents from file : FILE1, FILE2
```

As seen above, - is the place where you can read from stdin, accordingly 1 line from stdin has been inserted into the beginning of a new file called FILE3 with the latter contents from FILE1 & FILE2:

7. Don't Display Repeated Empty Output Lines

Sometimes the file would contain repeated empty lines which you don't want to display in the stdout while listing it out. cat command provides an option called -s which will suppress consecutive empty output lines into one and displays.

```
cat -sn FILENAME
```

8. Display End of Line and TAB characters

You can make the cat to display the \$ character at end of every line. Normally by listing file contents, users cant identify whitespaces at the end of each lines, by using the cat -e option.

For instance, use -e option on the file program.pl. respectively.

```
$ cat -ne FILENAME
```

Use option -T to display the tab characters. It displays ^I for TAB character.

```
$ cat -neT FILENAME
```

"tail" command

The **linux tail command** does something like its name. It displays the last few lines of a file. It is mostly used for viewing log file updates as these updates are appended to the log files. This article explains tail command with most common options.

1. With a **file name as argument**, it displays last 10 lines for that file.

```
Tail <filename>
```

2. If **multiple files are given as argument**, it precedes each with a header displaying the name of the file.

```
Tail file1 file2
```

3. If you **do not want to print the headers**, you can use quiet mode with -q or --quiet or --silent option.

Tail -q file1 file2

4. if you **always want to print headers**, even when only one file name is provided as argument, you can use -v or --verbose option

tail -v file

5. When you want to change the default behavior and change the last number of lines from 10 to something else, -n option is used

tail -n3 file1

6. For **printing the last K bytes of the file**, use -c or --bytes=K option

tail -c 30 file

7. Alternatively, you can use -c +K to output bytes starting with the Kth of each file.

Tail -c +30 file

8. When viewing some log file, if you want to check some changes being made at the end of the file, i.e. if you want to output appended data as the file grows, you will use -f or --follow option. For the illustration of this option, let us display the last 3 lines with follow option

tail -n3 -f file

9. You can **adjust the sleep interval** between the iterations, i.e. it will check after the provided number of seconds for the append updates with -s or --sleep-interval=N (default is 1) seconds.

tail -n 3 -f -s 4 file

"less" command

1. Less Command – Search Navigation

Once you've opened a log file (or any file) using less file-name, use the following keys to search. Please note that the match will be highlighted automatically by default.

Forward Search

- / – search for a pattern which will take you to the next occurrence.
- n – for next match in forward
- N – for previous match in backward

Backward Search

- ? – search for a pattern which will take you to the previous occurrence.
- n – for next match in backward direction
- N – for previous match in forward direction

Tip: If you don't bother about which direction the search is happening, and you want to search file path, or URL, such as `"/home/tux/"`, you can use backward search (`?pattern`) which will be handy as you don't want to escape slashes each time.

Search Path

In forward: `/\home\tux\`

In backward: `?/home/tux/`

2. Less Command – Screen Navigation

Use the following screen navigation commands while viewing large log files.

- CTRL+F – forward one window
- CTRL+B – backward one window
- CTRL+D – forward half window
- CTRL+U – backward half window

3. Less Command – Line navigation

In a smaller chunk of data, where you want to locate particular error, you may want to navigate line by line using these keys:

- j – navigate forward by one line
- k – navigate backward by one line

4. Less Command – Other Navigations

The following are other navigation operations that you can use inside the less pager.

- G – go to the end of file
- g – go to the start of file
- q or ZZ – exit the less pager

5. Simulate tail -f inside less pager – Press F

Once you've opened a file using less command, any content that is appended to the file after that will not be displayed automatically. However, you can press F less command will show the status 'waiting for data'. This is as similar to 'tail -f'.

6. Less Command – Count magic

Similar to Vim editor navigation command, you can give 10j to scroll 10 lines down, or 10k to go up by 10 lines.

- 10j – 10 lines forward.
- 10k – 10 lines backward.
- CTRL+G – show the current file name along with line, byte and percentage statistics.

7. Other useful Less Command Operations

- v – using the configured editor edit the current file.
- h – summary of less commands
- &pattern – display only the matching lines, not all.

8. Less Command – Marked navigation

When you are viewing a large log file using less command, you can mark a particular position and return back to that place again by using that mark.

- ma – mark the current position with the letter 'a',
- 'a – go to the marked position 'a'.

10. Less Command – Multiple file paging

Method 1: You can open multiple files by passing the file names as arguments.

```
$ less file1 file2
```

Method 2: While you are viewing file1, use :e to open the file2 as shown below.

```
$ less file1  
:e file2
```

Navigation across files: When you opened more than two files (for e.g – less *), use the following keys to navigate between files.

- :n – go to the next file.

- :p – go to the previous file.

“zip” command

1. Creating a zip file

The zip command in unix or linux system creates an archive with the specified files. This is shown below:

```
> zip archive linux-virtual-server.bat unix-server.dat
adding: linux-virtual-server.bat (deflated 80%)
adding: unix-server.dat (deflated 80%)
> ls
archive.zip docs linux-virtual-server.bat unix-server.dat
```

The above command creates the zip file with name archive.zip

2. Extracting files from zip

To extract files from the zip, use the unzip command in unix system. This is shown below:

```
> unzip archive.zip
Archive: archive.zip
  inflating: linux-virtual-server.bat
  inflating: unix-server.dat
> ls
archive.zip linux-virtual-server.bat unix-server.dat
```

3. Removing file from a zip file

After creating a zip file, you can remove a file from the archive using the -d option. To remove the file unix-server.dat from the archive, run the below zip command:

```
> zip -d archive.zip unix-server.dat
deleting: unix-server.dat

> unzip archive.zip
Archive: archive.zip
  inflating: linux-virtual-server.bat
```

4. Update existing zip file

You can update the files in already created zip file. If any of the files are modified after zipping, you can fresh the zip file with only those modified files using the -f option.

```
> zip -f archive.zip  
freshening: linux-virtual-server.bat (stored 0%)
```

Another way is using the -u option. This option can be used to update the specified list of files or add new files to the existing zip file.

```
> zip -u archive.zip linux-virtual-server.bat temp  
updating: linux-virtual-server.bat (deflated 79%)  
adding: temp (stored 0%)
```

5. Recursively zip files in directory.

To zip a directory recursively, use the -r option with the zip command. This example is shown below:

```
> zip -r dir_archive docs  
adding: docs/ (stored 0%)  
adding: docs/unix.pdf (stored 0%)  
adding: docs/oracle.pdf (stored 0%)  
adding: docs/linux.pdf (stored 0%)
```

6. Excluding files in zipping

Let say you are zipping all the files in the current directory and want to exclude some unwanted files. You can exclude these unwanted files using the -x option.

```
zip exclude_archive * -x linux-virtual-server.bat
```

The above command zips all the files in the current directory except the file linux-virtual-server.bat

7. Faster compressing

You can compress the files very fast using the -1 option with zip command. An example is shown below with and without using fast compression.

```
> zip -1 fast_archive linux-virtual-server.bat
adding: linux-virtual-server.bat (deflated 79%)
```

```
> zip normal_archive linux-virtual-server.bat
adding: linux-virtual-server.bat (deflated 80%)
```

If you use fast compression, the archive file created will occupy more space (size) when compared to normal compression.

8. Better compression.

To reduce more amount of size the files occupied, you can use the -9 option with the zip command. This gives a better compression.

```
> zip -9 better_archive linux-virtual-server.bat
adding: linux-virtual-server.bat (deflated 81%)
```

“unzip” command

1. To use *unzip* to extract all members of the archive *letters.zip* into the current directory and subdirectories below it, creating any subdirectories as necessary:

```
unzip letters
```

2. To extract all members of *letters.zip* into the current directory only:

```
unzip -j letters
```

3. To test *letters.zip*, printing only a summary message indicating whether the archive is OK or not:

```
unzip -tq letters
```

4. To test *all* zipfiles in the current directory, printing only the summaries:

```
unzip -tq \*.zip
```

5. To extract to standard output all members of *letters.zip* whose names end in *.tex*, auto-converting to the local end-of-line convention and piping the output into *more*(1):

```
unzip -ca letters \*.tex | more
```

6. To extract the binary file *paper1.dvi* to standard output and pipe it to a printing program:

```
unzip -p articles paper1.dvi | dvips
```

7. To extract all FORTRAN and C source files--*.f, *.c, *.h, and Makefile--into the /tmp directory:

```
unzip source.zip "[fch]" Makefile -d /tmp
```

(the double quotes are necessary only in Unix and only if globbing is turned on).

8. To extract all FORTRAN and C source files, regardless of case (e.g., both *.c and *.C, and any makefile, Makefile, MAKEFILE or similar):

```
unzip -C source.zip "[fch]" makefile -d /tmp
```

9. To extract any such files but convert any uppercase MS-DOS or VMS names to lowercase and convert the line-endings of all of the files to the local standard (without respect to any files that might be marked ``binary"):

```
unzip -aaCL source.zip "[fch]" makefile -d /tmp
```

10. To extract only newer versions of the files already in the current directory, without querying (NOTE: be careful of unzipping in one timezone a zipfile created in another--ZIP archives other than those created by Zip 2.1 or later contain no timezone information, and a ``newer" file from an eastern timezone may, in fact, be older):

```
unzip -fo sources
```

11. To extract newer versions of the files already in the current directory and to create any files not already there (same caveat as previous example):

```
unzip -uo sources
```

12. To display a diagnostic screen showing which *unzip* and *zipinfo* options are stored in environment variables, whether decryption support was compiled in, the compiler with which *unzip* was compiled, etc.:

`unzip -v`

13. In the last five examples, assume that UNZIP or UNZIP_OPTS is set to -q. To do a singly quiet listing:

`unzip -l file.zip`

To do a doubly quiet listing:

`unzip -ql file.zip`

(Note that the ``.zip" is generally not necessary.) To do a standard listing:

`unzip --ql file.zip`

or

`unzip -l-q file.zip`

or

`unzip -l--q file.zip`

“gzip” command

1. Compression the file with deleting original file.

`Gzip linuxFileName`

This will replace a linuxFileName.gz which has size of 80% of linuxFileName in the current directory. The filename size is reduced by this command. Once gz file is created, linuxFileName should be deleted

2. Compression the file with keeping original file.

`Gzip -c linuxFileName`

This command will behave same expect deleting the original file. So original file should be kept as it is.

2. Uncompress/decompress the gz file

`Gunzip fileName.gz`

This will unzip the filename.gz and get the original file before using gzip command

3. Compression multiple files in a directory

`Gzip -r directoryname`

using -r option, recursively traverse all the files, meaning all the files in current directory including all the files subdirectory and create a directoryname.gz which contains all the files in the current directory and subdirectory

After compression, total size of the files is approximately 20% less gz file.

4. Uncompress/decompress the gz file into multiple files

`Gunzip -r fileName.gz`

This will unzip the filename.gz into the the multiple original files before using gzip -r command

5. Compression files fastly:-

`Gzip -1 filename.txt`

`Gzip -fast filename.txt`

The both above options compress filename.txt very fast and create filename.txt.gz folder

6. Compression files fastly:-

`Gzip -9 filename.txt`

`Gzip -best filename.txt`

The both above options compress filename.txt files slowly and create filename.txt.gz folder

`"tar" command`

1. Creating an archive using tar command

Creating an uncompressed tar archive using option cvf

This is the basic command to create a tar archive.

```
$ tar cvf archive_name.tar dirname/
```

In the above command:

- c – create a new archive
- v – verbosely list files which are processed.
- f – following is the archive file name

Creating a tar gzipped archive using option cvzf

The above tar cvf option, does not provide any compression. To use a gzip compression on the tar archive, use the z option as shown below.

```
$ tar cvzf archive_name.tar.gz dirname/
```

- z – filter the archive through gzip

Note: .tgz is same as .tar.gz

Note: I like to keep the 'cvf' (or tvf, or xvf) option unchanged for all archive creation (or view, or extract) and add additional option at the end, which is easier to remember. i.e cvf for archive creation, cvfz for compressed gzip archive creation, cvfj for compressed bzip2 archive creation etc., For this method to work properly, don't give - in front of the options.

Creating a bzipipped tar archive using option cvjf

```
$ tar cvjf archive_name.tar.bz2 dirname/
```

- j – filter the archive through bzip2

gzip vs bzip2: bzip2 takes more time to compress and decompress than gzip. bzip2 archival size is less than gzip.

Note: .tbz and .tb2 is same as .tar.bz2

2. Extracting (untar) an archive using tar command

Extract a tar file using option x as shown below:

```
$ tar xvf archive_name.tar
```

- x – extract files from archive

Extract a gzipped tar archive (*.tar.gz) using option xvzf

Use the option z for uncompressing a gzip tar archive.

```
$ tar xvfz archive_name.tar.gz
```


Extracting a bziped tar archive (*.tar.bz2) using option xvjf

Use the option j for uncompressing a bzip2 tar archive.

```
$ tar xvfj archive_name.tar.bz2
```

Note: In all the above commands v is optional, which lists the file being processed.

3. Listing an archive using tar command

You can view the *.tar file content before extracting as shown below.

```
$ tar tvf archive_name.tar
```

You can view the *.tar.gz file content before extracting as shown below.

```
$ tar tvfz archive_name.tar.gz
```

You can view the *.tar.bz2 file content before extracting as shown below.

```
$ tar tvfj archive_name.tar.bz2
```

4. Extract a single file from tar, tar.gz, tar.bz2 file

To extract a specific file from a tar archive, specify the file name at the end of the tar xvf command as shown below. The following command extracts only a specific file from a large tar file.

```
$ tar xvf archive_file.tar /path/to/file
```

Use the relevant option z or j according to the compression method gzip or bzip2 respectively as shown below.

```
$ tar xvfz archive_file.tar.gz /path/to/file
```

```
$ tar xvfj archive_file.tar.bz2 /path/to/file
```

5. Extract a single directory from tar, tar.gz, tar.bz2 file

To extract a single directory (along with its subdirectory and files) from a tar archive, specify the directory name at the end of the tar xvf command as shown below. The following extracts only a specific directory from a large tar file.

```
$ tar xvf archive_file.tar /path/to/dir/
```

To extract multiple directories from a tar archive, specify those individual directory names at the end of the tar xvf command as shown below.

```
$ tar xvf archive_file.tar /path/to/dir1/ /path/to/dir2/
```

Use the relevant option z or j according to the compression method gzip or bzip2 respectively as shown below.

```
$ tar xvfz archive_file.tar.gz /path/to/dir/
```

```
$ tar xvfj archive_file.tar.bz2 /path/to/dir/
```

6. Extract group of files from tar, tar.gz, tar.bz2 archives using regular expression

You can specify a regex, to extract files matching a specified pattern. For example, following tar command extracts all the files with pl extension.

```
$ tar xvf archive_file.tar --wildcards '*.pl'
```

Options explanation:

- `--wildcards *.pl` – files with pl extension

7. Adding a file or directory to an existing archive using option -r

You can add additional files to an existing tar archive as shown below. For example, to append a file to *.tar file do the following:

```
$ tar rvf archive_name.tar newfile
```

This newfile will be added to the existing archive_name.tar. Adding a directory to the tar is also similar,

```
$ tar rvf archive_name.tar newdir/
```

Note: You cannot add file or directory to a compressed archive. If you try to do so, you will get “tar: Cannot update compressed archives” error as shown below.

```
$ tar rvfz archive_name.tgz newfile
tar: Cannot update compressed archives
Try `tar --help' or `tar --usage' for more information.
```

8. Verify files available in tar using option -W

As part of creating a tar file, you can verify the archive file that got created using the option W as shown below.

```
$ tar cvfW file_name.tar dir/
```

If you are planning to remove a directory/file from an archive file or from the file system, you might want to verify the archive file before doing it as shown below.

```
$ tar tvfW file_name.tar
Verify 1/file1
1/file1: Mod time differs
1/file1: Size differs
Verify 1/file2
Verify 1/file3
```

If an output line starts with Verify, and there is no differs line then the file/directory is Ok. If not, you should investigate the issue.

Note: for a compressed archive file (*.tar.gz, *.tar.bz2) you cannot do the verification.

Finding the difference between an archive and file system can be done even for a compressed archive. It also shows the same output as above excluding the lines with Verify.

Finding the difference between gzip archive file and file system

```
$ tar dfz file_name.tgz
```

Finding the difference between bzip2 archive file and file system

```
$ tar dfj file_name.tar.bz2
```

9. Estimate the tar archive size

The following command, estimates the tar file size (in KB) before you create the tar file.

```
$ tar -cf - /directory/to/archive/ | wc -c
20480
```

The following command, estimates the compressed tar file size (in KB) before you create the tar.gz, tar.bz2 files.

```
$ tar -czf - /directory/to/archive/ | wc -c
508
```

```
$ tar -cjf - /directory/to/archive/ | wc -c
428
```

"ln" command

1. What is Soft Link and Hard Link?

Soft Link

Linux OS recognizes the data part of this special file as a reference to another file path. The data in the original file can be accessed through the special file, which is called as Soft Link.

To create a soft link, do the following (ln command with -s option):

```
$ ln -s /full/path/of/original/file /full/path/of/soft/link/file
```

Hard Link

With Hard Link, more than one file name reference the same inode number. Once you create a directory, you would see the hidden directories "." and "..". In this, "." directory is hard linked to the current directory and the ".." is hard linked to the parent directory.

When you use link files, it helps us to reduce the disk space by having single copy of the original file and ease the administration tasks as the modification in original file reflects in other places.

To create a hard link, do the following (ln command with no option):

```
$ ln /full/path/of/original/file /full/path/of/hard/link/fil
```

2. Create Symbolic Link for File or Directory

Create a symbolic link for a File

The following examples creates a symbolic link library.so under /home/chris/lib, based on the library.so located under /home/chris/src/ directory.

```
$ cd /home/chris/lib
```

```
$ ln -s /home/chris/src/library.so library.so
```

```
$ ls -l library.so
```

```
lrwxrwxrwx 1 chris chris    21 2010-09-18 07:23 library.so ->
/home/chris/src/library.so
```

Create a symbolic link for a Directory

Just like file, you can create symbolic link for directories as shown below.

```
$ mkdir /home/chris/obj
```

```
$ cd tmp
```

```
$ ln -s /home/chris/obj objects
```

```
$ ls -l objects
```

```
lrwxrwxrwx 1 chris chris    6 2010-09-19 16:48 objects -> /home/chris/obj
```

Note: The inode of the original file/directory and the soft link should not be identical.

3. Create Hard Link for Files

The inode number for the hard linked files would be same. The hard link for files can be created as follows,

```
$ ln src_original.txt dst_link.txt
```

```
$ ls -i dst_link.txt
```

```
253564 dst_link.txt
```

```
$ ls -i src_original.txt
```

```
253564 src_original.txt
```

Note: Unix / Linux will not allow any user (even root) to create hard link for a directory.

4. Create Links Across Different Partitions

When you want to create the link across partitions, you are allowed to create only the symbolic links. Creating hard link across partitions is not allowed, as Unix can't create/maintain same inode numbers across partitions.

You would see the "Invalid cross-device link" error when you are trying to create a hard link file across partitions.

```
# mount /dev/sda5 /mnt
```

```
# cd /mnt
```

```
# ls
```

main.c Makefile

```
# In Makefile /tmp/Makefile
```

```
ln: creating hard link `/tmp/Makefile' to `Makefile': Invalid cross-device link
```

And the symbolic link can be created in the same way as we did in the above.

5. Backup the Target Files If it Already Exists

When you create a new link (if another file exist already with the same name as the new link name), you can instruct ln command to take a backup of the original file before creating the new link using the `--backup` option as shown below.

```
$ ls
```

```
ex1.c ex2.c
```

```
$ ln --backup -s ex1.c ex2.c
```

```
$ ls -lrt
```

```
total 8
```

```
-rw-r--r-- 1 chris chris 20 2010-09-19 16:57 ex1.c
```

```
-rw-r--r-- 1 chris chris 20 2010-09-19 16:57 ex2.c~
```

```
lrwxrwxrwx 1 chris chris  5 2010-09-19 17:02 ex2.c -> ex1.c
```

Note: If you don't want the backup and overwrite the existing file then use `-f` option.

6. Create Link Using "No-Deference" ln Command Option

While creating a new soft link, normally OS would de-reference the destination path before it creates the new soft link.

Sometimes you might not want ln command to create the new link, if the destination path is already a symbolic link that is pointing to a directory.

Following examples shows a normal way of creating soft link inside a directory.

```
$ cd ~
```

```
$ mkdir example
```

```
$ ln -s /etc/passwd example
```

```
$ cd example/
```

```
$ ls -l
```

```
total 0
lrwxrwxrwx 1 root root 16 2010-09-19 17:24 passwd -> /etc/passwd
```

In case the "example" directory in the above code-snippet is a symbolic link pointing to some other directory (for example second-dir), the ln command shown will still create the link under second-dir. If you don't want that to happen, use ln -n option as shown below.

```
$ cd ~
```

```
$ rm -rf example
```

```
$ mkdir second-dir
```

```
$ ln -s second-dir example
```

```
$ ln -n -s /etc/passwd example
ln: creating symbolic link `example': File exists
```

Note: In the above example, if you don't use the -n option, the link will be created under ~/second-dir directory.

7. Create Link for Multiple Files at the Same Time

In the following example, there are two directories — first-dir and second-dir. The directory first-dir contains couple of C program files. If you want to create soft links for these files in second-dir, you'll typically do it one by one. Instead, you can create soft list for multiple files together using -t option as shown below.

```
$ ls
first-dir second-dir
```

```
$ ls first-dir
ex1.c ex2.c
```

```
$ cd second-dir
```

```
$ ln -s ../first-dir/*.c -t .
```

```
$ ls -l
total 0
lrwxrwxrwx 1 chris chris 14 2010-09-19 15:20 ex1.c -> ../first-dir/ex1.c
lrwxrwxrwx 1 chris chris 14 2010-09-19 15:20 ex2.c -> ../first-dir/ex2.c
```

Keep in mind that whenever you are creating link files with -t option, it is better to go into target directory and perform the link creation process. Otherwise, you would face the broken link files as shown below.

```
$ cd first-dir
```

```
$ ln -s *.c /home/chris/second-dir
```

```
$ cd /home/chris/second-dir
```

```
$ ls -l
```

```
total 0
```

```
lrwxrwxrwx 1 chris chris 5 2010-09-19 15:26 ex1.c -> ex1.c
```

```
lrwxrwxrwx 1 chris chris 5 2010-09-19 15:26 ex2.c -> ex2.c
```

Instead, you might also use actual path for source files to create the link properly.

8. Removing the Original File When a Soft Link is pointing to it

When the original file referred by a soft-link is deleted, the soft link will be broken as shown below.

```
$ ln -s file.txt /tmp/link
```

```
$ ls -l /tmp/link
```

```
lrwxrwxrwx 1 chris chris 9 2010-09-19 15:38 /tmp/link -> file1.txt
```

```
$ rm file.txt
```

```
$ ls -l /tmp/link
```

```
lrwxrwxrwx 1 chris chris 9 2010-09-19 15:38 /tmp/link -> file1.txt
```

9. Links Help You to Increase the Partition Size Virtually

Let us assume that you have two partitions – 5GB and 20GB. The first partition does not have too much free space available in it. If a program located on the first partition needs more space (For example, for its log file), you can use some of the space from the second partition by creating a link for the log files as shown below.

Consider that partition1 is mounted on /, and partition2 is mounted to /mnt/. Let us assume that the logs that are located on partition1 is running out of space, and you've decided to move them to partition2. You can achieve this as shown below.


```
$ mkdir /mnt/logs  
$ cd /logs  
$ mv * /mnt/logs  
$ cd /; rmdir logs  
$ ln -s /mnt/logs logs
```

10. Removing the Hard Linked Files

When you delete a file that is hard linked, you would be still able to access the content of the file until you have the last file which is hard linked to it, as shown in the example below.

Create a sample file.

```
$ vim src_original.txt  
Created this file to test the hard link.
```

Create a hard link to the sample file.

```
$ ln src_original.txt dst_link.txt
```

Delete the original file.

```
$ rm src_original.txt
```

You can still access the original file content by using the hard link you created.

```
$ cat dst_link.txt  
Created this file to test the hard link.
```

“top” command

1. Show Processes Sorted by any Top Output Column – Press O

By default top command displays the processes in the order of CPU usage. When the top command is running, press M (upper-case) to display processes sorted by memory usage.

To sort top output by any column, Press O (upper-case O) , which will display all the possible columns that you can sort by as shown below.

Current Sort Field: P for window 1:Def

Select sort field via field letter, type any other key to return

a: PID	= Process Id	v: nDRT	= Dirty Pages count
d: UID	= User Id	y: WCHAN	= Sleeping in Function
e: USER	= User Name	z: Flags	= Task Flags
.....			

When the linux top command is running, Press R, which does the sort in reverse order.

2. Kill a Task Without Exiting From Top – Press k

Once you've located a process that needs to be killed, press 'k' which will ask for the process id, and signal to send.If you have the privilege to kill that particular PID, it will get killed successfully.

3. Renice a Unix Process Without Exiting From Top – Press r

Press r, if you want to just change the priority of the process (and not kill the process). This will ask PID for renice, enter the PID and priority.

4. Display Selected User in Top Output Using top -u

Use top -u to display a specific user processes only in the top command output.

```
$ top -u tux
```

While unix top command is running, press u which will ask for username.

Use top -p to display specific PIDs.

5. Display All CPUs / Cores in the Top Output – Press 1 (one)

Top output by default shows CPU line for all the CPUs combined together as shown below.

Press 1 (one), when the top command is running, which will break the CPU down and show details for all the individual CPUs running on the system.

6. Refresh Unix Top Command Output On demand (or) Change Refresh Interval

By default, linux top command updates the output every 3.0 seconds. When you want to update the output on-demand, press space bar.

To change the output update frequency, press d in interactive mode, and enter the time in seconds as shown below.

7. Highlight Running Processes in the Linux Top Command Output – Press z or b

Press z or b, which will highlight all running process as shown below.

8. Display Absolute Path of the Command and its Arguments – Press c

Press c which will show / hide command absolute path, and arguments.

9. Quit Top Command After a Specified Number of Iterations Using top -n

Until you press q, top continuously displays the output. If you would like to view only a certain iteration and want the top to exit automatically use -n option as shown below.

The following example will show 2 iterations of unix top command output and exit automatically

```
$ top -n 2
```

10. Executing Unix Top Command in Batch Mode

If you want to execute top command in the batch mode use option -b as shown below.

```
$ top -b -n 1
```

11. Split Top Output into Multiple Panels – Press A

To display multiple views of top command output on the terminal, press A. You can cycle through these windows using 'a'. This is very helpful, when you can sort the output on multiple windows using different top output columns.

12. Get Top Command Help from Command Line and Interactively

Get a quick command line option help using `top -h` as shown below.

```
$ top -h
```

Press `h` while `top` command is running, which will display help for interactive `top` commands.

13. Decrease Number of Processes Displayed in Top Output – Press `n`

Press `n` in the Interactive mode, which prompts for a number and shows only that.

14. Toggle Top Header to Increase Number of Processes Displayed

By default `top` displays total number process based on the window height. If you like to see additional process you might want to eliminate some of the top header information.

- Press `l` – to hide / show the load average. 1st header line.
- Press `t` – to hide / show the CPU states. 2nd and 3rd header line.
- Press `m` – to hide / show the memory information. 4th and 5th line.

15. Save Top Configuration Settings – Press `W`

If you've made any interactive `top` command configurations suggested in the above examples, you might want to save those for all future `top` command output. Once you've saved the top configuration, next time when you invoke the `top` command all your saved top configuration options will be used automatically.

To save the top configuration, press `W`, which will write the configuration files to `~/toprc`. This will display the write confirmation message.

“ssh” command

1. Login to remote host:

- The First time when you login to the remotehost from a localhost, it will display the host key not found message and you can give “yes” to continue.

The host key of the remote host will be added under .ssh2/hostkeys directory of your home directory.

- ssh -l user hostname
- The Second time when you login to the remote host from the localhost, it will prompt only for the password as the remote host key is already added to the known hosts list of the ssh client.

2. Debug SSH Client:

Sometimes it is necessary to view debug messages to troubleshoot any SSH connection issues. For this purpose, pass -v (lowercase v) option to the ssh as shown below.

- Example with debug message:
ssh -v -l USERNAME REMOTEHOST

5. Escape Character: (Toggle SSH session, SSH session statistics etc.)

Escape character ~ get's SSH clients attention and the character following the ~ determines the escape command.

Toggle SSH Session: When you've logged on to the remotehost using ssh from the localhost, you may want to come back to the localhost to perform some activity and go back to remote host again. In this case, you don't need to disconnect the ssh session to the remote host. Instead follow the steps below.

- Login to remotehost from localhost
 - Now you are connected to the remotehost
 - To come back to the localhost temporarily, type the escape character ~ and Control-Z. When you type ~ you will not see that immediately on the screen until you press <Control-Z> and press enter. So, on the remotehost in a new line enter the following key strokes for the below to work: ~<Control-Z>
 - Now you are back to the localhost and the ssh remotehost client session runs as a typical unix background job.
 - You can go back to the remote host ssh without entering the password again by bringing the background ssh remotehost session job to foreground on the localhost.

"scp" command

The basic syntax of scp is very simple to memorize. It looks like this

```
$ scp source_file_path destination_file_path
```

if you are "sending" file from your local machine to a remote machine (uploading) the syntax would look like this

```
$ scp ~/my_local_file.txt user@remote_host.com:/some/remote/directory
```

1. Verbose output

With verbose output, the scp program would output lots of information about what it does in the background. This is often useful when the program fails or is unable to complete the request. The verbose output would then indicate the exact point where the program ran into issues.

```
$ scp -v ~/test.txt root@192.168.1.3:/root/help2356.txt
```

2. Transfer multiple files

Multiple files can be specified separated by a space like this

```
$ scp foo.txt bar.txt username@remotehost:/path/directory/
```

To copy multiple files from remote host to current local directory

```
$ scp username@remotehost:/path/directory/{foo.txt,bar.txt} .
```

3. Copy entire directory (recursively)

To copy an entire directory from one host to another use the r switch and specify the directory

```
$ scp -v -r ~/Downloads root@192.168.1.3:/root/Downloads
```

4. Copy files across 2 remote hosts

Scp can copy files from 1 remote host to another remote host as well.

```
$ scp user1@remotehost1:/some/remote/dir/foobar.txt  
user2@remotehost2:/some/remote/dir/
```

5. Speed up the transfer with compression

A super cool option to speed up the transfer to save time and bandwidth. All you need to do is use the C option to enable compression. The files are compressed on the fly and decompressed on the destination.

```
$ scp -vrC ~/Downloads root@192.168.1.3:/root/Downloads
```

In the above example we moved the entire directory with compression enabled. The speed gain would depend on how much the files could be compressed.

6. Limit the bandwidth usage

If you do not want scp to take up the entire available bandwidth, then use the `-l` option to limit the maximum speed in Kbit/s.

```
$ scp -vrC -l 400 ~/Downloads root@192.168.1.3:/root/Downloads
```

7. Connect to a different port number on remote host

If the remote server has ssh daemon running on a different port (default is 22), then you need to tell scp to use that particular port number using the `-P` option.

```
$ scp -vC -P 2200 ~/test.txt root@192.168.1.3:/some/path/test.txt
```

8. Preserve file attributes

The `-p` option (smallcase), would preserve modification times, access times, and modes from the original file.

```
$ scp -C -p ~/test.txt root@192.168.1.3:/some/path/test.txt
```

9. Quiet mode

In quiet mode (`-q` option), the scp output would get suppressed, and would disable the progress meter as well as warning and diagnostic messages.

```
$ scp -vCq ~/test.txt root@192.168.1.3:/some/path/test.txt
```

10. Specify identity file

When using key based (passwordless) authentication, you would need to specify the identity file which contains the private key. This option is directly passed to the ssh command and works the same way.

```
$ scp -vCq -i private_key.pem ~/test.txt  
root@192.168.1.3:/some/path/test.txt
```

11. Use a different ssh_config file

Use the `-F` option to specify a different ssh_config file.

```
$ scp -vC -F /home/user/my_ssh_config ~/test.txt  
root@192.168.1.3:/some/path/test.txt
```

12. Use different cipher

Scp by default uses the AES cipher/encryption. Sometimes you might want to use a different cipher. Using a different cipher can speed up the transfer process. For example blowfish and arcfour are known to be faster than AES (but less secure).

```
$ scp -c blowfish -C ~/local_file.txt username@remotehost:/remote/path/file.txt
```

In the above example we use the blowfish cipher along with compression. This can give significant speed boost depending on available bandwidth.

Day 2

“wget” command

1. Download Single File

```
# wget http://mirror.nbrc.ac.in/centos/7.0.1406/isos/x86_64/CentOS-7.0-1406-x86_64-DVD.iso
```

This command will download the CentOS 7 ISO file in the user's current working directory.

2. Resume Partial Downloaded File

There are some scenarios where we start downloading a large file but in the middle Internet got disconnected , so using the option '-c' in wget command we can resume our download from where it got disconnected.

```
# wget -c http://mirror.nbrc.ac.in/centos/7.0.1406/isos/x86_64/CentOS-7.0-1406-x86_64-DVD.iso
```

3. Download Files in the background

We can download the file in the background using the option '-b' in wget command.

```
$ wget -b http://mirror.nbrc.ac.in/centos/7.0.1406/isos/x86_64/CentOS-7.0-1406-x86_64-DVD.iso
Continuing in background, pid 4505.
Output will be written to 'wget-log'.
```

As we can see above that downloading progress is capture in 'wget-log' file in user's current directory.


```
linuxtechi@localhost:~$ tail -f wget-log
```

```
2300K ..... 0% 48.1K 18h5m
2350K ..... 0% 53.7K 18h9m
2400K ..... 0% 52.1K 18h13m
2450K ..... 0% 58.3K 18h14m
2500K ..... 0% 63.6K 18h14m
2550K ..... 0% 63.4K 18h13m
2600K ..... 0% 72.8K 18h10m
2650K ..... 0% 59.8K 18h11m
2700K ..... 0% 52.8K 18h14m
2750K ..... 0% 58.4K 18h15m
2800K ..... 0% 58.2K 18h16m
2850K ..... 0% 52.2K 18h20m
```

4. Limiting Download Speed .

By default wget command try to use full bandwidth, but there may be case that you are using shared internet , so if you try to download huge file using wget , this may slow down Internet of other users. This situation can be avoided if you limit the download speed using `'-limit-rate'` option.

```
#wget --limit-rate=100k
http://mirror.nbrc.ac.in/centos/7.0.1406/isos/x86_64/CentOS-7.0-1406-x86_64-
DVD.iso
```

In the above example,the download speed is limited to 100k.

5. Download Multiple Files using `'-i'` option

If you want to download multiple files using wget command , then first create a text file and add all URLs in the text file.

```
# cat download-list.txt
url1
url2
url3
url4
```

Now issue issue below Command :

```
# wget -i download-list.txt
```

6. Increase Retry Attempts.

We can increase the retry attempts using `'-tries'` option in wget. By default wget command retries 20 times to make the download successful.

This option becomes very useful when you have internet connection problem and you are downloading a large file , then there is a chance of failures in the download.

```
# wget --tries=75 http://mirror.nbrc.ac.in/centos/7.0.1406/isos/x86_64/CentOS-7.0-1406-x86_64-DVD.iso
```

7. Redirect wget Logs to a log File using -o

We can redirect the wget command logs to a log file using '-o' option.

```
# wget -o download.log  
http://mirror.nbrc.ac.in/centos/7.0.1406/isos/x86_64/CentOS-7.0-1406-x86_64-DVD.iso
```

Download.log file will be created in the user's current directory.

8. Download Full website for local viewing.

```
# wget --mirror -p --convert-links -P ./<Local-Folder> website-url
```

Whereas

- -mirror : turn on options suitable for mirroring.
- -p : download all files that are necessary to properly display a given HTML page.
- -convert-links : after the download, convert the links in document for local viewing.
- -P ./Local-Folder : save all the files and directories to the specified directory.

9. Reject file types while downloading.

When you are planning to download full website , then we can force wget command not to download images using '-reject' option .

```
# wget --reject=png Website-To-Be-Downloaded
```

10. Setting Download Quota using wget -Q

We can force wget command to quit downloading when download size exceeds certain size using '-Q' option

```
# wget -Q10m -i download-list.txt
```

Note that quota will never affect downloading a single file. So if you specify wget -Q10m ftp://wuarchive.wustl.edu/ls-IR.gz, all of the ls-IR.gz will be downloaded. The same goes even when several URLs are specified on the command-line.

However, quota is respected when retrieving either recursively, or from an input file. Thus you may safely type 'wget -Q10m -i download-list.txt' download will be aborted when the quota is exceeded.

11. Downloading file from password protected site.

```
# wget --ftp-user=<user-name> --ftp-password=<password> Download-URL
```

"curl" command

1. Download a Single File

The following command will get the content of the URL and display it in the STDOUT (i.e on your terminal).

```
$ curl http://www.centos.org
```

To store the output in a file, you can redirect it as shown below. This will also display some additional download statistics.

```
$ curl http://www.centos.org > centos-org.html
```

2. Save the cURL Output to a file

We can save the result of the curl command to a file by using -o/-O options.

- -o (lowercase o) the result will be saved in the filename provided in the command line
- -O (uppercase O) the filename in the URL will be taken and it will be used as the filename to store the result

```
$ curl -o mygettext.html
```

```
http://www.gnu.org/software/gettext/manual/gettext.html
```

Now the page gettext.html will be saved in the file named 'mygettext.html'. You can also note that when running curl with -o option, it displays the progress meter for the download .

When you use curl -O (uppercase O), it will save the content in the file named 'gettext.html' itself in the local machine.

```
$ curl -O http://www.gnu.org/software/gettext/manual/gettext.html
```

Note: When curl has to write the data to the terminal, it disables the Progress Meter, to avoid confusion in printing. We can use `>`|`-o`|`-O` options to move the result to a file.

3. Fetch Multiple Files at a time

We can download multiple files in a single shot by specifying the URLs on the command line.

Syntax:

```
$ curl -O URL1 -O URL2
```

Please note that when we download multiple files from a same sever, curl will try to re-use the connection.

4. Follow HTTP Location Headers with -L option

By default CURL doesn't follow the HTTP Location headers. It is also termed as Redirects. When a requested web page is moved to another place, then an HTTP Location header will be sent as a Response and it will have where the actual web page is located.

For example, when someone types google.com in the browser from India, it will be automatically redirected to `google.co.in`. This is done based on the HTTP Location header as shown below.

```
$ curl http://www.google.com
```

```
<TITLE>302 Moved</TITLE>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.co.in/">here</A>
```

The above output says that the requested document is moved to `http://www.google.co.in/`.

We can insists curl to follow the redirection using -L option, as shown below. Now it will download the google.co.in's html source code.

```
$ curl -L http://www.google.com
```

5. Continue/Resume a Previous Download

Using curl -C option, you can continue a download which was stopped already for some reason. This will be helpful when you download large files, and the download got interrupted.

If we say '-C -', then curl will find from where to start resuming the download. We can also give an offset '-C <offset>'. The given offset bytes will be skipped from the beginning for the source file.

Start a big download using curl, and press Ctrl-C to stop it in between the download.

```
$ curl -O http://www.gnu.org/software/gettext/manual/gettext.html
##### 20.1%
```

Note: -# is used to display a progress bar instead of a progress meter.

Now the above download was stopped at 20.1%. Using "curl -C -", we can continue the download from where it left off earlier. Now the download continues from 20.1%.

```
curl -C - -O http://www.gnu.org/software/gettext/manual/gettext.html
##### 21.1%
```

6. Limit the Rate of Data Transfer

You can limit the amount at which the data gets transferred using -limit-rate option. You can specify the maximum transfer rate as argument.

```
$ curl --limit-rate 1000B -O
http://www.gnu.org/software/gettext/manual/gettext.html
```

The above command is limiting the data transfer to 1000 Bytes/second. curl may use higher transfer rate for short span of time. But on an average, it will come around to 1000B/second.

The following was the progress meter for the above command. You can see that the current speed is near to the 1000 Bytes.

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
		Dload	Upload	Total	Spent	Left	Speed
1 1215k	1 13601	0 0	957	0 0:21:40	0:00:14	0:21:26	999
1 1215k	1 14601	0 0	960	0 0:21:36	0:00:15	0:21:21	999
1 1215k	1 15601	0 0	962	0 0:21:34	0:00:16	0:21:18	999

7. Download a file only if it is modified before/after the given time

We can get the files that are modified after a particular time using -z option in curl. This will work for both FTP & HTTP.

```
$ curl -z 21-Dec-11 http://www.example.com/yy.html
```

The above command will download the yy.html only if it is modified later than the given date and time

```
$ curl -z -21-Dec-11 http://www.example.com/yy.html
```

The above command will download the yy.html, if it is modified before than the given date and time.

Please refer 'man curl_getdate' for the various syntax supported for the date expression

8. Pass HTTP Authentication in cURL

Sometime, websites will require a username and password to view the content (can be done with .htaccess file). With the help of -u option, we can pass those credentials from cURL to the web server as shown below.

```
$ curl -u username:password URL
```

Note: By default curl uses Basic HTTP Authentication. We can specify other authentication method using -ntlm | -digest.

9. Download Files from FTP server

cURL can also be used to download files from FTP servers. If the given FTP path is a directory, by default it will list the files under the specific directory.

```
$ curl -u ftpuser:ftppass -O ftp://ftp_server/public_html/xss.php
```

The above command will download the xss.php file from the ftp server and save it in the local directory.

```
$ curl -u ftpuser:ftppass -O ftp://ftp_server/public_html/
```

Here, the given URL refers to a directory. So cURL will list all the files and directories under the given URL

10. List/Download using Ranges

cURL supports ranges to be given in the URL. When a range is given, files matching within the range will be downloaded. It will be helpful to download packages from the FTP mirror sites.

```
$ curl ftp://ftp.uk.debian.org/debian/pool/main/[a-z]/
```

The above command will list out all the packages from a-z ranges in the terminal.

11. Upload Files to FTP Server

Curl can also be used to upload files to the FTP server with -T option.

```
$ curl -u ftpuser:ftppass -T myfile.txt ftp://ftp.testserver.com
```

The above command will upload the file named myfile.txt to the FTP server. You can also upload multiple files at a same time using the range operations.

```
$ curl -u ftpuser:ftppass -T "{file1,file2}" ftp://ftp.testserver.com
```

Optionally we can use "." to get the input from STDIN and transfer to the remote.

```
$ curl -u ftpuser:ftppass -T - ftp://ftp.testserver.com/myfile_1.txt
```

The above command will get the input from the user from Standard Input and save the contents in the ftp server under the name 'myfile_1.txt'.

You can provide one '-T' for each URL and the pair specifies what to upload where.

12. More Information using Verbose and Trace Option

You can get to know what is happening using the -v option. -v option enable the verbose mode and it will print the details

```
curl -v http://google.co.in
```

The about command will output the following

```
* About to connect() to www.google.co.in port 80 (#0)
* Trying 74.125.236.56... connected
* Connected to www.google.co.in (74.125.236.56) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.21.0 (i486-pc-linux-gnu) libcurl/7.21.0 OpenSSL/0.9.8o
zlib/1.2.3.4 libidn/1.15 libssh2/1.2.6
> Host: www.google.co.in
```

```
> Accept: */*
>
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Date: Tue, 10 Apr 2012 11:18:39 GMT
< Expires: -1
< Cache-Control: private, max-age=0
< Content-Type: text/html; charset=ISO-8859-1
< Set-Cookie:
  PREF=ID=7c497a6b15cc092d:FF=0:TM=1334056719:LM=1334056719:S=UORp
  BwxFmTRkbXLj; expires=Thu, 10-Apr-2014 11:18:39 GMT; path=/;
  domain=.google.co.in
.
.
```

If you need more detailed information then you can use the `-trace` option. The trace option will enable a full trace dump of all incoming/outgoing data to the given file

```
=> Send header, 169 bytes (0xa9)
0000: 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 0d 0a GET / HTTP/1.1..
0010: 55 73 65 72 2d 41 67 65 6e 74 3a 20 63 75 72 6c User-Agent: curl
..
0060: 2e 32 2e 33 2e 34 20 6c 69 62 69 64 6e 2f 31 2e .2.3.4 libidn/1.

0070: 31 35 20 6c 69 62 73 73 68 32 2f 31 2e 32 2e 36 15 libssh2/1.2.6
0080: 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 67 6f 6f 67 ..Host: www.goog
0090: 6c 65 2e 63 6f 2e 69 6e 0d 0a 41 63 63 65 70 74 le.co.in..Accept
00a0: 3a 20 2a 2f 2a 0d 0a 0d 0a : */*....
== Info: HTTP 1.0, assume close after body
<= Recv header, 17 bytes (0x11)
0000: 48 54 54 50 2f 31 2e 30 20 32 30 30 20 4f 4b 0d HTTP/1.0 200 OK.
0010: 0a
```

This verbose and trace option will come in handy when curl fails due to some reason and we don't know why.

13. Get Definition of a Word using DICT Protocol

You can use cURL to get the definition for a word with the help of DICT protocol. We need to pass a Dictionary Server URL to it.

```
$ curl dict://dict.org/d:bash
```

The above command will list the meaning for bash as follows

```
151 "Bash" gcide "The Collaborative International Dictionary of English v.0.48"
```


Bash \Bash\, v. t. [imp. & p. p. {Bashed}; p. pr. & vb. n. {Bashing}.] [Perh. of imitative origin; or cf. Dan. baske to strike, bask a blow, Sw. basa to beat, bas a beating.] To strike heavily; to beat; to crush. [Prov. Eng. & Scot.] --Hall Caine.
[1913 Webster]

Bash her open with a rock. --Kipling.
[Webster 1913 Suppl.]

151 "Bash" gcide "The Collaborative International Dictionary of English v.0.48"

Bash \Bash\, n.

1. a forceful blow, especially one that does damage to its target.
[PJC]

Now you can see that it uses "The Collaborative International Dictionary of English". There are many dictionaries are available. We can list all the dictionaries using

```
$ curl dict://dict.org/show:db
```

```
jargon "The Jargon File (version 4.4.7, 29 Dec 2003)"
foldoc "The Free On-line Dictionary of Computing (26 July 2010)"
easton "Easton's 1897 Bible Dictionary"
hitchcock "Hitchcock's Bible Names Dictionary (late 1800's)"
bouvier "Bouvier's Law Dictionary, Revised 6th Ed (1856)"
```

Now in-order to find the actual meaning of Bash in computer we can search for bash in "foldoc" dictionary as follows

```
$ curl dict://dict.org/d:bash:foldoc
```

The result will be,

bash

Bourne Again SHell. {GNU}'s {command interpreter} for {Unix}. Bash is a {Posix}-compatible {shell} with full {Bourne shell} syntax, and some {C shell} commands built in. The Bourne Again Shell supports {Emacs}-style command-line editing, job control, functions, and on-line help. Written by Brian Fox of {UCSB}.

For more details with regard to DICT please read RFC2229

14. Use Proxy to Download a File

We can specify cURL to use proxy to do the specific operation using -x option. We need to specify the host and port of the proxy.

```
$ curl -x proxyserver.test.com:3128 http://google.co.in
```

15. Send Mail using SMTP Protocol

cURL can also be used to send mail using the SMTP protocol. You should specify the from-address, to-address, and the mailserver ip-address as shown below.

```
$ curl --mail-from blah@test.com --mail-rcpt foo@test.com  
smtp://mailserver.com
```

Once the above command is entered, it will wait for the user to provide the data to mail. Once you've composed your message, type . (period) as the last line, which will send the email immediately.

```
Subject: Testing  
This is a test mail
```

"du" command

1. A basic example

```
$ du -a
```

-a flag is used to show the disk usage of all the files and directories.

2. Display output in human readable form using -h

```
$ du -ah
```

3. Display grand total in the output using -c

```
$ du -ahc
```

4. Display only the total count using -s

If you wish to fetch only the total size in form of a summary, use the -s flag.

```
$ du -sh
```

5. Feed input files from stdin using --files0-from=-

If there arises a situation wherein you want to provide 'du' input on the go, then this is possible through the '--files0-from' flag.

```
$ du -ah --files0-from=-
```

Note that we passed value as '-' to the flag '--files0-from' in order to indicate 'du' to start accepting the file names from stdin. After entering each file name press ctrl+D twice in order to produce the output for that file.

6. End the output with null byte using -0

If you wish to produce each output ending without a newline but with a null byte instead then use the -0 flag.

```
$ du -am -0
```

7. Customize the block size in output through -block-size

Talking of block size, some times it is required to have output in a different way. For example :

```
$ du -ac
```

The above output is represented in terms of number of 1024 bytes blocks. Now suppose if we require the output to be in number of 2048 bytes block, then in this case the flag '--block-size' can be used.

```
$ du -ahc --block-size=2048
```

8. Display output in bytes using -b

To get the output in terms of bytes, the -b flag can be used.

```
$ du -achb
```

9. Exclude particular types of file(s) using --exclude

Suppose we do not want to have the statistics of .txt files. So for these types of requirements there is a '--exclude' flag.

```
$ du -cbha --exclude="*.txt"
```

10. Display the modification time and customize the display style

This can be achieved through --time and --time-style flags.

```
$ du -cbha --time
```

```
$ du -cbha --time --time-style=iso
```

"df" Command

1. Check File System Disk Space Usage

The “df” command displays the information of device name, total blocks, total disk space, used disk space, available disk space and mount points on a file system.

```
df
```

2. Display Information of all File System Disk Space Usage

The same as above, but it also displays information of dummy file systems along with all the file system disk usage and their memory utilization.

```
df -a
```

3. Show Disk Space Usage in Human Readable Format

The df command provides an option to display sizes in Human Readable formats by using ‘-h’ (prints the results in human readable format (e.g., 1K 2M 3G)).

```
df -h
```

4. Display Information of /home File System

To see the information of only device /home file system in human readable format use the following command.

```
df -hT /home
```

5. Display Information of File System in Bytes

To display all file system information and usage in 1024-byte blocks, use the option ‘-k’ (e.g. –block-size=1K) as follows.

```
df -k
```

6. Display Information of File System in MB

To display information of all file system usage in MB (Mega Byte) use the option as ‘-m’.

```
df -m
```

7. Display Information of File System in GB

To display information of all file system statistics in GB (Gigabyte) use the option as ‘df -h’.

```
df -h
```

8. Display File System Inodes

Using `-i` switch will display the information of number of used inodes and their percentage for the file system.

```
df -i
```

9. Display File System Type

To check the file system type of your system use the option `-T`. It will display file system type along with other information.

```
df -T
```

10. Include Certain File System Type

If you want to display certain file system type use the `-t` option. For example, the following command will only display ext3 file system.

```
df -t ext3
```

11. Exclude Certain File System Type

If you want to display file system type that doesn't belong to ext3 type use the option as `-x`. For example, the following command will only display other file systems types other than ext3.

```
df -x ext3
```

"dd" Command

1. Create an iso file from a cdrom

In this example the dd command allows you to create an iso file from a source file.

```
dd if=/dev/cdrom of=image.iso bs=2k
```

2. Create an Image of a Hard Drive

This is a very useful use of the dd command. Here you are going to take an image of an existing hard drive and save it to another storage location.

```
dd if=/dev/sda of=~/.sda_disk.img
```

3. Restore an Image file to a Hard Drive

To restore an image file that you have saved from a hard drive you can issue a command similar to:

```
dd if=sda_disk.img of=/dev/sdb
```

The above command restores the image file taken from /dev/sda and restores it to the location of /dev/sdb.

4. Make a Backup of a Hard Drive

The following dd command will make a backup of the specified drive to another drive attached to the same system.

```
dd if=/dev/sda of=/dev/sdb conv=noerror,sync
```

The above copies /dev/sda to /dev/sdb. The options "conv=noerror,sync" is used to specify that we do not stop processing when an error occurs. The sync parameter specifies that any missing input is replaced with null bytes and processed normally.

5. Make a Backup of a specified Partition

The following dd command will allow you to make a copy of a specified partition.

```
dd if=/dev/sda1 of=~ /partition_sda1.img
```

6.Backup your MBR - Master Boot Record with dd

The following dd command will make a backup of the master boot record of the specified disk. The MBR is a 512 byte boot sector that is the first sector of a partitioned disk.

```
dd if=/dev/sda of=~ /partition_sda1.mbr bs=512 count=1
```

7.Restore your MBR - Master Boot Record with dd

The following dd command will restore a previously saved copy of your MBR to the specified drive.

```
dd if=~ /partition_sda1.mbr of=/dev/sda count=1 bs=512
```

8. Create a dummy test file with dd

The following dd command will create a dummy test file with the size specified by the block size and count.

```
dd if=/dev/zero of=~/.test1.file bs=1024 count=1
```

The above will create a dummy test file of 1024 bytes in size.

```
dd if=/dev/zero of=~/.test2.file bs=1024 count=1024
```

The above will create a dummy test file with a size of 1MB

```
dd if=/dev/zero of=~/.test3.file bs=1M count=10
```

“free” command

1. Display System Memory

Free command used to check the used and available space of physical memory and swap memory in KB. See the command in action below.

```
free
```

2. Display Memory in Bytes

Free command with option -b, display the size of memory in Bytes.

```
free -b
```

3. Display Memory in Kilo Bytes

Free command with option -k, display the size of memory in (KB) Kilobytes.

```
free -k
```

4. Display Memory in Megabytes

To see the size of the memory in (MB) Megabytes use option as -m.

```
free -m
```

5. Display Memory in Gigabytes

Using -g option with free command, would display the size of the memory in GB(Gigabytes).

```
free -g
```

6. Display Total Line

Free command with -t option, will list the total line at the end.

```
free -t
```

7. Disable Display of Buffer Adjusted Line

By default the free command display "buffer adjusted" line, to disable this line use option as -o.

```
free -o
```

8. Display Memory Status for Regular Intervals

The -s option with number, used to update free command at regular intervals. For example, the below command will update free command every 5 seconds.

```
free -s 5
```

9. Show Low and High Memory Statistics

The -l switch displays detailed high and low memory size statistics.

```
# free -l
```

"mount" command

1. Mount a CD-ROM

The device file for CD would exist under /dev directory. For example, a CD-ROM device will be mounted as shown below.

```
# mount -t iso9660 -o ro /dev/cdrom /mnt
```

In the above example, the option "-o ro" indicates that the cdrom should be mounted with read-only access. Also, make sure that the destination directory (in the above example, /mnt) exist before you execute the mount command.

2. View All Mounts

After you execute mount a partition or filesystem, execute the mount command without any arguments to view all the mounts.

```
# mount
```

3. Mount all the filesystem mentioned in /etc/fstab

The filesystems listed in /etc/fstab gets mounted during booting process. After booting, system administrator may unmount some of the partitions for various reasons. If you want all the filesystems to be mounted as specified in /etc/fstab, use -a option with mount as shown below:

Example /etc/fstab file entries:

```
# cat /etc/fstab
#
proc          /proc          proc    nodev,noexec,nosuid 0      0
# / was on /dev/sda5 during installation
/dev/sda5 /          ext4    errors=remount-ro 0      1
# /mydata was on /dev/sda6 during installation
/dev/sda6 /mydata    ext2    defaults          0      2
# /backup was on /dev/sda7 during installation
/dev/sda7 /backup    vfat    defaults          0      3
```

Execute mount command with -a option to mount all the /etc/fstab entries.

```
# mount -a
```

```
# mount
/dev/sda5 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
udev on /dev type devtmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
/dev/sda6 on /mydata type ext2 (rw)
/dev/sda7 on /backup type vfat (rw)
gvfs-fuse-daemon on /home/bala/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev,user=bala)
```

The same -a option can be used with umount to unmount all the filesystems mentioned in /etc/mtab

```
# umount -a
umount: /run/shm: device is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
```

```
umount: /run: device is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
umount: /dev: device is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
umount: /: device is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
```

Some filesystem are not unmounted as its busy or currently in use. Note that the files /etc/mstab and /proc/mounts contents would be similar.

4. Mount only a specific filesystem from /etc/fstab

When you pass only the directory name to mount, it looks for mount point entries, if not found, then search continuous for a device in /etc/fstab and gets mounted.

```
# mount | grep /mydata

# cat /etc/fstab | grep mydata
#####/mydata was on /dev/sda6 during installation#####
/dev/sda6 /mydata      ext2    defaults    0        2
```

As seen above, /mydata directory is not a mountpoint, but it is present in /etc/fstab.

```
# mount /mydata

# mount | grep /mydata
/dev/sda6 on /mydata type ext2 (rw)
```

If you execute the same again, you would get the error message as follows:

```
# mount /mydata
mount: /dev/sda6 already mounted or /mydata busy
mount: according to mtab, /dev/sda6 is already mounted on /mydata
```

Here you may also pass the device name instead of directory name (to be picked up from /etc/fstab file).

```
# mount /dev/sda6
```

Note that the files /etc/mstab and /proc/mounts contents would be similar.

5. View all mounted partitions of specific type

It is possible to list only the specific type of filesystem mounted using the option -l with -t as shown below:

```
# mount -l -t ext2  
/dev/sda6 on /mydata type ext2 (rw)
```

```
# mount -l -t ext4  
/dev/sda5 on / type ext4 (rw,errors=remount-ro)
```

As seen above, /dev/sda6 is the only ext2 partition and /dev/sda5 is the only ext4 partition accordingly.

6. Mount a Floppy Disk

The device file for floppy disk would exist under /dev directory. For example, a floppy disk will be mounted as shown below.

```
# mount /dev/fd0 /mnt  
# cd /mnt
```

After the successful mount, you would be able to access the contents of the floppy disk. Once you are done with it, use umount before you physically remove the floppy disk from the system.

```
# umount /mnt
```

7. Bind mount points to a new directory

The mountpoint can be binded to a new directory. So that you would be able to access the contents of a filesystem via more than one mountpoints at the same time.

Use -B option with olddir and newdir to be binded as follows,

```
# mount -B /mydata /mnt
```

Now the bind is done and you might verify it as follows,

```
# mount | grep /mydata  
/dev/sda6 on /mydata type ext2 (rw)  
/mydata on /mnt type none (rw,bind)
```

As seen above the bind is done properly. So when you do modification in filesystem in one place, you can see those reflection of it in other mount point as shown below:

```
# cd /mydata
# ls
test
# mkdir dir1
# mkdir dir2
# ls
test  dir1  dir2
# cd /mnt
# ls
test  dir1  dir2
```

8. Access contents from new mount point

Mount allows you to access the contents of a mount point from a new mount point. Its nothing but move a mounted tree to another place.

In the example below, the mount point /mydata will be accessed from /mnt using the option -M as shown below:

```
# mount -M /mydata /mnt/
```

Once its done, you cant use the old mount point as its moved to a new mount point and this can be verified as shown below:

```
# mount | grep /mydata
# mount | grep /mnt
/dev/sda6 on /mnt type ext2 (rw)
```

9. Mount without writing entry into /etc/mtab

During read only mount of /etc/, the /etc/mtab file entries cannot be modified by mount command. However, mount can be done without writing into /etc/mtab by using the option -n as follows,

```
# mount -n /dev/sda6 /mydata
```

You cannot see any entry for this /mydata in mount command output and as well from /etc/mtab file as follows:

```
# mount | grep /mydata
# cat /etc/mtab | grep /mydata
```

Access the contents of a mounted directory /mydata:

```
# cd /mydata
# ls
dir1 dir2 test
```

10. Mount filesystem with read or read/write access

To mount partition as read only, use -r option which is synonym to -o ro.

```
# mount /dev/sda6 /mydata -r
# mount | grep /mydata
/dev/sda6 on /mydata type ext4 (ro)
```

ext3 and ext4 filesystem would still allow you to do write operation when the filesystem is dirty. So, you may have to use "ro,noload" to prevent these kind of write operation.

```
# mount /dev/sda6 /mydata -t ext4 -o ro -o noload
# mount | grep /mydata
/dev/sda6 on /mydata type ext4 (ro,noload)
```

To mount a partition with read/write access, use -w option which is same as "-o rw" (i.e : default).

11. Remount the mounted filesystem

In order to mount the already mounted filesystem, use remount option and its normally used to remount the filesystem with read/write access when its previously mounted with read access.

The /mydata mount point is going to be remounted with read/write access from read access as shown below:

```
# mount | grep /mydata
/dev/sda6 on /mydata type ext4 (ro,noload)
# mount -o remount,rw /mydata
# mount | grep /mydata
/dev/sda6 on /mydata type ext4 (rw)
```

12. Mount an iso image into a directory

The iso image can be mounted as shown below:

```
# mount -t iso9660 -o loop pdf_collections.iso /mnt
# cd /mnt
# ls
perl/  php/  mysql/
```

"vmstat" command

1. Basic example

vmstat by default will display the memory usage (including swap).

```
$ vmstat
```

vmstat output contains the following fields:

- Procs – r: Total number of processes waiting to run
- Procs – b: Total number of busy processes
- Memory – swpd: Used virtual memory
- Memory – free: Free virtual memory
- Memory – buff: Memory used as buffers
- Memory – cache: Memory used as cache.
- Swap – si: Memory swapped from disk (for every second)
- Swap – so: Memory swapped to disk (for every second)
- IO – bi: Blocks in. i.e blocks received from device (for every second)
- IO – bo: Blocks out. i.e blocks sent to the device (for every second)
- System – in: Interrupts per second
- System – cs: Context switches
- CPU – us, sy, id, wa, st: CPU user time, system time, idle time, wait time

2. Display active and inactive memory

By default vmstat doesn't display this information. Use option -a, to display active and inactive memory information as shown below.

```
$ vmstat -a
```

3. Display number of forks since last boot

This displays all the fork system calls made by the system since the last boot. This displays all fork, vfork, and clone system call counts.

```
$ vmstat -f
```

4. Execute Every x seconds (for y number of times)

To execute every 2 seconds, do the following. You have to press Ctrl-C to stop this.

```
$ vmstat 2
```

To execute every 2 seconds for 10 times, do the following. You don't need to press Ctrl-C in this case. After executing 10 times, it will stop automatically.

```
$ vmstat 2 10
```

5. Display timestamp

When you use vmstat to monitor the memory usage repeatedly, it would be nice to see the timestamp along with every line item. Use option -t to display the time stamp as shown below.

```
$ vmstat -t 1 100
```

6. Display slab info

Use option -m, to display the slab info as shown below.

```
$ vmstat -m
```

7. Display statistics in a table format

Instead of displays the values in the record format, you can display the output of vmstat in table format using option -s as shown below.

```
$ vmstat -s
```

8. Display disk statistics

Use option -d to display the disk statistics as shown below. This displays the reads, writes, and I/O statistics of the disk.

```
$ vmstat -d
```

9. Increase the width of the display

The default output without increasing the width is shown below.

```
$ vmstat 1 3
procs -----memory----- ---swap-- -----io---- --system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0 0 3608688 148368 3898204 0 0 0 0 0 1 1 0 0 100 0 0
0 0 0 3608804 148368 3898204 0 0 0 0 0 72 30 0 0 100 0 0
0 0 0 3608804 148368 3898204 0 0 0 0 0 60 27 0 0 100 0 0
```

Use option -w to increase the width of the output columns as shown below. This give better readability.

```
$ vmstat -w 1 3
procs -----memory----- ---swap-- -----io---- --system-- --
---cpu-----
```

r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa
0	0	0	3608712	148368	3898204	0	0	0	0	1	1	0	0		
100	0	0													
0	0	0	3608712	148368	3898204	0	0	0	0	93	23	0	0		
100	0	0													
0	0	0	3608696	148368	3898204	0	0	0	0	35	34	0	0		
100	0	0													

10. Display statistics for a partition

To display the disk I/O statistics of a specific disk partition use option -p as shown below.

```
$ vmstat -p sdb1
```

11. Display in MB

By default vmstat displays the memory information in kb. To display in MB, use the option "-S m" as shown below.

```
$ vmstat -S m
```

"ifconfig" command

1. View All Network Setting

The "ifconfig" command with no arguments will display all the active interfaces details. The ifconfig command also used to check the assigned IP address of an server.

```
ifconfig
```

2. Display Information of All Network Interfaces

The following ifconfig command with -a argument will display information of all active or inactive network interfaces on server.

```
ifconfig -a
```

3. View Network Settings of Specific Interface

Using interface name (eth0) as an argument with "ifconfig" command will display details of specific network interface.


```
ifconfig eth0
```

4. How to Enable an Network Interface

The “up” or “ifup” flag with interface name (eth0) activates an network interface, if it is not in active state and allowing to send and receive information. For example, “ifconfig eth0 up” or “ifup eth0” will activate the eth0 interface.

```
ifconfig eth0 up  
OR  
ifup eth0
```

5. How to Disable an Network Interface

The “down” or “ifdown” flag with interface name (eth0) deactivates the specified network interface. For example, “ifconfig eth0 down” or “ifdown eth0” command deactivates the eth0 interface, if it is in active state.

```
ifconfig eth0 down  
OR  
ifdown eth0
```

6. How to Assign a IP Address to Network Interface

To assign an IP address to an specific interface, use the following command with an interface name (eth0) and ip address that you want to set. For example, “ifconfig eth0 192.168.1.4” will set the IP address to interface eth0.

```
ifconfig eth0 192.168.1.4
```

7. How to Assign a Netmask to Network Interface

Using the “ifconfig” command with “netmask” argument and interface name as (eth0) allows you to define an netmask to an given interface. For example, “ifconfig eth0 netmask 255.255.255.0” will set the network mask to an given interface eth0.

```
ifconfig eth0 netmask 255.255.255.0
```

8. How to Assign a Broadcast to Network Interface

Using the “broadcast” argument with an interface name will set the broadcast address for the given interface. For example, “ifconfig eth0 broadcast 192.168.1.125” command sets the broadcast address to an interface eth0.

```
ifconfig eth0 broadcast 192.168.1.125
```

9. How to Assign a IP, Netmask and Broadcast to Network Interface

To assign an IP address, Netmask address and Broadcast address all at once using "ifconfig" command with all arguments as given below.

```
ifconfig eth0 192.168.1.4 netmask 255.255.255.0 broadcast 192.168.1.125
```

10. How to Change MTU for an Network Interface

The "mtu" argument set the maximum transmission unit to an interface. The MTU allows you to set the limit size of packets that are transmitted on an interface. The MTU able to handle maximum number of octets to an interface in one single transaction. For example, "ifconfig eth0 mtu 1000" will set the maximum transmission unit to given set (i.e. 1000). Not all network interfaces supports MTU settings.

```
ifconfig eth0 mtu 1000
```

11. How to Enable Promiscuous Mode

What happens in normal mode, when a packet received by a network card, it verifies that the packet belongs to itself. If not, it drops the packet normally, but in the promiscuous mode is used to accept all the packets that flows through the network card.

Most of the today's network tools uses the promiscuous mode to capture and analyze the packets that flows through the network interface. To set the promiscuous mode, use the following command.

```
ifconfig eth0 promisc
```

12. How to Disable Promiscuous Mode

To disable promiscuous mode, use the "-promisc" switch that drops back the network interface in normal mode.

```
ifconfig eth0 -promisc
```

13. How to Add New Alias to Network Interface

The ifconfig utility allows you to configure additional network interfaces using alias feature. To add alias network interface of eth0, use the following command. Please note that alias network address in same sub-net mask. For example, if your eth0 network ip address is 192.168.1.4, then alias ip address must be 192.168.1.4

```
ifconfig eth0:0 172.16.25.127
```

Next, verify the newly created alias network interface address, by using "ifconfig eth0:0" command.

```
ifconfig eth0:0
```

14. How to Remove Alias to Network Interface

If you no longer required an alias network interface or you incorrectly configured it, you can remove it by using the following command.

```
ifconfig eth0:0 down
```

15. How to Change the MAC address of Network Interface

To change the MAC (Media Access Control) address of an eth0 network interface, use the following command with argument "hw ether". For example, see below.

```
ifconfig eth0 hw ether AA:BB:CC:DD:EE:FF
```

"netstat" commands

1. List all ports using netstat -a

```
# netstat -a | more
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	localhost:30037	*.*	LISTEN
udp	0	0	*:bootpc	*.*	

Active UNIX domain sockets (servers and established)

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ACC]	STREAM	LISTENING	6135	/tmp/.X11-unix/X0
unix	2	[ACC]	STREAM	LISTENING	5140	/var/run/acpid.socket

2. List all tcp ports using netstat -at

```
# netstat -at
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	localhost:30037	*.*	LISTEN
tcp	0	0	localhost:ipp	*.*	LISTEN
tcp	0	0	*:smtp	*.*	LISTEN
tcp6	0	0	localhost:ipp	:::*	LISTEN

3. List all udp ports using netstat -au

```
# netstat -au
```

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
-------	--------	--------	---------------	-----------------	-------

```

udp      0      0 *:bootpc      *.*
udp      0      0 *:49119        *.*
udp      0      0 *:mdns         *.*

```

4. List only listening ports using netstat -l

```

# netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 localhost:ipp        *.*                     LISTEN
tcp6     0      0 localhost:ipp        [::]:*                 LISTEN
udp      0      0 *:49119              *.*

```

5. List only listening TCP Ports using netstat -lt

```

# netstat -lt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 localhost:30037       *.*                     LISTEN
tcp      0      0 *:smtp               *.*                     LISTEN
tcp6     0      0 localhost:ipp        [::]:*                 LISTEN

```

6. List only listening UDP Ports using netstat -lu

```

# netstat -lu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp      0      0 *:49119              *.*
udp      0      0 *:mdns               *.*

```

7. List only the listening UNIX Ports using netstat -lx

```

# netstat -lx
Active UNIX domain sockets (only servers)
Proto RefCnt Flags   Type       State       I-Node  Path
unix  2      [ ACC ] STREAM    LISTENING   6294     private/maildrop
unix  2      [ ACC ] STREAM    LISTENING   6203     public/cleanup
unix  2      [ ACC ] STREAM    LISTENING   6302     private/ifmail
unix  2      [ ACC ] STREAM    LISTENING   6306     private/bsmtp

```

8. Show statistics for all ports using netstat -s

```

# netstat -s
Ip:
  11150 total packets received
    1 with invalid addresses
    0 forwarded
    0 incoming packets discarded
  11149 incoming packets delivered
  11635 requests sent out
Icmp:
  0 ICMP messages received
  0 input ICMP message failed.
Tcp:

```

582 active connections openings
2 failed connection attempts
25 connection resets received
Udp:
1183 packets received
4 packets to unknown port received.
.....

9. Show statistics for TCP (or) UDP ports using netstat -st (or) -su

```
# netstat -st
```

```
# netstat -su
```

10. Display PID and program names in netstat output using netstat -p

netstat -p option can be combined with any other netstat option. This will add the "PID/Program Name" to the netstat output. This is very useful while debugging to identify which program is running on a particular port.

```
# netstat -pt
```

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	1	0	ramesh-laptop.loc:47212	192.168.185.75:www	CLOSE_WAIT	2109/firefox
tcp	0	0	ramesh-laptop.loc:52750	lax:www	ESTABLISHED	2109/firefox

11. Don't resolve host, port and user name in netstat output

When you don't want the name of the host, port or user to be displayed, use netstat -n option. This will display in numbers, instead of resolving the host name, port name, user name.

This also speeds up the output, as netstat is not performing any look-up.

```
# netstat -an
```

If you don't want only any one of those three items (ports, or hosts, or users) to be resolved, use following commands.

```
# netsat -a --numeric-ports
```

```
# netsat -a --numeric-hosts
```

```
# netsat -a --numeric-users
```

12. Print netstat information continuously

netstat will print information continuously every few seconds.

```
# netstat -c
```

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	ramesh-laptop.loc:36130	101-101-181-225.ama:www	ESTABLISHED
tcp	1	1	ramesh-laptop.loc:52564	101.11.169.230:www	CLOSING
tcp	0	0	ramesh-laptop.loc:43758	server-101-101-43-2:www	ESTABLISHED
tcp	1	1	ramesh-laptop.loc:42367	101.101.34.101:www	CLOSING

^C

13. Find the non supportive Address families in your system

```
netstat --verbose
```

At the end, you will have something like this.

```
netstat: no support for `AF IPX' on this system.
netstat: no support for `AF AX25' on this system.
netstat: no support for `AF X25' on this system.
netstat: no support for `AF NETROM' on this system.
```

14. Display the kernel routing information using netstat -r

```
# netstat -r
Kernel IP routing table
Destination  Gateway      Genmask      Flags  MSS  Window  irtt  Iface
192.168.1.0  *            255.255.255.0  U      0 0      0  eth2
link-local   *            255.255.0.0   U      0 0      0  eth2
default      192.168.1.1  0.0.0.0       UG     0 0      0  eth2
```

Note: Use netstat -rn to display routes in numeric format without resolving for host-names.

15. Find out on which port a program is running

```
# netstat -ap | grep ssh
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp      1    0 dev-db:ssh      101.174.100.22:39213  CLOSE_WAIT -
tcp      1    0 dev-db:ssh      101.174.100.22:57643  CLOSE_WAIT -
```

Find out which process is using a particular port:

```
# netstat -an | grep ':80'
```

16. Show the list of network interfaces

```
# netstat -i
Kernel Interface table
Iface  MTU  Met  RX-OK RX-ERR RX-DRP RX-OVR  TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0    1500  0    0    0    0  0    0    0    0  0  BMU
eth2    1500  0   26196  0    0  0   26883  6    0    0  BMRU
lo      16436 0    4    0    0  0    4    0    0    0  LRU
```

Display extended information on the interfaces (similar to ifconfig) using netstat -ie:

```
# netstat -ie
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 00:10:40:11:11:11
          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)
          Memory:f6ae0000-f6b00000
```

"ping" command

1. Increase or Decrease the Time Interval Between Packets

By default ping waits for 1 second before sending the next packet. You can increase or decrease this using option -i as shown below.

Wait for 5 seconds before sending the next packet.

```
$ ping -i 5 IP
```

Wait 0.1 seconds before sending the next packet.

```
# ping -i 0.1 IP
```

Note: Only super user can specify interval less than 0.2 seconds. If not, you'll get the following error message.

2. Check whether the local network interface is up and running

Before checking whether the peer machine is reachable, first check whether the local network network is up and running using any one of the following 3 methods.

Ping localhost using zero (0)

```
$ ping 0
```

Ping localhost using name

```
$ ping localhost
```

Ping localhost using ip

```
$ ping 127.0.0.1
```

To quit the ping command, send SIGINT signal by pressing CTRL+C. If you have not specified any option to make the ping to exit automatically, then you will be terminating using CTRL+C (SIGINT) which will show the statistics and then terminate the ping process. When everything is working properly, it should say '0% packet loss'

```
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.051/0.053/0.055/0.002 ms
```

3. Send N packets and stop

Send N packets specified with -c option and then stop. This way the ping command can exit automatically instead of pressing CTRL+C to exit.

In the following example, ping command sends 5 packets, and waits for response from the destination host. Ping will exit after receiving the response or error.

```
$ ping -c 5 google.com
```

4. Show Version and Exit

Display the current version of ping program using -V option.

```
$ ping -V
```

5. Flood the network

Super users can send hundred or more packets per second using -f option. It prints a '.' when a packet is sent, and a backspace is printed when a packet is received.

```
# ping -f localhost
```

6. Audible ping: Give beep when the peer is reachable

This option is useful for sysadmin during troubleshooting. There is no need for you to look at the ping output after each and every change. You can continue working with your changes, and when the remote machine become reachable you'll hear the beep automatically.

```
$ ping -a IP
```

Note: It can give beep only from terminal number 1 through 7 and gnome-terminal (It will not work in console).

7. Find out the IP address

You can identify the ip-address using the host name as shown below.

```
$ ping -c 1 google.com
```

8. Print Only Ping Command Summary Statistics

Use option -q to view only the ping statistics summary as shown below.

```
$ ping -c 5 -q 127.0.0.1
```

9. Change Ping Packet Size

You can change the packet size of ping command using -s option.

Example: Change the default packet size from 56 to 100.

```
$ ping -s 100 localhost
```

Ping Packet Size

In the above example, when we set the packet size to 100, it displays '128 bytes' in the output. This is because of the Ping packet header size, which is 28 bytes. So, if you specify the packet size as 100, 28 bytes for header will be added to it and 128 bytes will be sent.

Ping Bytes Sent = Ping Packet Size + Ping Header Packet Size (28 bytes)

10. Timeout -w

Ping -w option specifies the deadline to terminate the ping output. This specifies the total number of seconds the ping command should send packets to the remote host.

The following example will ping for 5 seconds. i.e ping command will exit after 5 seconds irrespective of how many packets are sent or received.

```
$ ping -w 5 localhost
```

Note: When you specify both -w, and -c, whichever comes first will terminate the ping command.

11. Online ping

Ping from different locations and check the reachability (availability or time for reaching) of your server from different locations.

If you want to do an online ping, try just ping.

12. Option -w or -c Exits Ping

```
$ ping -c 4 0 -w 2
```

13. Shorter statistics with SIGQUIT

While ping is printing the individual packet status, when you want to view the shorter statistics you can use this technique.

Pressing CTRL+| (Control key followed by pipe symbol) for the shows the summary in between, and continues with it packet sending and receiving process.

```
$ ping -w 100 localhost
```

14. Specify path for ping to send the packet

You can also specify through which path the ping should send the packet to destination.

```
$ ping hop1 hop2 hop3 .. hopN destination
```

```
$ ping 192.168.3.33 192.168.7.1 192.168.4.45
```

Note: If one of the hop in the path is not reachable then you will have failure in pinging.

15. Record and print route of how ECHO_REQUEST sent and ECHO_REPLY received

It records, and prints the network route through which the packet is sent and received. This is useful for network engineers who wish to know how the packet is sent and received.

```
$ ping -R 192.168.1.63
```

“traceroute” command

1. How to run traceroute?

```
$ traceroute <server-name>
```

The server-name above is the destination name or IP address

Each line gives the details of interaction with each router encountered. So the traceroute not only gives the IP addresses of the intermediate routers but also three round trip times for that particular router as for each router the traceroute commands fires three packets.

The '*' field in output

There are times when one could encounter an '*' in the output rather than a value. This depicts that the required field could not be fetched. The reason can be anything from reverse DNS lookup failure to packets not hitting the target router to packets getting lost on their way back. So we see that the reason could be many but for all these type of cases the traceroute utility provides an * in the output.

2. Disable IP address and host name mapping

Traceroute provides an option through which the mapping of IP addresses with host name (that traceroute tries) is disabled. The option for doing this is '-n'. The following example illustrates this :

```
$ traceroute <server-name> -n
```

3. Configure Response Wait Time

The time for which traceroute utility waits after issuing a probe can also be configured. This can be done through '-w' option that it provides. The -w option expects a value which the utility will take as the response time to wait for. In this example, the wait time is 0.1 seconds and the traceroute utility was unable to wait for any response and it printed all the *'s.

```
$ traceroute google.com -w 0.1
traceroute to google.com (74.125.236.101), 30 hops max, 60 byte packets
 1 * * *
 2 * * *
 3 * * *
 ..
26 * * *
27 * * *
28 * * *
29 * * *
30 * * *
```

So we see that traceroute tried 30 attempts (the max hop attempts) and then gave up as no ICMP packet was received in 0.1 seconds.

4. Configure Number of Queries per Hop

The traceroute utility sends 3 packets per hop to provide 3 round trip times. This default value of 3 is configurable using the option '-q'. This option expects an integer which it sets as new value of number of probes per hop.

```
$ traceroute google.com -q 5
traceroute to google.com (173.194.36.46), 30 hops max, 60 byte packets
 1 220.224.141.129 (220.224.141.129) 91.579 ms 91.497 ms 91.458
ms 91.422 ms 91.385 ms
 2 115.255.239.65 (115.255.239.65) 91.356 ms 91.325 ms 98.868 ms 98.848
ms 98.829 ms
 3 124.124.251.245 (124.124.251.245) 94.581 ms 107.083 ms 107.044
ms 107.017 ms 106.981 ms
 4 115.255.239.45 (115.255.239.45) 106.948 ms 106.918 ms 144.432
ms 144.412 ms 144.392 ms
 5 72.14.212.118 (72.14.212.118) 115.565 ms 115.485 ms 115.446
ms 115.408 ms 115.381 ms
 6 72.14.232.202 (72.14.232.202) 115.351 ms 87.232 ms 117.157
ms 117.123 ms 117.049 ms
 7 209.85.241.189 (209.85.241.189) 126.998 ms 126.973 ms 126.950
ms 126.929 ms 126.912 ms
 8 bom04s02-in-f14.1e100.net (173.194.36.46) 126.889 ms 95.526 ms 95.450
ms 95.418 ms 105.392 ms
```

So we see that after configuring the number of probes to 5, the output started showing five round trip times per hop.

5. Configure the TTL value to start with

Traceroute utility is flexible enough to accept the TTL value that the user wants to start the utility with. By default its value is 1 which means it starts off with the first router in the path but using the '-f' option (which expects the new value of TTL) a new value of the TTL field can be set. For example, I tried a normal traceroute operation and then tried a traceroute with a different TTL value.

```
$ traceroute google.com
traceroute to google.com (74.125.236.132), 30 hops max, 60 byte packets
 1 220.224.141.129 (220.224.141.129) 89.181 ms 101.540 ms 101.503 ms
 2 115.255.239.65 (115.255.239.65) 101.468 ms 101.431 ms 101.324 ms
 3 124.124.251.245 (124.124.251.245) 121.373 ms 121.350 ms 158.694 ms
 4 115.255.239.45 (115.255.239.45) 101.223 ms 141.135 ms 123.932 ms
 5 72.14.212.118 (72.14.212.118) 123.867 ms 123.832 ms 123.802 ms
 6 72.14.232.202 (72.14.232.202) 123.773 ms 123.742 ms 587.812 ms
 7 216.239.48.179 (216.239.48.179) 587.723 ms 587.681 ms 587.642 ms
 8 bom03s02-in-f4.1e100.net (74.125.236.132) 577.548 ms 577.524
ms 587.512 ms
```

```
$ traceroute google.com -f 8
traceroute to google.com (74.125.236.129), 30 hops max, 60 byte packets
8  bom03s02-in-f1.1e100.net (74.125.236.129) 96.961 ms 96.886 ms 96.84
```

"ps" command

1. List Currently Running Processes (ps -ef, ps -aux)

Its a commonly used example with a ps command to list down all the process which are currently running in a machine. .

```
$ ps -ef
```

Where:

- -e to display all the processes.
- -f to display full format listing.

In case of BSD machines, you can use 'ps -aux' will give the details about all the process as shown above.

```
$ ps -aux
```

2. List the Process based on the UID and Commands (ps -u, ps -C)

Use -u option to displays the process that belongs to a specific username. When you have multiple username, separate them using a comma.

```
$ ps -f -u apache
```

Often ps is used with grep like "ps -aux | grep command" to get the list of process with the given command.

But ps command itself has an option to achieve the same.

```
$ ps -f -C test.sh
```

3. List the processes based on PIDs or PPIDs (ps -p, ps -ppid)

Each process will be assigned with the unique Process ID (PID).

When you launch some application, it might fork number of processes and each sub process will have its own PID. So, each process will have its own process id and parent processid.

For all the processes that a process forks will have the same PPID (parent process identifier). The following method is used to get a list of processes with a particular PPID.

```
$ ps -f --ppid 9576
```

The following example is to list the processes which has given PID.

```
$ ps -f -p 25009,7258,2426
```

4. List Processes in a Hierarchy (ps -forest)

The example below display the process Id and commands in a hierarchy. -forest is an argument to ps command which displays ASCII art of process tree. From this tree, we can identify which is the parent process and the child processes it forked in a recursive manner.

```
$ ps -e -o pid,args --forest
468  \_ sshd: root@pts/7
514  |  \_ -bash
17484 \_ sshd: root@pts/11
17513 |  \_ -bash
24004 |    \_ vi ./790310__11117/journal
15513 \_ sshd: root@pts/1
15522 |  \_ -bash
4280  \_ sshd: root@pts/5
4302  |  \_ -bash
```

5. List elapsed wall time for processes (ps -o pid,etime=)

If you want the get the elapsed time for the processes which are currently running ps command provides etime which provides the elapsed time since the process was started, in the form [[dd-]hh:]mm:ss.

The below command displays the elapsed time for the process IDs 1 (init) and process id 29675.

For example "10-22:13:29" in the output represents the process init is running for 10days, 22hours,13 minutes and 29seconds. Since init process starts during the system startup, this time will be same as the output of the 'uptime' command.

```
# ps -p 1,29675 -o pid,etime=
PID
1 10-22:13:29
```

29675 1-02:58:46

6. List all threads for a particular process (ps -L)

You can get a list of threads for the processes. When a process hangs, we might need to identify the list of threads running for a particular process as shown below.

```
$ ps -C java -L -o pid,tid,pcpu,state,nlwp,args
```

-L option is used to display the list of threads for a process which has the command given. And it also displays nlwp, which represents number of light weight processes. In the above example, a total of 15 java threads are running.

7. Finding memory Leak (ps -sort pmem)

A memory leak, technically, is an ever-increasing usage of memory by an application.

With common desktop applications, this may go unnoticed, because a process typically frees any memory it has used when you close the application.

However, In the client/server model, memory leakage is a serious issue, because applications are expected to be available 24×7. Applications must not continue to increase their memory usage indefinitely, because this can cause serious issues. To monitor such memory leaks, we can use the following commands.

```
$ ps aux --sort pmem
```

In the above ps command, -sort option outputs the highest %MEM at bottom. Just note down the PID for the highest %MEM usage. Then use ps command to view all the details about this process id, and monitor the change over time. You had to manually repeat it or put it as a cron to a file.

“killall” command

1. Display list of all known signal names

```
killall -l
```

2. kill process owned by the specific user

```
killall -u {user} process
```

3.To make sure that the killall command ignores the case, use the -I option. Here is an example :

`killall -I <process>`

4. killall to terminate processes interactively,

`killall -i <process>`

5. killall to carry out its work quietly, you can use the -q option :

`killall -q <process>`

6. Kill process by given time

Pass the -y TIME option to kill processes younger than given TIME. Pass the -o TIME option to kill processes older than given TIME. TIME can be expressed in the following float format then the unit:

1. s for seconds
2. m for minutes
3. h for hours
4. d for days
5. w for weeks
6. M for Months
7. y for years

“kill” command

1. Kill command to forcefully kill a process in UNIX

kill -9 is used to forcefully terminate a process in Unix. Here is syntax of kill command in UNIX.

`kill -9 PID`

2. Unix kills command to kill multiple processes

With kill command in UNIX you can specify multiple PID at same time and all process will be signaled or if signal is KILL they get killed like below kill command in UNIX

Syntax of kill in UNIX for killing multiple processes

`kill -9 PID1 PID2`

3. Kill command in UNIX to find Signal name

Kill command can also show you name of Signal if you rung it with option "-l". For example "9" is KILL signal while "3" is QUIT signal.


```
kill -l 3  
kill -9 l
```

4. Printing all signals supported by kill in UNIX

You can use `kill -l` to list down all signals supported by kill command in UNIX as shown in below example:

```
kill -l
```

5. Sending signals using -s option of kill command in UNIX.

Instead of specifying number you can specify name of signal you are sending to other process with kill command option "-s". Here is an example of using Kill command in UNIX with signal code.

```
kill -s KILL PID
```

“chkconfig” Command

1. Check Service Status

With '*chkconfig*', you can determine whether a service is configured for startup, using the '*exit status*' of the command. For this, just mention the service name as an argument to '*chkconfig*' command. If the service is configured for startup, it's exit status will be '**0**', if not then it will be '**1**' (or non-zero).

```
Chkconfig <service-name>
```

2. Display configuration status of all services

When '*chkconfig*' is executed with option `--list`, it will display all the services and their startup configuration status.

```
Chkconfig --list
```

With the combination of *chkconfig* and *grep*, you can find all the services which have been configured to start at a particular run-level, as follows:

```
chkconfig --list | grep 5:on
```

3. Adding and Deleting Services

'*chkconfig*' command with the options `--add` and `--del` would add and delete, respectively, services would be turned on the run-levels 2 through 5.

```
chkconfig --add <service name> (add it to the startup configuration.)
```

`chkconfig --del <service-name>` (remove the service from the startup list.)

4. Add or Remove a Service at a particular run-level

'chkconfig' command when executed with `--add` or `--del` options, it would impact all the run-levels from 2 through 5. In order to turn on or off any particular service at a particular run-level, you can use:

`chkconfig --level 245 <service-name> <on/off>`

"crontab" command

1. Scheduling a Job For a Specific Time

The basic usage of cron is to execute a job in a specific time as shown below. This will execute the Full backup shell script (full-backup) on **10th June 08:30 AM**.

Please note that the time field uses 24 hours format. So, for 8 AM use 8, and for 8 PM use 20.

```
30 08 10 06 * /home/ramesh/full-backup
```

2. Schedule a Job For More Than One Instance (e.g. Twice a Day)

The following script take a incremental backup twice a day every day.

This example executes the specified incremental backup shell script (incremental-backup) at 11:00 and 16:00 on every day. The comma separated value in a field specifies that the command needs to be executed in all the mentioned time.

```
00 11,16 * * * /home/ramesh/bin/incremental-backup
```

3. Schedule a Job for Specific Range of Time (e.g. Only on Weekdays)

If you wanted a job to be scheduled for every hour with in a specific range of time then use the following.

Cron Job everyday during working hours

This example checks the status of the database everyday (including weekends) during the working hours 9 a.m – 6 p.m

```
00 09-18 * * * /home/ramesh/bin/check-db-status
```

4. How to View Crontab Entries?

To view your crontab entries type `crontab -l` from your unix account as shown below.

```
Crontab -l
```

5. How to Edit Crontab Entries?

To edit a crontab entries, use `crontab -e` as shown below. By default this will edit the current logged-in users crontab.

```
Crontab -e
```

5. Remove an Entry & Prompt Before Deleting Crontab

`crontab` with `-i` option will prompt you confirmation from user before deleting user's crontab.

```
Crontab -i -r
```

“lsof” command

1. Introduction to lsof

Simply typing `lsof` will provide a list of all open files belonging to all active processes.

```
# lsof
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE NAME
---------	-----	------	----	------	--------	----------	-----------

By default One file per line is displayed. Most of the columns are self explanatory. We will explain the details about couple of cryptic columns (FD and TYPE).

FD – Represents the file descriptor. Some of the values of FDs are,

- `cwd` – Current Working Directory

- txt – Text file
- mem – Memory mapped file
- mmap – Memory mapped device
- NUMBER – Represent the actual file descriptor. The character after the number i.e '1u', represents the mode in which the file is opened. r for read, w for write, u for read and write.

TYPE – Specifies the type of the file. Some of the values of TYPEs are,

- REG – Regular File
- DIR – Directory
- FIFO – First In First Out
- CHR – Character special file

For a complete list of FD & TYPE, refer man lsof.

2. List processes which opened a specific file

You can list only the processes which opened a specific file, by providing the filename as arguments.

```
# lsof /var/log/syslog
```

3. List opened files under a directory

You can list the processes which opened files under a specified directory using '+D' option. +D will recurse the sub directories also. If you don't want lsof to recurse, then use '+d' option.

```
# lsof +D /var/log/
```

4. List opened files based on process names starting with

You can list the files opened by process names starting with a string, using '-c' option. -c followed by the process name will list the files opened by the process starting with that processes name. You can give multiple -c switch on a single command line.

```
# lsof -c ssh -c init
```

5. List processes using a mount point

Sometime when we try to umount a directory, the system will say "Device or Resource Busy" error. So we need to find out what are all the processes using the mount point and kill those processes to umount the directory. By using lsof we can find those processes.

```
# lsof /home
```

The following will also work.

```
# lsof +D /home/
```

6. List files opened by a specific user

In order to find the list of files opened by a specific users, use `'-u'` option.

```
# lsof -u tux
```

Sometimes you may want to list files opened by all users, except some 1 or 2. In that case you can use the `'^'` to exclude only the particular user as follows

```
# lsof -u ^tux
```

The above command listed all the files opened by all users, except user `'lakshmanan'`.

7. List all open files by a specific process

You can list all the files opened by a specific process using `'-p'` option. It will be helpful sometimes to get more information about a specific process.

```
# lsof -p 1753
```

8. Kill all process that belongs to a particular user

When you want to kill all the processes which has files opened by a specific user, you can use `'-t'` option to list output only the process id of the process, and pass it to kill as follows

```
# kill -9 `lsof -t -u tux`
```

The above command will kill all process belonging to user `'tux'`, which has files opened.

Similarly you can also use `'-t'` in many ways. For example, to list process id of a process which opened `/var/log/syslog` can be done by

```
# lsof -t /var/log/syslog
```

9. Combine more list options using OR/AND

By default when you use more than one list option in lsof, they will be ORed. For example,

```
# lsof -u tux -c init
```

The above command uses two list options, `-u` and `-c`. So the command will list process belongs to user `tux` as well as process name starts with `init`.

But when you want to list a process belongs to user `tux` and the process name starts with `init`, you can use `-a` option.

```
# lsof -u tux -c init -a
```

The above command will not output anything, because there is no such process named `init` belonging to user `tux`.

10. Execute lsof in repeat mode

lsof also support Repeat mode. It will first list files based on the given parameters, and delay for specified seconds and again list files based on the given parameters. It can be interrupted by a signal.

Repeat mode can be enabled by using `-r` or `+r`. If `+r` is used then, the repeat mode will end when no open files are found. `-r` will continue to list,delay,list until a interrupt is given irrespective of files are opened or not.

Each cycle output will be separated by using `=====`. You also also specify the time delay as `-r` | `+r`.

```
# lsof -u tux -c init -a -r5
```

In the above output, for the first 5 seconds, there is no output.

11. List all network connections

You can list all the network connections opened by using `-i` option.

```
# lsof -i
```

You can also use `-i4` or `-i6` to list only `IPV4` or `IPV6` respectively.

12. List all network files in use by a specific process

You can list all the network files which is being used by a process as follows

```
# lsof -i -a -p 234
```

You can also use the following

```
# lsof -i -a -c ssh
```

The above command will list the network files opened by the processes starting with ssh.

13. List processes which are listening on a particular port

You can list the processes which are listening on a particular port by using `'-i'` with `':'` as follows

```
# lsof -i :25
```

14. List all TCP or UDP connections

You can list all the TCP or UDP connections by specifying the protocol using `'-i'`.

```
# lsof -i tcp; lsof -i udp;
```

15. List all Network File System (NFS) files

You can list all the NFS files by using `'-N'` option. The following lsof command will list all NFS files used by user `'tux'`.

```
# lsof -N -u tux -a
```

"unset" command

unset - unset values and attributes of variables and functions

SYNOPSIS

```
unset [-fv] name ...
```

DESCRIPTION

Each variable or function specified by name shall be unset.

If `-v` is specified, name refers to a variable name and the shell shall unset it and remove

it from the environment. Read-only variables cannot be unset.

If `-f` is specified, `name` refers to a function and the shell shall unset the function definition.

If neither `-f` nor `-v` is specified, `name` refers to a variable; if a variable by that name does not exist, it is unspecified whether a function by that name, if any, shall be unset.

Unsetting a variable or function that was not previously set shall not be considered an error and does not cause the shell to abort.

"strace" command

1. Trace the Execution of an Executable

You can use `strace` command to trace the execution of any executable.

```
$ strace <command-name>
```

2. Trace a Specific System Calls in an Executable Using Option `-e`

By default, `strace` displays all system calls for the given executable. To display only a specific system call, use the `strace -e` option as shown below.

```
$ strace -e <system-call> <command-name>
```

If you want to trace multiple system calls use the `"-e trace="` option. The following example displays both `open` and `read` system calls.

```
$ strace -e trace=open,read ls /home
```

3. Save the Trace Execution to a File Using Option `-o`

The following examples store the `strace` output to `output.txt` file.

```
$ strace -o output.txt ls
Desktop Documents Downloads examples.desktop libflashplayer.so
Music output.txt Pictures Public Templates Ubuntu_OS Videos
```

4. Execute Strace on a Running Linux Process Using Option `-p`

You could execute `strace` on a program that is already running using the process id. First, identify the PID of a program using `ps` command.

For example, if you want to do strace on the firefox program that is currently running, identify the PID of the firefox program.

```
$ ps -C firefox-bin
  PID TTY          TIME CMD
 1725 ?        00:40:50 firefox-bin
```

Use strace -p option as shown below to display the strace for a given process id.

```
$ sudo strace -p 1725 -o firefox_trace.txt
```

```
$ tail -f firefox_trace.txt
```

Now the execution trace of firefox process will be logged into firefox_trace.txt text file. You can tail this text file to watch the live trace of the firefox executable.

Strace will display the following error when your user id does not match the user id of the given process.

```
$ strace -p 1725 -o output.txt
attach: ptrace(PTRACE_ATTACH, ...): Operation not permitted
Could not attach to process.  If your uid matches the uid of the target
process, check the setting of /proc/sys/kernel/yama/ptrace_scope, or try
again as the root user.  For more details, see /etc/sysctl.d/10-ptrace.conf
```

5. Print Timestamp for Each Trace Output Line Using Option -t

To print the timestamp for each strace output line, use the option -t as shown below.

```
$ strace -t -e open ls /home
```

6. Print Relative Time for System Calls Using Option -r

Strace also has the option to print the execution time for each system calls as shown below.

```
$ strace -r ls
0.000000 execve("/bin/ls", ["ls"], [/ * 37 vars */]) = 0
0.000846 brk(0)                                = 0x8418000
0.000143 access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or
directory)
0.000163 mmap2(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb787b000
0.000119 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or
directory)
0.000123 open("/etc/ld.so.cache", O_RDONLY) = 3
0.000099 fstat64(3, {st_mode=S_IFREG|0644, st_size=67188, ...}) = 0
```

```
0.000155 mmap2(NULL, 67188, PROT_READ, MAP_PRIVATE, 3, 0) =
0xb786a000
```

```
...
```

```
...
```

7. Generate Statistics Report of System Calls Using Option -c

Using option -c, strace provides useful statistical report for the execution trace. The "calls" column in the following output indicated how many times that particular system call was executed.

```
$ strace -c ls /home
```

% time	seconds	usecs/call	calls	errors	syscall
-nan	0.000000	0	9		read
-nan	0.000000	0	1		write
-nan	0.000000	0	11		open
-nan	0.000000	0	13		close
-nan	0.000000	0	1		execve
-nan	0.000000	0	9	9	access
-nan	0.000000	0	3		brk
-nan	0.000000	0	2		ioctl
-nan	0.000000	0	3		munmap
-nan	0.000000	0	1		uname
-nan	0.000000	0	11		mprotect
-nan	0.000000	0	2		rt_sigaction
-nan	0.000000	0	1		rt_sigprocmask
-nan	0.000000	0	1		getrlimit
-nan	0.000000	0	25		mmap2
-nan	0.000000	0	1		stat64
-nan	0.000000	0	11		fstat64
-nan	0.000000	0	2		getdents64
-nan	0.000000	0	1		fcntl64
-nan	0.000000	0	2	1	futex
-nan	0.000000	0	1		set_thread_area
-nan	0.000000	0	1		set_tid_address
-nan	0.000000	0	1		statfs64
-nan	0.000000	0	1		set_robust_list
100.00	0.000000		114	10	total

Day3

"awk" command

1 Print all the lines from a file.

By default, awk prints all lines of a file , so to print every line of above created file use below command :

```
awk '{print;}' awk_file
```

2 Print only Specific field like 2nd & 3rd.

```
awk -F "," '{print $2, $3;}' awk_file
```

3 Print the lines which matches the pattern

I want to print the lines which contains the word "Hari & Ram"

```
awk '/Hari|Ram/' awk_file
```

4 How do we find unique values in the first column of name

```
awk -F, '{a[$1];}END{for (i in a)print i;}' awk_file
```

5 How to find the sum of data entry in a particular column .

```
awk -F, '$1=="Item1"{x+=$2;}END{print x}' awk_file
```

6 How to find the total of all numbers in a column.

For eg we take the 2nd and the 3rd column.

```
awk -F"," '{x+=$2;}END{print x}' awk_file
```

7 How to find the sum of individual group records.

Eg if we consider the first column than we can do the summation for the first column based on the items

```
awk -F, '{a[$1]+=$2;}END{for(i in a)print i, "a[i];}' awk_file
```

8 How to find the sum of all entries in second column and append it to the end of the file.

```
awk -F"," '{x+=$2;y+=$3;print}END{print "Total,"x,y}' awk_file
```

9 How to find the count of entries against every column based on the first column:

```
awk -F, '{a[$1]++;}END{for (i in a)print i, a[i];}' awk_file
```

10 How to print only the first record of every group:

```
awk -F, '!a[$1]++' awk_file
```

11 How to populate each column names along with their corresponding data.

```
awk 'BEGIN{print "Names\ttotal\tPPT\tDoc\txls"}{printf "%-s\t%d\t%d\t%d\t%d\n", $1,$2,$3,$4,$5}' datafile
```

12 How to change the Field Separator

As we can see space is the field separator in the datafile , in the below example we will change field separator from space to "|"

```
awk 'BEGIN{OFS="|"}{print $1,$2,$3,$4,$5}' datafile
```

"sed" command

<http://how-to.linuxcareer.com/learning-linux-commands-sed>

"grep" command

1. Search for the given string in a single file

The basic usage of grep command is to search for a specific string in the specified file as shown below.

```
grep "literal_string" filename
```

2. Checking for the given string in multiple files.

```
grep "string" FILE_PATTERN
```

3. Case insensitive search using grep -i

```
grep -i "string" FILE
```

This is also a basic usage of the grep. This searches for the given string/pattern case insensitively.

4. Match regular expression in files

```
grep "REGEX" filename
```

This is a very powerful feature, if you can use regular expression effectively.

From documentation of grep: A regular expression may be followed by one of several repetition operators:

- ? The preceding item is optional and matched at most once.
- * The preceding item will be matched zero or more times.
- + The preceding item will be matched one or more times.
- {n} The preceding item is matched exactly n times.
- {n,} The preceding item is matched n or more times.
- {,m} The preceding item is matched at most m times.
- {n,m} The preceding item is matched at least n times, but not more than m times.

5. Checking for full words, not for sub-strings using grep -w

If you want to search for a word, and to avoid it to match the substrings use -w option. Just doing out a normal search will show out all the lines.

The following example is the regular grep where it is searching for "is". When you search for "is", without any option it will show out "is", "his", "this" and everything which has the substring "is".

```
$ grep -i "is" demo_file
```

```
THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE.
```

```
this line is the 1st lower case line in this file.
```

```
This Line Has All Its First Character Of The Word With Upper Case.
```

```
Two lines above this line is empty.
```

And this is the last line.

The following example is the WORD grep where it is searching only for the word "is". Please note that this output does not contain the line "This Line Has All Its First Character Of The Word With Upper Case", even though "is" is there in the "This", as the following is looking only for the word "is" and not for "this".

```
$ grep -iw "is" demo_file
THIS LINE IS THE 1ST UPPER CASE LINE IN THIS FILE.
this line is the 1st lower case line in this file.
Two lines above this line is empty.
And this is the last line.
```

6. Displaying lines before/after/around the match using grep -A, -B and -C

When doing a grep on a huge file, it may be useful to see some lines after the match. You might feel handy if grep can show you not only the matching lines but also the lines after/before/around the match.

Display N lines after match

-A is the option which prints the specified N lines after the match as shown below.

```
grep -A <N> "string" FILENAME
```

Display N lines before match

-B is the option which prints the specified N lines before the match.

```
grep -B <N> "string" FILENAME
```

Display N lines around match

-C is the option which prints the specified N lines before the match. In some occasion you might want the match to be appeared with the lines from both the side. This options shows N lines in both the side(before & after) of match.

```
grep -C <N> "string" FILENAME
```

7. Highlighting the search using GREP_OPTIONS

As grep prints out lines from the file by the pattern / string you had given, if you wanted it to highlight which part matches the line, then you need to follow the following way.

When you do the following export you will get the highlighting of the matched

searches. In the following example, it will highlight all the this when you set the GREP_OPTIONS environment variable as shown below.

```
$ export GREP_OPTIONS='--color=auto' GREP_COLOR='100;8'
```

8. Searching in all files recursively using grep -r

When you want to search in all the files under the current directory and its sub directory. -r option is the one which you need to use. The following example will look for the string "tux" in all the files in the current directory and all its subdirectory.

```
$ grep -r "tux" *
```

9. Invert match using grep -v

You had different options to show the lines matched, to show the lines before match, and to show the lines after match, and to highlight match. So definitely You'd also want the option -v to do invert match.

When you want to display the lines which does not matches the given string/pattern, use the option -v as shown below. This example will display all the lines that did not match the word "go".

```
grep -v "string" FILENAME
```

10. display the lines which does not matches all the given pattern.

```
grep -v -e "pattern" -e "pattern"
```

11. Counting the number of matches using grep -c

When you want to count that how many lines matches the given pattern/string, then use the option -c.

```
grep -c "pattern" filename
```

12. Display only the file names which matches the given pattern using grep -l

If you want the grep to show out only the file names which matched the given pattern, use the -l (lower-case L) option.

When you give multiple files to the grep as input, it displays the names of file

which contains the text that matches the pattern, will be very handy when you try to find some notes in your whole directory structure.

```
$ grep -l this demo_*  
demo_file  
demo_file1
```

13. Show only the matched string

By default grep will show the line which matches the given pattern/string, but if you want the grep to show out only the matched string of the pattern then use the -o option.

It might not be that much useful when you give the string straight forward. But it becomes very useful when you give a regex pattern and trying to see what it matches as

```
$ grep -o "is.*line" demo_file  
is line is the 1st lower case line  
is line  
is is the last line
```

14. Show the position of match in the line

When you want grep to show the position where it matches the pattern in the file, use the following options as

```
grep -o -b "pattern" file
```

15. Show line number while displaying the output using grep -n

To show the line number of file with the line matched. It does 1-based line numbering for each file. Use -n option to utilize this feature.

```
$ grep -n "go" demo_text  
5: * e - go to the end of the current word.  
6: * E - go to the end of the current WORD.  
7: * b - go to the previous (before) word.  
8: * B - go to the previous (before) WORD.  
9: * w - go to the next word.  
10: * W - go to the next WORD.
```


“find” command

1. List all files in current and sub directories

This command lists out all the files in the current directory as well as the subdirectories in the current directory.

```
$ find
```

The command is same as the following

```
$ find .  
$ find . -print
```

2. Search specific directory or path

The following command will look for files in the test directory in the current directory. Lists out all files by default

```
$ find ./test
```

The following command searches for files by their name.

```
$ find ./test -name "abc.txt"  
./test/abc.txt
```

We can also use wildcards

```
$ find ./test -name "*.php"  
./test/subdir/how.php  
./test/cool.php
```

Note that all sub directories are searched recursively. So this is a very powerful way to find all files of a given extension.

Trying to search the "/" directory which is the root, would search the entire file system including mounted devices and network storage devices. So be careful. Of course you can press Ctrl + c anytime to stop the command.

When specifying the directory ("./test" in this example), its fine to omit the trailing slash. However, if the directory is actually a symlink to some other location then you MUST specify the trailing slash for it to work properly (find ./test/ ...)

Ignore the case

It is often useful to ignore the case when searching for file names. To ignore the case, just use the "iname" option instead of the "name" option.

```
$ find ./test -iname "*.Php"  
./test/subdir/how.php  
./test/cool.php
```

Its always better to wrap the search term (name parameter) in double or single quotes. Not doing so will seem to work sometimes and give strange results at other times.

3. Limit depth of directory traversal

The find command by default travels down the entire directory tree recursively, which is time and resource consuming. However the depth of directory traversal can be specified. For example we don't want to go more than 2 or 3 levels down in the sub directories. This is done using the maxdepth option.

```
$ find ./test -maxdepth 2 -name "*.php"  
./test/subdir/how.php  
./test/cool.php
```

```
$ find ./test -maxdepth 1 -name *.php  
./test/cool.php
```

The second example uses maxdepth of 1, which means it will not go lower than 1 level deep, either only in the current directory.

This is very useful when we want to do a limited search only in the current directory or max 1 level deep sub directories and not the entire directory tree which would take more time.

Just like maxdepth there is an option called mindepth which does what the name suggests, that is, it will go atleast N level deep before searching for the files.

4. Invert match

It is also possible to search for files that do not match a given name or pattern. This is helpful when we know which files to exclude from the search.

```
$ find ./test -not -name "*.php"  
./test  
./test/abc.txt  
./test/subdir
```

So in the above example we found all files that do not have the extension of php, either non-php files. The find command also supports the exclamation mark inplace of not.

```
find ./test ! -name "*.php"
```

5. Combine multiple search criterias

It is possible to use multiple criterias when specifying name and inverting. For example

```
$ find ./test -name 'abc*' ! -name '*.php'
./test/abc.txt
./test/abc
```

The above find command looks for files that begin with abc in their names and do not have a php extension. This is an example of how powerful search expressions can be build with the find command.

OR operator

When using multiple name criterias, the find command would combine them with AND operator, which means that only those files which satisfy all criterias will be matched. However if we need to perform an OR based matching then the find command has the "o" switch.

```
$ find -name '*.php' -o -name '*.txt'
./abc.txt
./subdir/how.php
./abc.php
./cool.php
```

The above command search for files ending in either the php extension or the txt extension.

6. Search only files or only directories

Sometimes we want to find only files or only directories with a given name. Find can do this easily as well.

```
$ find ./test -name abc*
./test/abc.txt
./test/abc
```

Only files

```
$ find ./test -type f -name "abc*"
```

```
./test/abc.txt
```

Only directories

```
$ find ./test -type d -name "abc*"
./test/abc
```

Quite useful and handy!

7. Search multiple directories together

So lets say you want to search inside 2 separate directories. Again, the command is very simple

```
$ find ./test ./dir2 -type f -name "abc*"
./test/abc.txt
./dir2/abcdefgh.txt
```

Check, that it listed files from 2 separate directories.

8. Find hidden files

Hidden files on linux begin with a period. So its easy to mention that in the name criteria and list all hidden files.

```
$ find ~ -type f -name ".*"
```

9. Find files with certain permissions

The find command can be used to find files with a specific permission using the "perm" option. The following command searches for files with the permission 0664

```
$ find . -type f -perm 0664
./abc.txt
./subdir/how.php
./abc.php
./cool.php
```

This can be useful to find files with wrong permissions which can lead to security issues. Inversion can also be applied to permission checking.

```
$ find . -type f ! -perm 0777
./abc.txt
./subdir/how.php
./abc.php
./cool.php
```

10. Find files with sgid/suid bits set

The "perm" option of find command accepts the same mode string like chmod. The following command finds all files with permission 644 and sgid bit set.

```
# find / -perm 2644
```

Similarly use 1664 for sticky bit. The perm option also supports using an alternative syntax instead of octal numbers.

```
$ find / -maxdepth 2 -perm /u=s 2>/dev/null
/bin/mount
/bin/su
/bin/ping6
/bin/fusermount
/bin/ping
/bin/umount
/sbin/mount.ecryptfs_private
```

Note that the "2>/dev/null" removes those entries that have an error of "Permission Denied"

11. Find readonly files

Find all Read Only files.

```
$ find /etc -maxdepth 1 -perm /u=r
/etc
/etc/thunderbird
/etc/brltty
/etc/dkms
/etc/phpmyadmin
... output truncated ...
```

12. Find executable files

The following command will find executable files

```
$ find /bin -maxdepth 2 -perm /a=x
/bin
/bin/preseed_command
/bin/mount
/bin/zfgrep
/bin/tempfile
... output truncated ...
```

13. Find files belonging to particular user

To find all or single file called tecmint.txt under /root directory of owner root.

```
$ find . -user bob
```

```
.  
./abc.txt  
./abc  
./subdir  
./subdir/how.php  
./abc.php
```

We could also specify the name of the file or any name related criteria along with user criteria

```
$ find . -user bob -name '*.php'
```

Its very easy to see, how we can build up criteria after criteria to narrow down our search for matching files.

14. Search files belonging to group

Find all files that belong to a particular group.

```
# find /var/www -group developer
```

Did you know you could search your home directory by using the ~ symbol ?

```
$ find ~ -name "hidden.php"
```

15. Find files modified N days back

To find all the files which are modified 50 days back.

```
# find / -mtime 50
```

16. Find files accessed in last N days

Find all files that were accessed in the last 50 days.

```
# find / -atime 50
```

17. Find files modified in a range of days

Find all files that were modified between 50 to 100 days ago.

```
# find / -mtime +50 -mtime -100
```

18. Find files changed in last N minutes.

Find files modified within the last 1 hour.

```
$ find /home/bob -cmin -60
```

19. Files modified in last hour

To find all the files which are modified in last 1 hour.

```
# find / -mmin -60
```

20. Find Accessed Files in Last 1 Hour

To find all the files which are accessed in last 1 hour.

```
# find / -amin -60
```

21. Find files of given size

To find all 50MB files, use.

```
# find / -size 50M
```

22. Find files in a size range

To find all the files which are greater than 50MB and less than 100MB.

```
$ find / -size +50M -size -100M
```

23. Find largest and smallest files

The find command when used in combination with the ls and sort command can be used to list out the largest files.

The following command will display the 5 largest file in the current directory and its subdirectory. This may take a while to execute depending on the total number of files the command has to process.

```
$ find . -type f -exec ls -s {} \; | sort -n -r | head -5
```

Similarity when sorted in ascending order, it would show the smallest files first

```
$ find . -type f -exec ls -s {} \; | sort -n | head -5
```

24. Find empty files and directories

The following command uses the "empty" option of the find command, which finds all files that are empty.

```
# find /tmp -type f -empty
```

To file all empty directories use the type "d".

```
$ find ~/ -type d -empty
```

Really very simple and easy

25. List out the found files

Lets say we found files using find command, and now want to list them out as the ls command would have done. This is very easy.

```
$ find . -exec ls -ld {} \;  
drwxrwxr-x 4 enlightened enlightened 4096 Aug 11 19:01 .  
-rw-rw-r-- 1 enlightened enlightened 0 Aug 11 16:25 ./abc.txt  
drwxrwxr-x 2 enlightened enlightened 4096 Aug 11 16:48 ./abc  
drwxrwxr-x 2 enlightened enlightened 4096 Aug 11 16:26 ./subdir  
-rw-rw-r-- 1 enlightened enlightened 0 Aug 11 16:26 ./subdir/how.php  
-rw-rw-r-- 1 enlightened enlightened 29 Aug 11 19:13 ./abc.php  
-rw-rw-r-- 1 enlightened enlightened 0 Aug 11 16:25 ./cool.php
```

26. Delete all matching files or directories

The following command will remove all text files in the tmp directory.

```
$ find /tmp -type f -name "*.txt" -exec rm -f {} \;
```

The same operating can be carried out with directories, just put type d, instead of type f.

Lets take another example where we want to delete files larger than 100MB

```
$ find /home/bob/dir -type f -name *.log -size +10M -exec rm -f {} \;
```

"cut" Command

1. Select Column of Characters

To extract only a desired column from a file use -c option. The following example displays 2nd character from each line of a file test.txt

```
$ cut -c2 test.txt
```

2. Select Column of Characters using Range

Range of characters can also be extracted from a file by specifying start and end position delimited with -. The following example extracts first 3 characters of each line from a file called test.txt


```
$ cut -c1-3 test.txt
```

3. Select Column of Characters using either Start or End Position

Either start position or end position can be passed to cut command with -c option.

The following specifies only the start position before the '-'. This example extracts from 3rd character to end of each line from test.txt file.

```
$ cut -c3- test.txt
```

The following specifies only the end position after the '-'. This example extracts 8 characters from the beginning of each line from test.txt file.

```
$ cut -c-8 test.txt
```

The entire line would get printed when you don't specify a number before or after the '-' as shown below.

```
$ cut -c- test.txt
```

4. Select a Specific Field from a File

Instead of selecting x number of characters, if you like to extract a whole field, you can combine option -f and -d. The option -f specifies which field you want to extract, and the option -d specifies what is the field delimiter that is used in the input file.

The following example displays only first field of each lines from /etc/passwd file using the field delimiter : (colon). In this case, the 1st field is the username. The file

```
$ cut -d':' -f1 /etc/passwd
```

5. Select Multiple Fields from a File

You can also extract more than one fields from a file or stdout. Below example displays username and home directory of users who has the login shell as "/bin/bash".

```
$ grep "/bin/bash" /etc/passwd | cut -d':' -f1,6
```

To display the range of fields specify start field and end field as shown below. In this example, we are selecting field 1 through 4, 6 and 7

```
$ grep "/bin/bash" /etc/passwd | cut -d':' -f1-4,6,7
```

6. Select Fields Only When a Line Contains the Delimiter

In our `/etc/passwd` example, if you pass a different delimiter other than `:` (colon), `cut` will just display the whole line.

In the following example, we've specified the delimiter as `|` (pipe), and `cut` command simply displays the whole line, even when it doesn't find any line that has `|` (pipe) as delimiter.

```
$ grep "/bin/bash" /etc/passwd | cut -d'|' -f1
```

But, it is possible to filter and display only the lines that contains the specified delimiter using `-s` option.

The following example doesn't display any output, as the `cut` command didn't find any lines that has `|` (pipe) as delimiter in the `/etc/passwd` file.

```
$ grep "/bin/bash" /etc/passwd | cut -d'|' -s -f1
```

7. Select All Fields Except the Specified Fields

In order to complement the selection field list use option `-complement`.

The following example displays all the fields from `/etc/passwd` file except field 7

```
$ grep "/bin/bash" /etc/passwd | cut -d':' --complement -s -f7
```

8. Change Output Delimiter for Display

By default the output delimiter is same as input delimiter that we specify in the `cut -d` option.

To change the output delimiter use the option `-output-delimiter` as shown below. In this example, the input delimiter is `:` (colon), but the output delimiter is `#` (hash).

```
$ grep "/bin/bash" /etc/passwd | cut -d':' -s -f1,6,7 -output-delimiter='#'
```

9. Change Output Delimiter to Newline

In this example, each and every field of the `cut` command output is displayed in a separate line. We still used `-output-delimiter`, but the value is `$'\n'` which indicates that we should add a newline as the output delimiter.

```
$ grep bala /etc/passwd | cut -d':' -f1,6,7 -output-delimiter=$'\n'
```

10. Combine Cut with Other Unix Command Output

The power of `cut` command can be realized when you combine it with the `stdout` of some other Unix command.

Once you master the basic usage of cut command that we've explained above, you can wisely use cut command to solve lot of your text manipulation requirements.

The following example indicates how you can extract only useful information from the ps command output. We also showed how we've filtered the output of ps command using grep and sed before the final output was given to cut command. Here, we've used cut option -d and -f which we've explained in the above examples.

```
$ ps axu | grep python | sed 's/\s\+ / /g' | cut -d' ' -f2,11-
```

"tr" command

1. Convert lower case to upper case

The following command will convert lower case to upper case

```
$ tr [:lower:] [:upper:]
```

You can also use ranges in tr. The following command uses ranges to convert lower to upper case.

```
$ tr a-z A-Z
```

2. Translate braces into parenthesis

You can also translate from and to a file. In this example we will translate braces in a file with parenthesis.

```
$ tr '{}()' < inputfile > outputfile
```

The above command will read each character from "inputfile", translate if it is a brace, and write the output in "outputfile".

3. Translate white-space to tabs

The following command will translate all the white-space to tabs

```
$ echo "This is for testing" | tr [:space:] '\t'  
This    is    for    testing
```

4. Squeeze repetition of characters using -s

In 3, we see how to translate space with tabs. But if there are two or more spaces present continuously, then the previous command will translate each space to a tab as follows.

```
$ echo "This is for testing" | tr [:space:] '\t'  
This is for testing
```

We can use -s option to squeeze the repetition of characters.

```
$ echo "This is for testing" | tr -s [:space:] '\t'  
This is for testing
```

Similarly you can convert multiple continuous spaces with a single space

```
$ echo "This is for testing" | tr -s [:space:] ' '  
This is for testing
```

5. Delete specified characters using -d option

tr can also be used to remove particular characters using -d option.

```
$ echo "Hello Worldf" | tr -d 'o'  
Hell Wrld
```

To remove all the digits from the string, use

```
$ echo "my username is 432234" | tr -d [:digit:]  
my username is
```

6. Complement the sets using -c option

You can complement the SET1 using -c option. For example, to remove all characters except digits, you can use the following.

```
$ echo "my username is 432234" | tr -cd [:digit:]  
432234
```

7. Remove all non-printable character from a file

The following command can be used to remove all non-printable characters from a file.

```
$ tr -cd [:print:] < file.txt
```

8. Join all the lines in a file into a single line

The below command will translate all newlines into spaces and make the result as a single line.

```
$ tr -s '\n' ' ' < file.txt
```

`"uniq"` command

1. Basic Usage

```
$ uniq [-options]
```

For example, when `uniq` command is run without any option, it removes duplicate lines and displays unique lines as shown below.

```
$ uniq test
```

2. Count Number of Occurrences using `-c` option

This option is to count occurrence of lines in file.

```
$ uniq -c test
```

3. Print only Duplicate Lines using `-d` option

This option is to print only duplicate repeated lines in file.

```
$ uniq -d test
```

The above example displayed all the duplicate lines, but only once. But, this `-D` option will print all duplicate lines in file.

```
$ uniq -D test
```

4. Print only Unique Lines using `-u` option

This option is to print only unique lines in file.

```
$ uniq -u test
```

5. Limit Comparison to 'N' characters using `-w` option

This option restricts comparison to first specified 'N' characters only.

The following `uniq` command using option `'w'` is compares the first 8 characters of lines in file, and then using `'c'` option prints number of occurrences of lines of file.

```
$ uniq -c -w 8 test
```

The following `uniq` command using option `'w'` is compares first 8 characters of lines in file, and then using `'D'` option prints all duplicate lines of file.

```
$ uniq -D -w 8 test
```

6. Avoid Comparing first 'N' Characters using -s option

This option skips comparison of first specified 'N' characters.

The following uniq command using option 's' skips comparing first 2 characters of lines in file, and then using 'D' option prints all duplicate lines of file.

```
$ uniq -D -s 2 test3
```

7. Avoid Comparing first 'N' Fields using -f option

This option skips comparison of first specified 'N' fields of lines in file.

The following uniq command using option 'f' skips comparing first 2 fields of lines in file, and then using 'D' option prints all duplicate lines of file.

```
$ uniq -D -f 2 test2
```

"wc" command

1. A Basic Example of WC Command

The 'wc' command without passing any parameter will display a basic result of 'test.txt' file.

```
wc tecmint.txt
```

2. Count Number of Lines

To count number of newlines in a file use the option '-l', which prints the number of lines from a given file. Say, the following command will display the count of newlines in a file. In the output the first field assigned as count and second field is the name of file.

```
wc -l test.txt
```

```
10 test.txt
```

3. Display Number of Words

Using `-w` argument with `wc` command prints the number of words in a file. Type the following command to count the words in a file.

```
wc -w test.txt
```

4. Count Number of Bytes and Characters

When using options `-c` and `-m` with `wc` command will print the total number of bytes and characters respectively in a file.

```
wc -m/c test.txt
```

5. Display Length of Longest Line

The `wc` command allow an argument `-L`, it can be used to print out the length of longest (number of characters) line in a file. So, we have the longest character line (`Scientific Linux`) in a file.

```
wc -L test.txt
```

"sort" command

Sort Command Syntax:

```
$ sort [-options]
```

For example, here is a test file:

```
$ cat test
```

```
zzz
```

```
sss
```

```
qqq
```

```
aaa
```

```
BBB
```

```
ddd
```

```
AAA
```

And, here is what you get when sort command is executed on this file without any option. It sorts lines in test file and displays sorted output.

```
$ sort test
```

```
aaa
```

```
AAA
```

```
BBB
ddd
qqq
sss
zzz
```

1. Perform Numeric Sort using -n option

If we want to sort on numeric value, then we can use -n or --numeric-sort option.

Create the following test file for this example:

```
$ cat test
22 zzz
33 sss
11 qqq
77 aaa
55 BBB
```

The following sort command sorts lines in test file on numeric value in first word of line and displays sorted output.

```
$ sort -n test
11 qqq
22 zzz
33 sss
55 BBB
77 aaa
```

2. Sort Human Readable Numbers using -h option

If we want to sort on human readable numbers (e.g., 2K 1M 1G), then we can use -h or --human-numeric-sort option.

Create the following test file for this example:

```
$ cat test
2K
2G
1K
6T
1T
1G
2M
```


The following sort command sorts human readable numbers (i.e 1K = 1 Thousand, 1M = 1 Million, 1G = 1 Giga, 1T = 1 Tera) in test file and displays sorted output.

```
$ sort -h test
1K
2K
2M
1G
2G
1T
6T
```

3. Sort Months of an Year using -M option

If we want to sort in the order of months of year, then we can use -M or -month-sort option.

Create the following test file for this example:

```
$ cat test
sept
aug
jan
oct
apr
feb
mar11
```

The following sort command sorts lines in test file as per month order. Note, lines in file should contain at least 3 character name of month name at start of line (e.g. jan, feb, mar). If we will give, ja for January or au for August, then sort command would not consider it as month name.

```
$ sort -M test
jan
feb
mar11
apr
aug
sept
oct
```

4. Check if Content is Already Sorted using -c option

If we want to check data in text file is sorted or not, then we can use -c or -check, -check=diagnose-first option.

Create the following test file for this example:

```
$ cat test
2
5
1
6
```

The following sort command checks whether text file data is sorted or not. If it is not, then it shows first occurrence with line number and disordered value.

```
$ sort -c test
sort: test:3: disorder: 1
```

5. Reverse the Output and Check for Uniqueness using -r and -u options

If we want to get sorted output in reverse order, then we can use -r or -reverse option. If file contains duplicate lines, then to get unique lines in sorted output, "-u" option can be used.

Create the following test file for this example:

```
$ cat test
5
2
2
1
4
4
```

The following sort command sorts lines in test file in reverse order and displays sorted output.

```
$ sort -r test
5
4
4
2
2
1
```

The following sort command sorts lines in test file in reverse order and removes duplicate lines from sorted output.

```
$ sort -r -u test
5
4
2
1
```

6. Selectively Sort the Content, Customize delimiter, Write output to a file using -k, -t, -o options

If we want to sort on the column or word position in lines of text file, then "-k" option can be used. If we each word in each line of file is separated by delimiter except 'space', then we can specify delimiter using "-t" option. We can get sorted output in any specified output file (using "-o" option) instead of displaying output on standard output.

Create the following test file for this example:

```
$ cat test
aa aa zz
aa aa ff
aa aa tt
aa aa kk
```

The following sort command sorts lines in test file on the 3rd word of each line and displays sorted output.

```
$ sort -k3 test
aa aa ff
aa aa kk
aa aa tt
aa aa zz
```

```
$ cat test
aa|5a|zz
aa|2a|ff
aa|1a|tt
aa|3a|kk
```

Here, several options are used altogether. In test file, words in each line are separated by delimiter '|'. It sorts lines in test file on the 2nd word of each line on the basis of numeric value and stores sorted output into specified output file.

```
$ sort -n -t'|' -k2 test -o outfile
```

The contents of output file are shown below.

```
$ cat outfile  
aa|1a|tt  
aa|2a|ff  
aa|3a|kk  
aa|5a|zz
```

“tee” command

The tee command reads standard input, then writes the output of a program to standard output and simultaneously copies it into the specified file or files.

1. writes the output both to the screen (stdout) and to the file.

```
$ ls | tee file1.txt
```

2. You can instruct tee command to append to the file using the option `-a` as shown below

```
$ ls | tee -a fileq.txt
```

3. You can also write the output to multiple files as shown below.

```
$ ls | tee file1 file2 file3
```

4. Write the output to two commands

You can also use tee command to store the output of a command to a file and redirect the same output as an input to another command.

The following command will long list directory contents, store it in `abc.txt` and then count the no. of lines and echo it

```
$ ls -l | tee abc.txt | wc -l
```

“xargs” command

1. Xargs Basic Example

The xargs command (by default) expects the input from stdin, and executes `/bin/echo` command over the input. The following is what happens when you execute xargs without any argument, or when you execute it without combining with any other commands.

When you type xargs without any argument, it will prompt you to enter the input through stdin:

```
$ xargs
Hi,
Welcome to TGS.
```

After you type something, press ctrl+d, which will echo the string back to you on stdout as shown below.

```
$ xargs
Hi,
Welcome to TGS.Hi, Welcome to TGS.
```

2. Specify Delimiter Using -d option

Delimiters can be applied so that each character in the input is taken literally using -d option in xargs.

In the previous example, even though the input contained a \n (newline) after 'Hi,' but the echoed output did not contain the newline '\n' . So, the output in the previous example was combined into a single line.

In the following example, when you use the -d\n, it will preserve newline delimiter in the output, and display the output exactly as it was typed.

```
$ xargs -d\n
Hi,
Welcome to TGS.
```

After you type the above, press ctrl+d, which will echo the string back to you on stdout as shown below. But, this time it will preserve the newline.

```
$ xargs -d\n
Hi,
Welcome to TGS.Hi,
Welcome to TGS.
```

3. Limit Output Per Line Using -n Option

By default as explained earlier, xargs displays whatever comes to its stdin as shown below.

```
$ echo a b c d e f | xargs
a b c d e f
```

But, the output of the xargs command can be split into multiple lines using -n option.

In the following example, we used -n 3, which will display only 3 items per line in the xargs output.

```
$ echo a b c d e f | xargs -n 3
a b c
d e f
```

In the same way, you can also split the output with 2 items per line as shown below using -n 2.

```
$ echo a b c d e f | xargs -n 2
a b
c d
e f
```

4. Prompt User Before Execution using -p option

Using option -p, you can confirm the execution of the xargs command from the user.

Considering the previous example, if we want to confirm each execution of the /bin/echo command by the user, use the -p option along with -n option as shown below.

```
$ echo a b c d e f | xargs -p -n 3
/bin/echo a b c ?...y
/bin/echo d e f ?...a b c
y
d e f
```

In the following output, I said "n" to all the echo output. So, xargs did not execute anything.

```
$ echo a b c d e f | xargs -p -n 3
/bin/echo a b c ?...n
/bin/echo d e f ?...n
/bin/echo ?...n
```

Note: This is helpful when you are combining xargs with commands that are disruptive like rm. In those cases, you may want to see what xargs does.

5. Avoid Default /bin/echo for Blank Input Using -r Option

When there is a blank input (i.e no input was given to xargs command), it will execute a /bin/echo command which will display a new line as shown below.

```
$ xargs -p
```

Press ctrl-d after typing "xargs -p", which will indicate that it executed a /bin/echo as shown below.

```
$ xargs -p  
    /bin/echo ?...y
```

```
$
```

6. Print the Command Along with Output Using -t Option

In the following example, type "abcd" as the input for the xargs -t command.

```
$ xargs -t  
abcd
```

Press ctrl-d to complete the above xargs -t command, which will display the command that xargs really executes before displaying the output. In this case, the command that xargs executes is "/bin/echo abcd", which is displayed here.

```
$ xargs -t  
abcd  
/bin/echo abcd  
abcd
```

7. Combine Xargs with Find Command

It is one of the most important usage of xargs command. When you need to find certain type of files and perform certain actions on them (most popular being the delete action).

The xargs command is very effective when we combine with other commands.

In the following example, we took the output of the find command, and passed it as input to the xargs command. But, instead of executing the default /bin/echo command, we are instructing xargs command to execute the rm -rm command on the input.

So, in this example, the output of the find command is all the files with *.c extension, which is given as input to the xargs command, which in-turn execute "rm -rf" command on all the *.c files.

```
$ ls  
one.c one.h two.c two.h
```

```
$ find . -name "*.c" | xargs rm -rf
```

```
$ ls  
one.h two.h
```

8. Delete Files that has White-space in the Filename

So we see that despite of running the rm command on the .c files in this directory, the file 'The Geek Stuff.c' was not deleted. This is because this file contains white space characters in its name.

As shown in the following example, it deleted all the files with the *.c extension except one. i.e the file that has white space in the filename (i.e "The Geek Stuff.c") was not deleted by the rm command.

```
$ touch "test.c"
```

```
$ ls  
one.c one.h two.c two.h test.c
```

```
$ find . -name "*.c" | xargs rm -rf
```

```
$ ls  
one.h two.h The Geek Stuff.c
```

In this situation, use the -print0 option with find command and -0 option with xargs command to delete files including those that has space in the filenames as shown below.

```
$ ls  
one.c one.h two.c two.h test.c
```

```
$ find . -name "*.c" -print0 | xargs -0 rm -rf
```

```
$ ls  
one.h two.h
```

9. Display System Limits on xargs using --show-limits option

See the example below:

The following example displays all the limits set by the OS that will have an impact on the way how xargs command works.

```
$ xargs --show-limits
```

Your environment variables take up 1203 bytes

POSIX upper limit on argument length (this system): 2093901

POSIX smallest allowable upper limit on argument length (all systems): 4096

Maximum length of command we could actually use: 2092698

Size of command buffer we are actually using: 131072

Execution of xargs will continue now, and it will try to read its input and run commands; if this is not what you wanted to happen, please type the

end-of-file keystroke.

Warning: /bin/echo will be run at least once. If you do not want that to happen, then press the interrupt keystroke

10. Combine Xargs with Grep command

The xargs command can be combined with grep command to filter particular files from the search results of the find command.

In the following example, find command provided all the .c files as input to xargs.

The xargs command executes the grep command to find all the files (among the files provided by find command) that contained a string 'stdlib.h'.

```
$ find . -name '*.c' | xargs grep 'stdlib.h'
./tgsthreads.c:#include
./valgrind.c:#include
./dirent.c:#include
./xv.c:#include
./temp.c:#include
```

Using dig command you can query DNS name servers for your DNS lookup related tasks. This article explains 10 examples on how to use dig command.

1. Simple dig Command Usage (Understand dig Output)

When you pass a domain name to the dig command, by default it displays the A record (the ip-address of the site that is queried) as shown below.

In this example, it displays the A record of redhat.com in the "ANSWER SECTION" of the dig command output.

```
$ dig redhat.com ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> redhat.com ;; global options:
+cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62863 ;; flags: qr
rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 3 ;; QUESTION SECTION: ;redhat.com.
IN      A      ;; ANSWER SECTION: redhat.com.                37      IN      A
209.132.183.81 ;; AUTHORITY SECTION: redhat.com.            73      IN      NS
ns4.redhat.com. redhat.com.                73      IN      NS      ns3.redhat.com.
redhat.com.                73      IN      NS      ns2.redhat.com. redhat.com.                73
IN      NS      ns1.redhat.com. ;; ADDITIONAL SECTION: ns1.redhat.com.                73      IN
A      209.132.186.218 ns2.redhat.com.        73      IN      A      209.132.183.2
ns3.redhat.com.            73      IN      A      209.132.176.100 ;; Query time: 13 msec ;;
```

```
SERVER: 209.144.50.138#53(209.144.50.138) ;; WHEN: Thu Jan 12 10:09:49 2012 ;; MSG SIZE  
rcvd: 164
```

The dig command output has the following sections:

- **Header:** This displays the dig command version number, the global options used by the dig command, and few additional header information.
- **QUESTION SECTION:** This displays the question it asked the DNS. i.e This is your input. Since we said 'dig redhat.com', and the default type dig command uses is A record, it indicates in this section that we asked for the A record of the redhat.com website
- **ANSWER SECTION:** This displays the answer it receives from the DNS. i.e This is your output. This displays the A record of redhat.com
- **AUTHORITY SECTION:** This displays the DNS name server that has the authority to respond to this query. Basically this displays available name servers of redhat.com
- **ADDITIONAL SECTION:** This displays the ip address of the name servers listed in the AUTHORITY SECTION.
- **Stats section** at the bottom displays few dig command statistics including how much time it took to execute this query

2. Display Only the ANSWER SECTION of the Dig command Output

For most part, all you need to look at is the "ANSWER SECTION" of the dig command. So, we can turn off all other sections as shown below.

- **+nocomments** – Turn off the comment lines
- **+noauthority** – Turn off the authority section
- **+noadditional** – Turn off the additional section
- **+nostats** – Turn off the stats section
- **+noanswer** – Turn off the answer section (Of course, you wouldn't want to turn off the answer section)

The following dig command displays only the ANSWER SECTION.

```
$ dig redhat.com +nocomments +noquestion +noauthority +noadditional +nostats ; <<>> DiG  
9.7.3-RedHat-9.7.3-2.el6 <<>> redhat.com +nocomments +noquestion +noauthority  
+noadditional +nostats ;; global options: +cmd redhat.com.          9      IN      A  
209.132.183.81
```

Instead of disabling all the sections that we don't want one by one, we can disable all sections using +noall (this turns off answer section also), and add the +answer which will show only the answer section.

The above command can also be written in a short form as shown below, which displays only the ANSWER SECTION.

```
$ dig redhat.com +noall +answer ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> redhat.com
+noall +answer ;; global options: +cmd redhat.com.          60      IN      A
209.132.183.81
```

3. Query MX Records Using dig -t MX

To query MX records, pass MX as an argument to the dig command as shown below.

```
$ dig redhat.com MX +noall +answer ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> redhat.com
MX +noall +answer ;; global options: +cmd redhat.com.          513      IN      MX
5 mx1.redhat.com. redhat.com.          513      IN      MX      10 mx2.redhat.com.
```

You can also use option -t to pass the query type (for example: MX) as shown below.

```
$ dig -t MX redhat.com +noall +answer ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> -t MX
redhat.com +noall +answer ;; global options: +cmd redhat.com.          489      IN
MX      10 mx2.redhat.com. redhat.com.          489      IN      MX      5
mx1.redhat.com.
```

4. Query NS Records Using dig -t NS

To query the NS record use the type NS as shown below.

```
$ dig redhat.com NS +noall +answer ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> redhat.com NS
+noall +answer ;; global options: +cmd redhat.com.          558      IN      NS
ns2.redhat.com. redhat.com.          558      IN      NS      ns1.redhat.com.
redhat.com.          558      IN      NS      ns3.redhat.com. redhat.com.
558      IN      NS      ns4.redhat.com.
```

You can also use option -t to pass the query type (for example: NS) as shown below.

```
$ dig -t NS redhat.com +noall +answer ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> -t NS
redhat.com +noall +answer ;; global options: +cmd redhat.com.          543      IN
NS      ns4.redhat.com. redhat.com.          543      IN      NS      ns1.redhat.com.
redhat.com.          543      IN      NS      ns3.redhat.com. redhat.com.
543      IN      NS      ns2.redhat.com.
```

5. View ALL DNS Records Types Using dig -t ANY

To view all the record types (A, MX, NS, etc.), use ANY as the record type as shown below.

```
$ dig redhat.com ANY +noall +answer ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> redhat.com
ANY +noall +answer ;; global options: +cmd redhat.com.          430      IN      MX
5 mx1.redhat.com. redhat.com.          430      IN      MX      10 mx2.redhat.com.
redhat.com.          521      IN      NS      ns3.redhat.com. redhat.com.
521      IN      NS      ns1.redhat.com. redhat.com.          521      IN      NS
ns4.redhat.com. redhat.com.          521      IN      NS      ns2.redhat.com.
```

(or) Use -t ANY

```
$ dig -t ANY redhat.com +noall +answer ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> -t ANY
redhat.com +noall +answer ;; global options: +cmd redhat.com.          367      IN
MX      10 mx2.redhat.com. redhat.com.          367      IN      MX      5
mx1.redhat.com. redhat.com.          458      IN      NS      ns4.redhat.com.
redhat.com.          458      IN      NS      ns1.redhat.com. redhat.com.
458      IN      NS      ns2.redhat.com. redhat.com.          458      IN      NS
ns3.redhat.com.
```

6. View Short Output Using dig +short

To view just the ip-address of a web site (i.e the A record), use the short form option as shown below.

```
$ dig redhat.com +short 209.132.183.81
```

You can also specify a record type that you want to view with the +short option.

```
$ dig redhat.com ns +short ns2.redhat.com. ns3.redhat.com. ns1.redhat.com. ns4.redhat.com.
```

7. DNS Reverse Look-up Using dig -x

To perform a DNS reverse look up using the ip-address using dig -x as shown below

For example, if you just have an external ip-address and would like to know the website that belongs to it, do the following.

```
$ dig -x 209.132.183.81 +short www.redhat.com.
```

To view the full details of the DNS reverse look-up, remove the +short option.

```
$ dig -x 209.132.183.81 ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> -x 209.132.183.81 ;;
global options: +cmd ;; Got answer: ;; ->HEADER<- opcode: QUERY, status: NOERROR, id:
62435 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 3 ;; QUESTION
SECTION: ;81.183.132.209.in-addr.arpa. IN PTR ;; ANSWER SECTION:
81.183.132.209.in-addr.arpa. 600 IN PTR www.redhat.com. ;; AUTHORITY SECTION:
183.132.209.in-addr.arpa. 248 IN NS ns2.redhat.com. 183.132.209.in-addr.arpa.
248 IN NS ns1.redhat.com. 183.132.209.in-addr.arpa. 248 IN NS
ns3.redhat.com. 183.132.209.in-addr.arpa. 248 IN NS ns4.redhat.com. ;;
ADDITIONAL SECTION: ns1.redhat.com. 363 IN A 209.132.186.218
ns2.redhat.com. 363 IN A 209.132.183.2 ns3.redhat.com. 363
IN A 209.132.176.100 ;; Query time: 35 msec ;; SERVER:
209.144.50.138#53(209.144.50.138) ;; WHEN: Thu Jan 12 10:15:00 2012 ;; MSG SIZE rcvd: 193
```

8. Use a Specific DNS server Using dig @dnsserver

By default dig uses the DNS servers defined in your /etc/resolv.conf file.

If you like to use a different DNS server to perform the query, specify it in the command line as @dnsserver.

The following example uses ns1.redhat.com as the DNS server to get the answer (instead of using the DNS servers from the /etc/resolv.conf file).

```
$ dig @ns1.redhat.com redhat.com ; <<>> DiG 9.7.3-RedHat-9.7.3-2.el6 <<>> @ns1.redhat.com
redhat.com ; (1 server found) ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<-
opcode: QUERY, status: NOERROR, id: 20963 ;; flags: qr aa rd; QUERY: 1, ANSWER: 1,
AUTHORITY: 4, ADDITIONAL: 4 ;; WARNING: recursion requested but not available ;; QUESTION
SECTION: ;redhat.com.                IN      A      ;; ANSWER SECTION: redhat.com.
60      IN      A      209.132.183.81  ;; AUTHORITY SECTION: redhat.com.        600
IN      NS      ns1.redhat.com. redhat.com.        600      IN      NS
ns4.redhat.com. redhat.com.        600      IN      NS      ns3.redhat.com.
redhat.com.        600      IN      NS      ns2.redhat.com.  ;; ADDITIONAL SECTION:
ns1.redhat.com.        600      IN      A      209.132.186.218 ns2.redhat.com.
600      IN      A      209.132.183.2 ns3.redhat.com.        600      IN      A
209.132.176.100 ns4.redhat.com.        600      IN      A      209.132.188.218  ;; Query
time: 160 msec ;; SERVER: 209.132.186.218#53(209.132.186.218) ;; WHEN: Thu Jan 12 10:22:11
2012 ;; MSG SIZE rcvd: 180
```

9. Bulk DNS Query Using dig -f (and command line)

Query multiple websites using a data file:

You can perform a bulk DNS query based on the data from a file.

First, create a sample names.txt file that contains the website that you want to query.

```
$ vi names.txt redhat.com centos.org
```

Next, execute dig -f as shown below, which will perform DNS query for the websites listed in the names.txt file and display the output.

```
$ dig -f names.txt +noall +answer redhat.com.        60      IN      A
209.132.183.81 centos.org.        60      IN      A      72.232.194.162
```

You can also combine record type with the -f option. The following example displays the MX records of multiple websites that are located in the names.txt file.

```
$ dig -f names.txt MX +noall +answer redhat.com.          600      IN      MX      10
mx2.redhat.com. redhat.com.          600      IN      MX      5 mx1.redhat.com.
centos.org.      3600      IN      MX      10 mail.centos.org.
```

Query multiple websites from dig command line:

You can also query multiple websites from the dig command line as shown below. The following example queries MX record for redhat.com, and NS record for centos.org from the command line

```
$ dig redhat.com mx +noall +answer centos.org ns +noall +answer ; <<>> DiG 9.7.3-RedHat-
9.7.3-2.el6 <<>> redhat.com mx +noall +answer centos.org ns +noall +answer ;; global
options: +cmd redhat.com.          332      IN      MX      10 mx2.redhat.com.
redhat.com.      332      IN      MX      5 mx1.redhat.com. centos.org.
3778      IN      NS      ns3.centos.org. centos.org.          3778      IN      NS
ns4.centos.org. centos.org.          3778      IN      NS      ns1.centos.org.
```

10. Use \$HOME/.digrc File to Store Default dig Options

If you are always trying to view only the ANSWER section of the dig output, you don't have to keep typing "+noall +answer" on your every dig command. Instead, add your dig options to the .digrc file as shown below.

```
$ cat $HOME/.digrc +noall +answer
```

Now anytime you execute dig command, it will always use +noall and +answer options by default. Now the dig command line became very simple and easy to read without you have to type those options every time.

```
$ dig redhat.com redhat.com.          60      IN      A      209.132.183.81 $ dig
redhat.com MX redhat.com.          52      IN      MX      5 mx1.redhat.com.
redhat.com.      52      IN      MX      10 mx2.redhat.com.
```

