

1. Linux Introduction

What is Open Source?

- Open source: software and source code available to all
- The Free Software Foundation specifies four freedoms
 - The freedom to run the program for any purpose.
 - The freedom to study and modify the source code
 - The freedom to redistribute the program
 - The freedom to create derivative programs
- Many open-source licenses exist, each with different particulars

Linux Origins

- 1984: The GNU Project and the Free Software Foundation
 - Creates open source version of UNIX utilities
 - Creates the General Public License (GPL)
 - Software license enforcing open source principles
- 1991: Linus Torvalds
 - Creates open source, UNIX-like kernel, released under the GPL
 - Ports some GNU utilities, solicits assistance online
- Today:
 - Linux kernel + GNU utilities = complete, open source, UNIX-like operating system
 - Packaged for targeted audiences as *distributions*

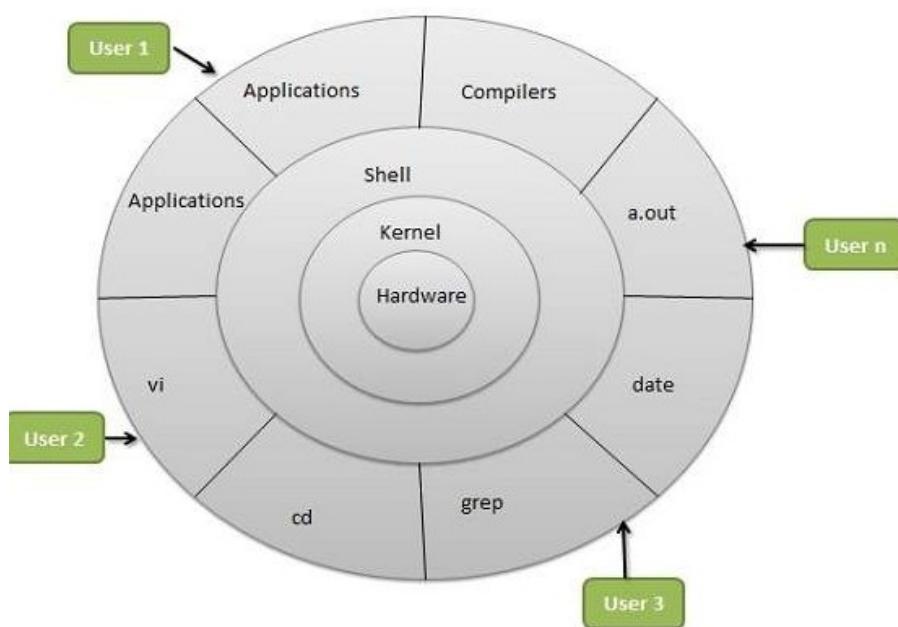
Linux principles

- Everything is a file (including hardware)
- Small, single-purpose programs
- Ability to chain programs together to perform complex tasks
- Avoid captive user interfaces
- Configuration data stored in text

Why Linux?

- OpenSource.
- Community support.
- Heavily customizable.
- Most Servers runs on Linux.
- DevOps most of the tools implements on Linux only.
- Automation
- Secure.

Architecture of Linux



Some Important Directories

- Home Directories: /root, /home/username
- User Executable: /bin, /usr/bin, /usr/local/bin
- System Executables: /sbin, /usr/sbin, /usr/local/sbin
- Other Mountpoints: /media, /mnt
- Configuration: /etc
- Temporary Files: /tmp
- Kernels and Bootloader: /boot
- Server Data: /var, /srv
- System Information: /proc, /sys
- Shared Libraries: /lib, /usr/lib, /usr/local/lib

Different Linux distros

→ Popular Desktop Linux OS

- Ubuntu Linux
- Linux Mint
- Arch Linux
- Fedora
- Debian
- OpenSuse

→ Popular Server Linux OS

- Red Hat Enterprise Linux
- Ubuntu Server
- Centos
- SUSE Enterprise Linux

Most used Linux distros currently in IT industry.

RPM based:- RHEL & Centos

Debian based :- Ubuntu Server

Difference between RPM based and Debian based.

From user's point of view, there isn't much difference in these tools. The RPM and DEB formats are both just archive files, with some metadata attached to them. They are both equally arcane, have hardcoded install paths and only differ in subtle details. DEB files are installation files for Debian based distributions. RPM files are installation files for Red Hat based distributions. Ubuntu is based on Debian's package manager based on APT and DPKG. Red Hat, CentOS and Fedora are based on the old Red Hat Linux package management system, RPM.

DEB or .deb (Debian based softwares)

DEB is the extension of the Debian software package format and the most often used name for such binary packages. DEB was developed by Bedian.

Example: Google chrome software

Package name: google-chrome-stable_current_amd64.deb

Installation: dpkg -i google-chrome-stable_current_amd64.deb

RPM or .rpm (Red Hat based softwares.)

It is a package management system. The name RPM variously refers to the .rpm file format, files in this format, software packaged in such files, and the package manager itself. RPM was intended primarily for Linux distributions; the file format is the baseline package format of the Linux Standard Base. RPM was developed by Community & Red Hat.

Example: Google chrome software

Package Name: google-chrome-stable-57.0.2987.133-1.x86_64.rpm

Installation: rpm -ivh google-chrome-stable-57.0.2987.133-1.x86_64.rpm

NOTE: You will also encounter different commands, packages and service names while using both kinds of distros.

2. Basic Commands

→ Open Terminal

→ Know where you are? Present Working Directory

```
imran@DevOps: ~
File Edit View Search Terminal Help
imran@DevOps:~$ pwd
/home/imran
imran@DevOps:~$ █
```

→ Create a directory/folder in your home directory.

```
imran@DevOps:~$ mkdir linux-practices
imran@DevOps:~$ █
```

→ Change your current working directory to linux-practices(Go to linux-practices folder).

```
imran@DevOps:~$ cd linux-practices/
imran@DevOps:~/linux-practices$ █
```

→ Create some more directories and list them with “ls” command.

```
imran@DevOps:~/linux-practices$ mkdir vmdir
imran@DevOps:~/linux-practices$ mkdir testdir
imran@DevOps:~/linux-practices$ mkdir devopsdir
imran@DevOps:~/linux-practices$ ls
devopsdir testdir vmdir
```

→ Create some empty files with “touch” command and list them.

```
imran@DevOps:~/linux-practices$ touch file2 file3 file4
imran@DevOps:~/linux-practices$ ls
devopsdir file1 file2 file3 file4 testdir vmdir
```

→ **Reconfirm your location in your system.**

```
imran@DevOps:~/linux-practices$ pwd  
/home/imran/linux-practices  
imran@DevOps:~/linux-practices$ ls  
devopsdir file1 file2 file3 file4 testdir vkdir
```

Absolute path and Relative path

What is a path?

A path is a unique location to a file or a folder in a file system of an OS. A path to a file is a combination of / and alpha-numeric characters.

What is an absolute path?

An absolute path is defined as specifying the location of a file or directory from the root directory(/). In other words we can say absolute path is a complete path from start of actual filesystem from / directory.

Some examples of absolute path:

/home/imran/linux-practices/

/var/ftp/pub

/etc/samba.smb.conf

/boot/grub/grub.conf

If you see all these paths started from / directory which is a root directory for every Linux/Unix machines.

What is the relative path?

Relative path is defined as path related to the present working directory(pwd). Suppose I am located in /home/imran and I want to change directory to /home/imran/linux-practices. I can use relative path concept to change directory to linux-practices and also devopsdir directory.

```
File Edit View Search Terminal Help  
imran@DevOps:~$ pwd  
/home/imran  
imran@DevOps:~$ cd linux-practices/  
imran@DevOps:~/linux-practices$ ls  
devopsdir file1 file2 file3 file4 testdir vkdir  
imran@DevOps:~/linux-practices$ pwd  
/home/imran/linux-practices  
imran@DevOps:~/linux-practices$ cd devopsdir/  
imran@DevOps:~/.../devopsdir$ pwd  
/home/imran/linux-practices/devopsdir  
imran@DevOps:~/.../devopsdir$ █
```

If you see all these paths did not start with / directory.

→ **Creating directories in devopsdir directory with absolute and relative path.**

```
File Edit View Search Terminal Help
imran@DevOps:~/linux-practices$ ls
devopsdir file1 file2 file3 file4 testdir vmdir
imran@DevOps:~/linux-practices$ mkdir devopsdir/ansible
imran@DevOps:~/linux-practices$ mkdir /home/imran/linux-practices/devopsdir/aws
imran@DevOps:~/linux-practices$ ls devopsdir/
ansible aws
imran@DevOps:~/linux-practices$
```

→ **Copying files into directory.**

```
File Edit View Search Terminal Help
imran@DevOps:~/linux-practices$ pwd
/home/imran/linux-practices
imran@DevOps:~/linux-practices$ ls
devopsdir file1 file2 file3 file4 testdir vmdir
imran@DevOps:~/linux-practices$ cp file1 testdir/
imran@DevOps:~/....testdir$ cd testdir/
file1
imran@DevOps:~/....testdir$
```

→ **Copying directories from one location to another.**

```
File Edit View Search Terminal Help
imran@DevOps:~/linux-practices$ cd
imran@DevOps:~$ cd -
/home/imran/linux-practices
imran@DevOps:~/linux-practices$ pwd
/home/imran/linux-practices
imran@DevOps:~/linux-practices$ ls
devopsdir file1 file2 file3 file4 testdir vmdir
imran@DevOps:~/linux-practices$ cp -rvfp testdir/ vmdir/
'testdir/' -> 'vmdir/testdir'
'testdir/file1' -> 'vmdir/testdir/file1'
imran@DevOps:~/linux-practices$ ls vmdir/
testdir
imran@DevOps:~/linux-practices$
```

→ Moving files from one location to another.

```
imran@DevOps:~/linux-practices$ pwd
/home/imran/linux-practices
imran@DevOps:~/linux-practices$ ls
devopsdir file1 file2 file3 file4 testdir vmdir
imran@DevOps:~/linux-practices$ mv devopsdir/ vmdir/
imran@DevOps:~/linux-practices$ ls
file1 file2 file3 file4 testdir vmdir
imran@DevOps:~/linux-practices$ ls vmdir/
devopsdir testdir
imran@DevOps:~/linux-practices$ mv file3 file4 vmdir/
imran@DevOps:~/linux-practices$ ls
file1 file2 testdir vmdir
```

→ Removing files and directories.

```
imran@DevOps:~/linux-practices$ rm file1
imran@DevOps:~/linux-practices$ ls
file2 testdir vmdir
imran@DevOps:~/linux-practices$ rm -rf testdir/
imran@DevOps:~/linux-practices$ ls
file2 vmdir
```

3. VIM EDITOR

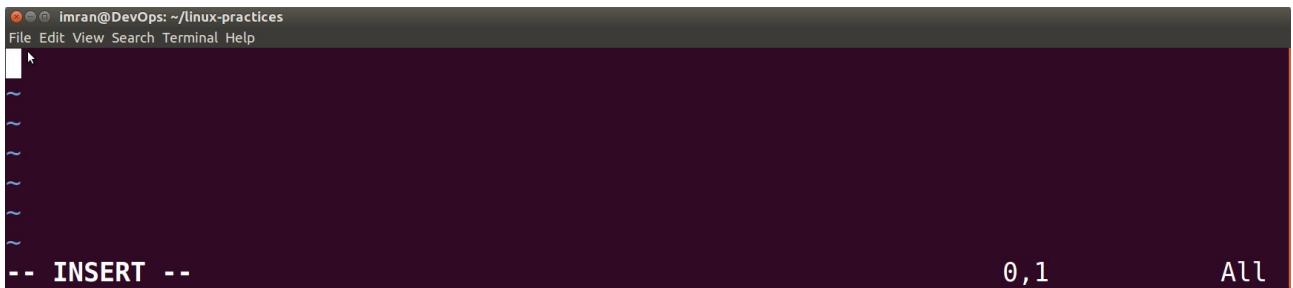
→ Install vim editor.

```
imran@DevOps:~/linux-practices$ sudo apt-get install vim
[sudo] password for imran:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

→ Open up a file in vim editor

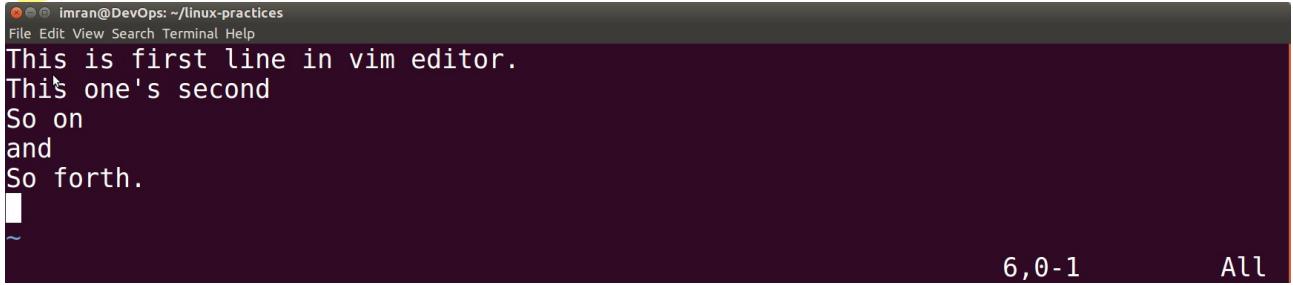
```
imran@DevOps:~/linux-practices$ vim firstfile.txt
```

→ Hit i to enter into insert mode



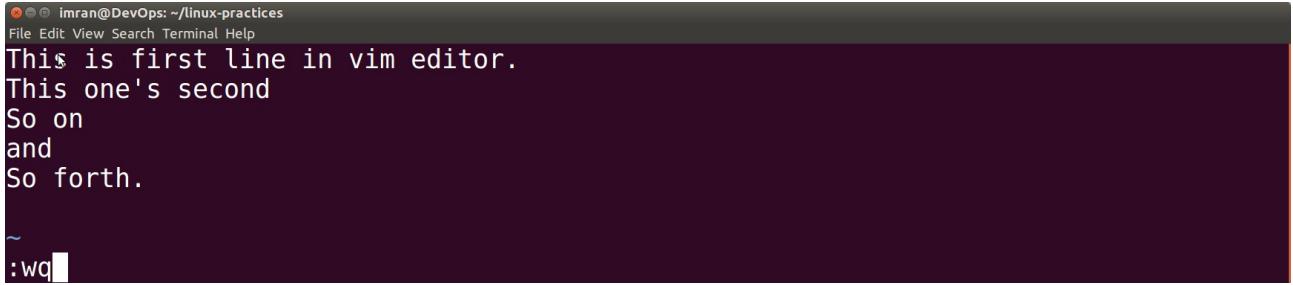
```
imran@DevOps: ~/linux-practices
File Edit View Search Terminal Help
~
~
~
~
~
-- INSERT --          0,1          All
```

=> type few lines => hit Esc



```
imran@DevOps: ~/linux-practices
File Edit View Search Terminal Help
This is first line in vim editor.
This one's second
So on
and
So forth.
~          6,0-1          All
```

=> type :wq



```
imran@DevOps: ~/linux-practices
File Edit View Search Terminal Help
This is first line in vim editor.
This one's second
So on
and
So forth.
~          6,0-1          All
:wq
```

=> Enter.

→ Read file with cat command.

```

imran@DevOps:~/linux-practices$ cat firstfile.txt
This is first line in vim editor.
This one's second
So on
and
So forth.

imran@DevOps:~/linux-practices$ █

```

VIM EDITOR

VI Visual display editor

VIM Visual display editor improved

This is command mode editor for files. Other editors in Linux are emacs, gedit
 vi editor is most popular

It has 3 modes:

- 1 Command Mode
- 2 Insert mode (edit mode)
- 3 extended command mode

Note: When you open the vim editor, it will be in the command mode by default.

Command Mode:

gg	To go to the beginning of the page
G	To go to end of the page
w	To move the cursor forward, word by word
b	To move the cursor backward, word by word
nw	To move the cursor forward to n words (5W)
nb	To move the cursor backward to n words (5B)
u	To undo last change (word)

U	To undo the previous changes (entire line)
Ctrl+R	To redo the changes
YY	To copy a line
nyy	To copy n lines (5yy or 4yy)
p	To paste line below the cursor position
P	To paste line above the cursor position
dw	To delete the word letter by letter (like Backspace)
x	To delete the world letter by letter (like DEL Key)
dd	To delete entire line
ndd	To delete n no. of lines from cursor position(5dd)
/	To search a word in the file

Extended Mode: (Colon Mode)

Extended Mode is used for save and quit or save without quit using "Esc" Key with ":"

Esc+(:w	To Save the changes
Esc+(:q	To quit (Without saving)
Esc+(:wq	To save and quit
Esc+(:w!	To save forcefully
Esc+(:wq!	To save and quit forcefully
Esc+(:x	To save and quit
Esc+(:X	To give password to the file and remove password
Esc+(:20(n)	To go to line no 20 or n
Esc+(:se nu	To set the line numbers to the file
Esc+(:se nonu	To Remove the set line numbers

ls syntax

```
$ ls [options] [file|dir]
```

ls command options

ls command main options:

option	description
<code>ls -a</code>	list all files including hidden file starting with ''
<code>ls --color</code>	colored list [=always/never/auto]
<code>ls -d</code>	list directories - with ' */'
<code>ls -F</code>	add one char of */=>@ to enteries
<code>ls -i</code>	list file's inode index number
<code>ls -l</code>	list with long format - show permissions
<code>ls -la</code>	list long format including hidden files
<code>ls -lh</code>	list long format with readable file size
<code>ls -ls</code>	list with long format with file size
<code>ls -r</code>	list in reverse order
<code>ls -R</code>	list recursively directory tree
<code>ls -s</code>	list file size
<code>ls -S</code>	sort by file size
<code>ls -t</code>	sort by time & date
<code>ls -X</code>	sort by extension name

Types of files in linux.

File Type	First Character in File Listing	Description
Regular file	-	Normal files such as text, data, or executable files
Directory	d	Files that are lists of other files
Link	l	A shortcut that points to the location of the actual file
Special file	c	Mechanism used for input and output, such as files in /dev
Socket	s	A special file that provides inter-process networking protected by the file system's access control
Pipe	p	A special file that allows processes to communicate with each other without using network socket semantics

Symbolic links

Symbolic links are like desktop shortcuts we use in windows.

Create a soft link for /var/log directory in our current working directory.

```
imran@DevOps:~/linux-practices$ ls
file2 firstfile.txt vmdir
imran@DevOps:~/linux-practices$ ls /var/log/
alternatives.log auth.log.1 cups fontconfig.log kern.log.1 prime-supported.log vbox-install.log Xorg.1.log.old
alternatives.log.1 boot.log dist-upgrade fsck lastlog speech-dispatcher wtmp Xorg.2.log
apport.log boot-sav dmesg gpu-manager.log lightdm syslog wtmp.1
apport.log.1 bootstrap.log dpkg.log installer old-logs syslog.1 Xorg.0.log
apt btmp dpkg.log.1 jenkins php7.0-fpm.log unattended-upgrades Xorg.0.log.old
auth.log btmp.1 faillog kern.log php7.0-fpm.log.1 upstart Xorg.1.log
imran@DevOps:~/linux-practices$ ln -s /var/log/ logdir
imran@DevOps:~/linux-practices$ ls -l
total 8
-rw-rw-r-- 1 imran imran 0 Apr 2 18:00 file2
-rw-rw-r-- 1 imran imran 73 Apr 2 18:29 firstfile.txt
lrwxrwxrwx 1 imran imran 9 Apr 2 18:41 logdir -> /var/log/
drwxrwxr-x 4 imran imran 4096 Apr 2 18:21 vmdir
imran@DevOps:~/linux-practices$ ls logdir
alternatives.log auth.log.1 cups fontconfig.log kern.log.1 prime-supported.log vbox-install.log Xorg.1.log.old
alternatives.log.1 boot.log dist-upgrade fsck lastlog speech-dispatcher wtmp Xorg.2.log
apport.log boot-sav dmesg gpu-manager.log lightdm syslog wtmp.1
apport.log.1 bootstrap.log dpkg.log installer old-logs syslog.1 Xorg.0.log
apt btmp dpkg.log.1 jenkins php7.0-fpm.log unattended-upgrades Xorg.0.log.old
auth.log btmp.1 faillog kern.log php7.0-fpm.log.1 upstart Xorg.1.log
imran@DevOps:~/linux-practices$
```

4. Filter & IO redirection command.

Grep

grep command is used to find texts from any text input.

Passwd file: stores information about all the users in the system

```
imran@DevOps:~/linux-practices$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
imran:x:1000:1000:imran:/home/imran:/bin/bash
```

→ Finding line which contains word as “root” from /etc/passwd file.

```
imran@DevOps:~/linux-practices$ grep root /etc/passwd
root:x:0:0:root:/root:/bin/bash
imran@DevOps:~/linux-practices$
```

→ Linux is case sensitive, Root is different than root. Ignoring case in grep with -i option.

```
imran@DevOps:~/linux-practices$ grep Root /etc/passwd
imran@DevOps:~/linux-practices$ grep -i Root /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

→ To display things except the given word use -v option

Filter Commands

- less: Displays file content page wise or line wise.

Ex: less /etc/passwd

Note: -press **Enter** key to scroll down line by line (or)

Use **d** to go to next page

Use **b** to go to previous page

Use **/** to search for a word in the file

Use **v** to go vi mode where you can edit the file and once you save it you will back to less command

• more

more is exactly same like **less**

Ex: #more /etc/passwd

Note: -press **Enter** key to scroll down line by line (or)

Use **d** to go to next page

Use **/** to search for a word in the file

Use **v** to go vi mode where you can edit the file and once you save it you will back to more command

• head

It is used to display the **top 10 lines** of the file.

Ex:# head /etc/passwd

```
[root@ktlinux ~]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
```

• tail

It is used to display the **last 10 lines** of the file

#tail /etc/passwd

```
[root@ktlinux ~]# tail /etc/passwd
apache:x:48:48:Apache:/var/www:/sbin/nologin
nslcd:x:65:55:LDAP Client User:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
pulse:x:496:494:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tcpdump:x:72:72::/sbin/nologin
visitor:x:500:500:visitor:/home/visitor:/bin/bash
ktuser:x:501:501::/home/ktuser:/bin/bash
```

- **cut**

```
# cut -d -f filename (where d stands for delimiter ex. :, " etc and f stands for field)
```

```
[root@ktlinux ~]# cut -d: -f1 /etc/passwd
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
uucp
```

To delimit spaces and print the field

```
#cut -d " " -f1 filename
```

- **sed**

sed stands for **stream editor**, which is used to search a word in the file and replace it with the word required to be in the output

Note: it will only modify the output, but there will be no change in the original file.

```
#sed 's/searchfor/replacewith/g' filename
```

```
[root@ktlinux ~]# cat ktfile
Welcome to Kernel Tech
[root@ktlinux ~]# sed 's/Tech/Technologies/g' ktfile
Welcome to Kernel Technologies
[root@ktlinux ~]# cat ktfile
Welcome to Kernel _Tech
```

I/O redirection

Redirection is a process where we can copy the output of any command(s), file(s) into a new file. There are two ways of redirecting the output into a file.

Using **>** or **>> filename** after the command, and

→ Create a file named devopstools with below content.

```
imran@DevOps:~/linux-practices$ cat devopstools
chef tech
ansible tech
git tech
docker tech
aws tech
```

→ Search for text “tech” replace it with “tools” and redirect output to a new file.

```
imran@DevOps:~/linux-practices$ sed 's/tech/tools/g' devopstools
chef tools
ansible tools
git tools
docker tools
aws tools
imran@DevOps:~/linux-practices$ sed 's/tech/tools/g' devopstools > newtools.txt
imran@DevOps:~/linux-practices$ cat newtools.txt
chef tools
ansible tools
git tools
docker tools
aws tools
```

Note: if the given name of the file is not available a new file will be created automatically. If the file already exists then it will overwrite contents of that file.

→ Appending another output in same file with “>>” .

```
imran@DevOps:~/linux-practices$ tail /etc/passwd >> newtools.txt
imran@DevOps:~/linux-practices$ cat newtools.txt
chef tools
ansible tools
git tools
docker tools
aws tools
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
imran:x:1000:1000:Imran,,,:/home/imran:/bin/bash
jenkins:x:121:131:Jenkins,,,:/var/lib/jenkins:/bin/bash
guest-lx1wni:x:999:999:Guest:/tmp/guest-lx1wni:/bin/bash
guests:x:1001:1001,,,:/home/guests:/bin/bash
nvidia-persistenced:x:122:132:NVIDIA Persistence Daemon,,,:/sbin/nologin
guest-yjzlgk:x:998:998:Guest:/tmp/guest-yjzlgk:/bin/bash
imran@DevOps:~/linux-practices$
```

Piping

So far we've dealt with sending data to and from files. Now we'll take a look at a mechanism for sending data from one program to another. It's called piping and the operator we use is (|). What this operator does is feed the output from the program on the left as input to the program on the right.

```
imran@DevOps:~$ cd linux-practices/
imran@DevOps:~/linux-practices$ ls
chefdk_1.2.22-1_amd64.deb  file2      logdir      tree_1.7.0-3_amd64.deb
devopstools                 firstfile.txt newtools.txt  vpdir
imran@DevOps:~/linux-practices$ ls | head -3
chefdk_1.2.22-1_amd64.deb
devopstools
file2
imran@DevOps:~/linux-practices$ ls | grep logdir
logdir
imran@DevOps:~/linux-practices$ cat /etc/passwd | grep root
root:x:0:0:root:/root:/bin/bash
imran@DevOps:~/linux-practices$
```

Find

find command is used to find the files or directory's path, it is exactly like the find option in windows where you can search for a file.

```
imran@DevOps:~/linux-practices$ find /home/imran/ -name newtools.txt
/home/imran/linux-practices/newtools.txt
```

Options that can be used with find command:

Option	Usage
-name	For searching a file with its name
-inum	For searching a file with particular inode number
-type	For searching a particular type of file
-user	For files whose owner is a particular user
-group	For files belonging to particular group

5. Users & Groups.

USERS

Some Important Points related to Users:

- Users and groups are used to control access to files and resources
 - Users login to the system by supplying their username and password
 - Every file on the system is owned by a user and associated with a group
 - Every process has an owner and group affiliation, and can only access the resources its owner or group can access.
-
- Every user of the system is assigned a unique user ID number (the UID)
 - Users name and UID are stored in **/etc/passwd**
 - User's password is stored in **/etc/shadow** in encrypted form.
 - Users are assigned a **home directory** and a program that is run when they login (**Usually a shell**)
 - Users cannot read, write or execute each other's files without permission.

Types of users In Linux and their attributes:

TYPE	EXAMPLE	USER ID (UID)	GROUP ID (GID)	HOME DIRECTORY	SHELL
Super User	Root	0	0	/root	/bin/bash
System User	ftp, ssh, apache nobody	1 to 499	1 to 499	/var/ftp , etc	/sbin/nologin
Normal User	Visitor, ktuser,etc	500 to 60000	500 to 60000	/home/user name	/bin/bash

In Linux there are three types of users.

1. Super user or root user

Super user or the root user is the most powerful user. He is the administrator user.

2. System user

System users are the users created by the softwares or applications. For example if we install Apache it will create a user apache. These kinds of users are known as system users.

3. Normal user

Normal users are the users created by root user. They are normal users like Rahul, Musab etc. Only the root user has the permission to create or remove a user.

Whenever a user is created in Linux things created by default:-

- A home directory is created(/home/username)
- A mail box is created(/var/spool/mail)
- unique UID & GID are given to user

Passwd file

1. /etc/passwd

```
[root@ktlinux ~]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

The above fields are

- **root** =name
- **x**= link to password file i.e. /etc/shadow
- **0 or 1**= UID (user id)
- **0 or 1**=GID (group id)
- **root or bin** = comment (brief information about the user)
- **/root or /bin** = home directory of the user
- **/bin/bash or /sbin/nologin** = shell

\$ id with login and group	=> Show the active user id
\$ last system (few more examples)	=> Show last logins on the
\$ who system(real user who logged in)	=> Show who is logged on the
\$ sudo groupadd admin (force add existing group)	=> Add group "admin"

```
$ sudo adduser sam => Create user "sam" and add to group  
"admin"(here read all parameter)
```

```
$ sudo usermod => Modify user  
information(mostly useful for linux system admins)
```

```
imran@DevOps:~/linux-practices$ sudo groupadd deck  
imran@DevOps:~/linux-practices$ sudo usermod -G deck sam  
imran@DevOps:~/linux-practices$ id sam  
uid=1002(sam) gid=1002(sam) groups=1002(sam),1007(deck)
```

```
$ sudo passwd sam => Changing password for other user
```

```
$ passwd => Changing your own password.
```

```
$ sudo userdel -r sam => Delete user sam  
(force,file removal)
```

```
$ sudo groupdel deck => Deletes a group
```

6.File permissions

Viewing Permissions from the Command-Line

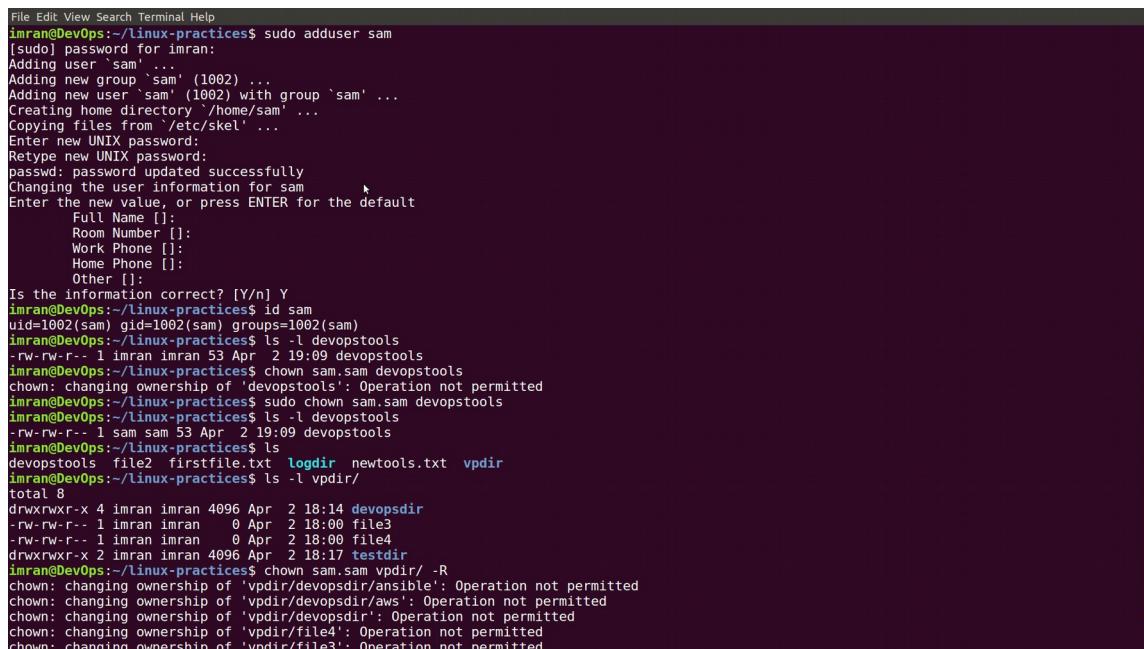
- File permissions may be viewed using **ls -l**

```
$ ls -l /bin/login
-rwxr-xr-x 1 root root 19080 Apr 1 18:26 /bin/login
```

- Four symbols are used when displaying permissions:
 - r: permission to read a file or list a directory's contents
 - w: permission to write to a file or create and remove files from a directory
 - x: permission to execute a program or change into a directory and do a long listing of the directory
 - : no permission (in place of the r, w, or x)

Changing File Ownership

- Only root can change a file's owner
- Only root or the owner can change a file's group
- Ownership is changed with **chown**:
 - chown [-R] user_name file|directory ...**
- Group-Ownership is changed with **chgrp**:
 - chgrp [-R] group_name file|directory ...**



A terminal window showing the creation of a new user 'sam' and subsequent ownership changes. The session starts with adding a user:

```
imran@DevOps:~/linux-practices$ sudo adduser sam
[sudo] password for imran:
Adding user 'sam' ...
Adding new group 'sam' (1002) ...
Adding new user 'sam' (1002) with group 'sam' ...
Creating home directory '/home/sam' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for sam
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] Y
```

Then, it shows attempting to change ownership of a directory while logged in as 'imran', which fails due to lack of root权限:

```
imran@DevOps:~/linux-practices$ id sam
uid=1002(sam) gid=1002(sam) groups=1002(sam)
imran@DevOps:~/linux-practices$ ls -l devopstools
-rw-rw-r-- 1 imran imran 53 Apr 2 19:09 devopstools
imran@DevOps:~/linux-practices$ chown sam.sam devopstools
chown: changing ownership of 'devopstools': Operation not permitted
imran@DevOps:~/linux-practices$ sudo chown sam.sam devopstools
imran@DevOps:~/linux-practices$ ls -l devopstools
-rw-rw-r-- 1 sam sam 53 Apr 2 19:09 devopstools
imran@DevOps:~/linux-practices$ ls
devopstools file2 firstfile.txt logdir newtools.txt vmdir
imran@DevOps:~/linux-practices$ ls -l vmdir/
total 8
drwxrwxr-x 4 imran imran 4096 Apr 2 18:14 devopsdir
-rw-rw-r-- 1 imran imran 0 Apr 2 18:00 file3
-rw-rw-r-- 1 imran imran 0 Apr 2 18:00 file4
drwxrwxr-x 2 imran imran 4096 Apr 2 18:17 testdir
imran@DevOps:~/linux-practices$ chown sam.sam vmdir/ -R
chown: changing ownership of 'vmdir/devopsdir/ansible': Operation not permitted
chown: changing ownership of 'vmdir/devopsdir/aws': Operation not permitted
chown: changing ownership of 'vmdir/devopsdir': Operation not permitted
chown: changing ownership of 'vmdir/file4': Operation not permitted
chown: changing ownership of 'vmdir/file3': Operation not permitted
```

Changing Permissions - Symbolic Method

- To change access modes:
 - **chmod [-OPTION]... mode[,mode] file| directory ...**
- *mode* includes:
 - **u, g or o** for user, group and other
 - **+ - or =** for grant, deny or set
 - **r, w or x** for read, write and execute
- Options include:
 - **-R** Recursive
 - **-v** Verbose
 - **--reference** Reference another file for its mode
- Examples:
 - **chmod ugo+r file**: Grant read access to all for *file*
 - **chmod o-wx dir**: Deny write and execute to others for *dir*

Changing Permissions - Numeric Method

- Uses a three-digit mode number
 - first digit specifies owner's permissions
 - second digit specifies group permissions
 - third digit represents others' permissions
- Permissions are calculated by adding:
 - 4 (for read)
 - 2 (for write)
 - 1 (for execute)
- Example:
 - **chmod 640 myfile**

```
imran@DevOps:~/linux-practices$ ls -l
total 16
-rw-rw-r-- 1 sam    sam      53 Apr  2 19:09 devpstools
-rw-rw-r-- 1 imran  imran     0 Apr  2 18:00 file2
-rw-rw-r-- 1 imran  imran    73 Apr  2 18:29 firstfile.txt
lrwxrwxrwx 1 imran  imran    9 Apr  2 18:41 logdir -> /var/log/
-rw-rw-r-- 1 imran  imran   612 Apr  2 19:14 newtools.txt
drwxrwxr-x 4 sam    sam    4096 Apr  2 18:21 vmdir
imran@DevOps:~/linux-practices$ chmod u+x newtools.txt
imran@DevOps:~/linux-practices$ ls -l newtools.txt
-rwxrwr--- 1 imran  imran  612 Apr  2 19:14 newtools.txt
imran@DevOps:~/linux-practices$ chmod o-r newtools.txt
imran@DevOps:~/linux-practices$ ls -l newtools.txt
-rwxrw---- 1 imran  imran  612 Apr  2 19:14 newtools.txt
imran@DevOps:~/linux-practices$ chmod 700 newtools.txt
imran@DevOps:~/linux-practices$ ls -l newtools.txt
-rwx----- 1 imran  imran  612 Apr  2 19:14 newtools.txt
imran@DevOps:~/linux-practices$ chmod 755 newtools.txt
imran@DevOps:~/linux-practices$ ls -l newtools.txt
-rwxr-xr-x 1 imran  imran  612 Apr  2 19:14 newtools.txt
imran@DevOps:~/linux-practices$
```

7. Sudo

sudo gives power to a normal user to execute commands which is owned by root user.

Example shown below:

If a user has already full sudoers privilege, it can become a root user anytime.

→ sudo -i changes from normal user to root user

```
imran@DevOps:~/linux-practices$ id
uid=1000(imran) gid=1000(imran) groups=1000(imran),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),
,128(sambashare),130(docker),1006(dev_dock_gr)
imran@DevOps:~/linux-practices$ sudo -i
[sudo] password for imran:
root@DevOps:~# id
uid=0(root) gid=0(root) groups=0(root)
root@DevOps:~# █
```

Note: User imran was already a sudo user with full privilege.

→ Adding user sam in sudoers list.

```
imran@DevOps:~/linux-practices$ sudo -i
root@DevOps:~# export EDITOR=vim
root@DevOps:~# visudo
```

```
File Edit View Search Terminal Help
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
sam    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
%dev_dock      ALL=(ALL:ALL) ALL
```

21,3 Top

→ Like a user a group can also be added into sudoers list.

```
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL
```

- Every time you enter sudo command it asks your own password. To turn that off use NOPASSWD in sudoers file.

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
sam     ALL=(ALL:ALL) NOPASSWD: ALL
```

- Changing to any other user with “su -” command.

```
imran@DevOps:~/linux-practices$ su - sam
Password:
sam@DevOps:~$ █
```

- Become a root user from sam user login.

```
sam@DevOps:~$ sudo -i
root@DevOps:~# █
```

8. Software Management.

→ Download package from internet.

```
# wget http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree\_1.7.0-3\_amd64.deb
```

```
imran@DevOps:~/linux-practices$ wget http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree_1.7.0-3_amd64.deb
--2017-04-02 20:19:42--  http://archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree_1.7.0-3_amd64.deb
Resolving archive.ubuntu.com (archive.ubuntu.com)... 91.189.88.152, 2001:67c:1560:8001::11, 2001:67c:1360:80
01::17, ...
Connecting to archive.ubuntu.com (archive.ubuntu.com)|91.189.88.152|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 40572 (40K) [application/x-debian-package]
Saving to: 'tree_1.7.0-3_amd64.deb'

tree_1.7.0-3_amd64.deb      100%[=====]  39.62K   153KB/s    in 0.3s

2017-04-02 20:19:43 (153 KB/s) - 'tree_1.7.0-3_amd64.deb' saved [40572/40572]
```

REDHAT family(single package)

```
$ sudo rpm -i pkgname.rpm                      # Install rpm
based package (Installing, Uninstalling, Updating, Querying
,Verifying)

$ sudo rpm -e pkgname                          # Remove package
```

REDHAT family(Package with dependency from internet)

```
$ sudo yum install package-name                 #Install with dep

$ sudo yum remove package-name                  # Remove package
```

UBUNTU/DEBIAN

```
$ sudo dpkg -i package-name                   #Install deb pack

$ sudo dpkg -e package-name                   #Remove deb pack
```

UBUNTU/Debian(Package with dependency from internet)

```
$ sudo apt-get install package-name           #Install with dep

$ sudo apt-get remove package-name             #Remove deb pack
```

Install from source

```
./configure
make
make install (what it is)
```

9. SEARCH

```
$ grep pattern files          # Search for pattern in files  
(you will this command often)  
  
$ grep -r pattern dir        # Search recursively for  
pattern in dir  
  
$ locate file                # Find all instances of file  
  
$ find /home/tom -name 'index*'    # Find files names that start  
with "index"(10 find examples)  
  
$ find /home -size +10000k      # Find files larger than  
10000k in /home
```

10. LOGIN (SSH AND TELNET)

```
$ ssh user@host          # Connect to host as user  
(secure data communication command)  
  
$ ssh -p port user@host    # Connect to host using  
specific port  
  
$ telnet host              # Connect to the system  
using telnet port
```

11. FILE TRANSFER

```
scp  
  
$ scp file.txt server2:/tmp          # Secure copy  
file.txt to remote host /tmp folder  
  
$ scp nixsavy@server2:/www/*.html   /www/tmp    # Copy *.html files  
from remote host to current system /www/tmp folder  
  
$ scp -r nixsavy@server2:/www     /www/tmp      # Copy all files  
and folders recursively from remote server to the current system  
/www/tmp folder  
  
rsync  
  
$ rsync -a /home/apps /backup/          # Synchronize  
source to destination  
  
$ rsync -avz /home/apps linoxide@192.168.10.1:/backup    #  
Synchronize files/directories between the local and remote system  
with compression enabled
```

12. DISK USAGE

```
$ df -h          # Show free space on mounted  
filesystems(commonly used command)  
  
$ df -i          # Show free inodes on mounted  
filesystems  
  
$ fdisk -l       # Show disks partitions sizes and  
types(fdisk command output)  
  
$ du -ah         # Display disk usage in human  
readable form (command variations)  
  
$ du -sh         # Display total disk usage on the  
current directory  
  
$ findmnt        # Displays target mount point for  
all filesystem (refer type,list,evaluate output)  
  
$ mount device-path mount-point # Mount a device
```

13. DIRECTORY TRAVERSE

```
$ cd ..          # To go up one level of the  
directory tree(simple & most needed)  
  
$ cd             # Go to $HOME directory  
  
$ cd /test       # Change to /test directory
```

14. SERVICES

```
$ sudo service apache2 start      # Starts apache2 on ubuntu  
  
$ sudo service httpd start        # Starts apache2 on Redhat  
  
$ sudo service httpd stop  
  
$ sudo service httpd restart      # Restart services  
  
$ sudo service httpd reload       # Reload conf  
  
$ chkconfig httpd on              # starts httpd at boot time  
  
$ chkconfig httpd off             # stops httpd at boot time
```

15. COMPRESSION / ARCHIVES

```
$ tar cf home.tar    home          # Create tar named home.tar  
containing home/ (11 tar examples)  
  
$ tar xf file.tar           # Extract the files from  
file.tar  
  
$ tar czf  file.tar.gz   files      # Create a tar with gzip  
compression  
  
$ gzip file               # Compress file and renames it  
to file.gz (untar gzip file)
```

16. PROCESS RELATED

17. SYSTEM

```
$ uname -a                      => Display linux system  
information  
  
$ uname -r                      => Display kernel release  
information (refer uname command in detail)  
  
$ cat /etc/redhat_release       => Show which version of redhat  
installed  
  
$ uptime                         => Show how long system running +  
load (learn uptime command)  
  
$ hostname                       => Show system host name  
  
$ hostname -i                    => Display the IP address of the  
host (all options hostname)  
  
$ last reboot                   => Show system reboot history  
(more examples last command)  
  
$ date                           => Show the current date and time  
(options of date command)  
  
$ cal                            => Show this month calendar (what  
more in cal)  
  
$ w                             => Display who is online (learn  
more about w command)  
  
$ whoami                         => Who you are logged in as  
(example + screenshots)  
  
$ finger user                   => Display information about user  
(many options of finger command)
```

18. HARDWARE

```
$ dmesg                          => Detected hardware and boot  
messages (dmesg many more options)  
  
$ cat /proc/cpuinfo               => CPU model  
  
$ cat /proc/meminfo              => Hardware memory  
  
$ cat /proc/interrupts          => Lists the number of interrupts  
per CPU per I/O device  
  
$ lshw                           => Displays information on  
hardware configuration of the system  
  
$ lsblk                          => Displays block device related  
information in Linux (sudo yum install util-linux-ng)  
  
$ free -m                         => Used and free memory (-m for  
MB) (free command in detail)
```

```

$ lspci -tv                               => Show PCI devices (very useful
to find vendor ids)

$ lsusb -tv                                => Show USB devices (read more
lsusb options)

$ lshal                                    => Show a list of all devices with
their properties

$ dmidecode                               => Show hardware info from the
BIOS (vendor details)

$ hdparm -i /dev/sda                      # Show info about disk sda

$ hdparm -Tt /dev/sda                      # Do a read speed test on disk sda

$ badblocks -s /dev/sda                   # Test for unreadable blocks on

disk sda

```

19. STATISTICS

```

$ top                                     => Display and update the top
cpu processes (30 example options)

$ mpstat 1                                 => Display processors related
statistics (learn mpstat command)

$ vmstat 2                                => Display virtual memory
statistics (very useful performance tool)

$ iostat 2                                 => Display I/O statistics (2sec
Intervals) (more examples)

$ tail -n 500 /var/log/messages    => Last 10 kernel/syslog
messages (everyday use tail options)

$ tcpdump -i eth1                         => Capture all packets flows on
interface eth1 (useful to sort network issue)

$ tcpdump -i eth0 'port 80'                => Monitor all traffic on port
80 ( HTTP )

$ lsof                                    => List all open files belonging
to all active processes.(sysadmin favorite command)

$ lsof -u testuser                        => List files opened by specific
user

$ free -m                                  => Show amount of RAM (daily
usage command)

$ watch df -h                            => Watch changeable data continuously(interesting
linux command)

```