

Module 3 – File system

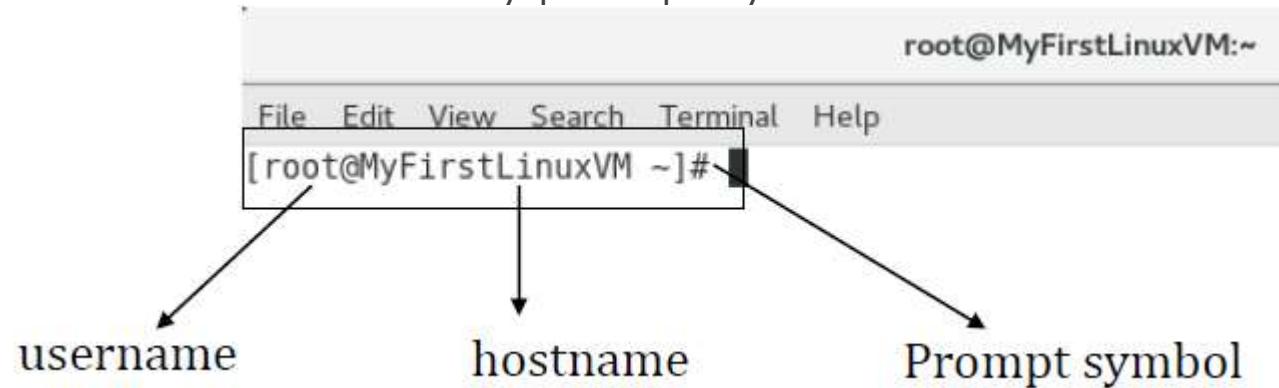
Important points to remember

- ▶ Linux has super-user account called root
 - ▶ root is the most powerful account that can create, modify, delete accounts and make changes to system configuration files
- ▶ Linux is case-sensitive system **ABC** is **NOT** same as **abc**
- ▶ Avoid using spaces when creating files and directories
- ▶ Linux kernel is not an operating system. It is a small software within Linux operating system that takes commands from users and pass them to system hardware
- ▶ Linux is mostly CLI not GUI
- ▶ Linux is very flexible as compared to other operating systems.

Command Prompts

What are command prompts?

- ▶ A command **prompt**, also referred to simply as a prompt, is a short text at the start of the command line followed by prompt symbol on a command line interface



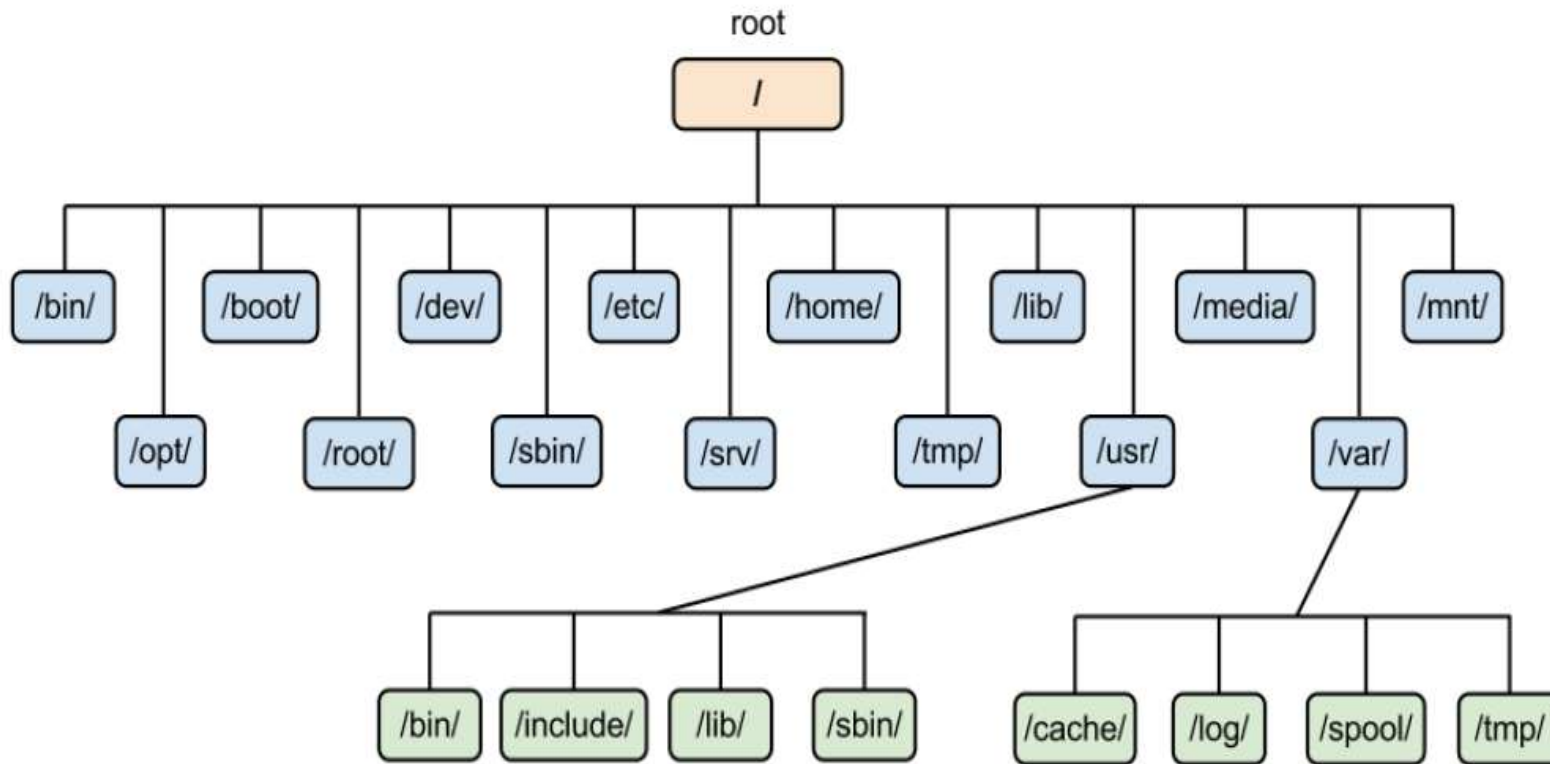
- ▶ To get your **prompt** back - **Ctrl + c**

Intro to File System

What is File System ?

- ▶ It is a **method** used by an operating system to **manage** files. The system **controls** how data is saved or retrieved
- ▶ Operating system stores files and directories in an organized and structured way
- ▶ There are many different types of **file systems**.
e.g. **ext3, ext4, xfs, NTFS, FAT etc.**
- ▶ Imp **Commands** for **Navigating** File system
 - ▶ **cd** – change directory
 - ▶ **pwd** – Print working directory
 - ▶ **ls** – lists (display the contents of a directory)

File System Structure



File System Description

/boot	Contains file that is used by the boot loader (grub.cfg)
/root	root user home directory. It is not same as /
/dev	System devices (e.g. disk, cdrom, speakers, flashdrive, keyboard etc.)
/etc	Configuration files
/bin → /usr/bin	Everyday user commands
/sbin → /usr/sbin	System/filesystem commands
/opt	Optional add-on applications (Not part of OS apps)
/proc	Running processes (Only exist in Memory)
/lib → usr/lib	C programming library files needed by commands and apps
strace -e open pwd	
/tmp	Directory for temporary files
/home	Directory for user
/var	System logs
/run	System daemons that start very early (e.g. systemd and udev) to store temporary runtime files like PID files
/mnt	To mount external filesystem. (e.g. NFS)
/media	For cdrom mounts.

Linux File / Directory Properties

Each file or directory in Linux has detail information or properties

Type	# of Links	Owner	Group	Size	Month	Day	Time	Name
drwxr-xr-x.	21	root	root	4096	Feb	27	13:33	var
lrwxrwxrwx.	1	root	root	7	Feb	27	13:15	bin
-rw-r--r--	1	root	root	0	Mar	2	11:15	testfile

What is Root

There are 3 types of root on Linux system

- ▶ **Root account:** root is an account or a username on Linux machine and it is the most powerful account which has access to all commands and files
- ▶ **Root as /:** the very first directory in Linux is also referred as root directory
- ▶ **Root home directory:** the root user account also has a directory located in /root which is called root home directory

File System paths

There are two paths to navigate to a filesystem

- ▶ Absolute Path
- ▶ Relative Path
- ▶ An absolute path always begins with a "/". This indicates that the path starts at the root directory.

```
cd /var/log/httpd
```

- ▶ A relative path does not begin with a "/". It identifies a location relative to your current position.

```
cd /var
```

```
cd log
```

```
cd httpd
```

Help Commands

- ▶ There are 3 **types** of **help** commands
 - ▶ **whatis** command
 - ▶ Command **--help**
 - ▶ **man** command

Working with files

- ▶ Creating Files
 - ▶ touch – used to create empty files
 - ▶ Touch filename
 - ▶ Touch filename1 filename2 etc
 - ▶ cp – copy file(s) to same or another directory
 - ▶ cp filename newfilename/dirname
 - ▶ cp filename1 filename2 <destination dir>
 - ▶ vi editor – vi filename to create a file and input data

File display cmds (working with contents of file)

- ▶ Following are used to display contents of a file
 - ▶ **cat**
 - ▶ **more**
 - ▶ **less**
 - ▶ **head**
 - ▶ **tail**
- ▶ Display the message /content on terminal
 - ▶ **echo**

Working with directories

- ▶ Creating Directories

- ▶ `mkdir`

- ▶ `mkdir -p dir1/dir2/dir3` (creates parent and child directories recursively)

- ▶ Copy a Directory

- ▶ `cp -r <source_folder><destination_folder>`

Working with files and Directories (remove)

- ▶ Remove files
 - ▶ `rm filename` or `rm filename1 filename 2 etc`
- ▶ Remove Directories - (to remove empty directory)
 - ▶ `rmdir dirname`
 - ▶ `rm -d dirname`
- ▶ Remove Directories - (to remove non- empty directory recursively)
 - ▶ `rm -rf dirname`

Working with files and Directories (move)

- ▶ Use **mv** to rename a file or to move the file to another directory
 - ▶ `mv filename newfilename -rename`
 - ▶ `mv filename dirname/newfilename -rename`
 - ▶ `mv filename dirname`
- ▶ The same mv command can be used to rename directories
 - ▶ `mv dirname newdirname -rename`
 - ▶ `mv dirname <destinationdir> -move dir to destination dir`

Find Files / directories

- ▶ Find files and directories using these cmds

- ▶ **find**

- ▶ Find all files in /etc and put the list in etcfiles.txt **find /etc > etcfiles.txt**
- ▶ Find all files of the entire system and put the list in allfiles.txt **find / > allfiles.txt**
- ▶ Find files that end in .conf in the current directory (and all subdirs).

find . -name "*.conf"

- ▶ Find files of type file (not directory etc.) that end in .conf.

find . -type f -name "*.conf"

- ▶ Find files of type directory that end in .bak . **find /data -type d -name "*.bak"**
- ▶ Find files that are newer than file42.txt **find . -newer file42.txt**

Find Files / directories

- ▶ The **locate** tool is very different from **find** in that it uses an index to locate files
- ▶ **Locate** uses a prebuilt database, which should be regularly updated, while **find** iterates over a file system to locate files.
- ▶ This is a lot faster than traversing all the directories, but it also means that it is always outdated.
- ▶ Become root and run **updatedb** to update **database**
- ▶ Most Linux distributions will schedule the updatedb to run once every day.
- ▶ Syntax : **locate filename**

File links -Soft and Hard Links

Inode = Pointer or number of a file on the hard disk

- ▶ **Soft Link** = Link will be removed if file is removed or renamed
 - ▶ ln -s pathoffile
- ▶ **Hard Link** = Deleting renaming or moving the original file will not affect the hard link
 - ▶ ln pathoffile

for directories hard links cannot be created but soft links can be created.

