RedHat/CentOS 7 Fundamentals

THINKNYX
TECHNOLOGIES

Yogesh Raheja
+91-9810344919
yogesh.raheja@thinknyx.com

Topics

- Brief History of Unix/Linux
- Introduction to Linux/RHEL
- The Linux kernel
- Introduction to the Shell
- Some common commands on shell
- Features of shell
- VI editor
- Lab exercise

Brief history of Unix

- In 1960's MIT, Bell Labs and GE were working on a timesharing operating system called **multics** but that could not be successful.

- Unix was originally developed at AT&T Bell Laboratories by Ken Thompson and Dennis Ritchie in 1969-1970. **Was called AT&T Unix.**

- In 1973, Unix was migrated from assembly language to C. This made possible for Unix to be installed and run on many different types of computers which had a C compiler.

Brief history of Unix

- AT&T gave Unix to the Universities for research. The first one was University of California at Berkeley, that developed "Berkeley Unix" or BSD (Berkeley Standard Distribution).

- The influence of Unix in academic circles led to large-scale adoption of Unix by commercial startups, including HP-UX, Solaris, AIX, and Xenix

Popular Unix versions

Version	Developped By
UNIX SYSTEM	AT & T
UNIX BSD 4.x.x	BERKELEY SW DIST
XENIX	MICROSOFT
ULTRIX	DEC
AIX	IBM
HP-UX	HP
SOLARIS	SUN MICROSYSTEM

GNU/GPL

- In earlier days, programmers used to share code and software among themselves but this trend changed when software started to be used under a business model.

- Companies started restricting sharing of source code and started selling software under copyrights.

GNU/GPL

- Frustrated by this, Richard Stallman founded GNU project under which he planned to develop and distribute software as opensource and free.
- He formed GPL (General Public License) which laid rules as to how the free software can be used and distributed.
- Most commonly known as the GPL.

History of Linux

- In the 1985, a professor by name Andy Tanenbaum wrote a Unix like Operating system from scratch for the Intel i386 platform. He named it Minix. It was used for teaching students about operating systems.

- In 1990, Linus Torvalds came to know about Minix and wanted to upgrade it by putting in more features and improvements but he was prohibited by Tanenbaum to do so.

- Then Linus decided to write his own kernel and released it under GPL. This is now popularly known as Linux.

RHEL

- As Linux kernel was free and opensource, so anyone was free to modify and distribute it under GPL.
- So, lots of "Linux distributions", both completely free and commercialized started appearing. All used the same Linux kernel.
- One of those was Red Hat Linux which was the product of a U.S. company called Red Hat in 1993.
- Red Hat Enterprise Linux 7 is the latest commercial offering from Red Hat.

Popular Linux Distibutions

- CentOS
- Ubuntu
- Debian
- Fedora
- Linux Mint
- SuSE
- Slackware
- RHEL

Installation of Linux

- To install CentOS Linux OS, you will need to download the ISO images (CD Images) of the installation CD-ROM from https://www.centos.org/download/

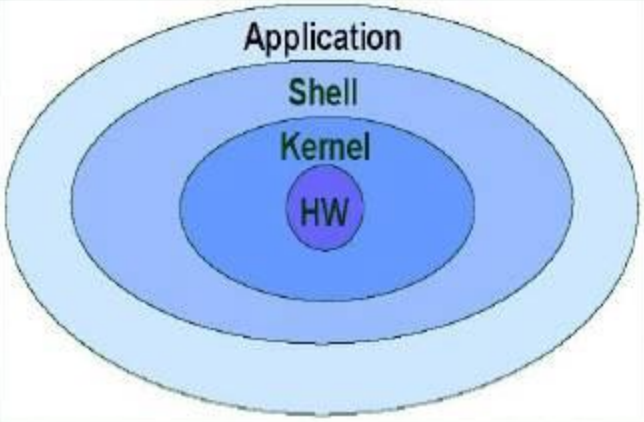
- Lets begin with the Installation and then understand about the major Linux concepts.

Adobe Acrobat Document

Linux Architecture

- **Kernel:** the heart of the UNIX Operating System. Keeps track of the disks, tapes, printers, terminals, communication lines and any other devices attached to the computer. It also interfaces between the computer's hardware and the users.

- **Shell:** a program that interfaces between the user and the UNIX Operating System. It listens to the user's terminal and translates the actions requested by the user. There are a number of different Shells that may be used.



- **Application:** application programs, such as Word Processors, spreadsheets and Database Management Systems, may be installed alongside the UNIX Commands. A user may run an utility or application through the shell.

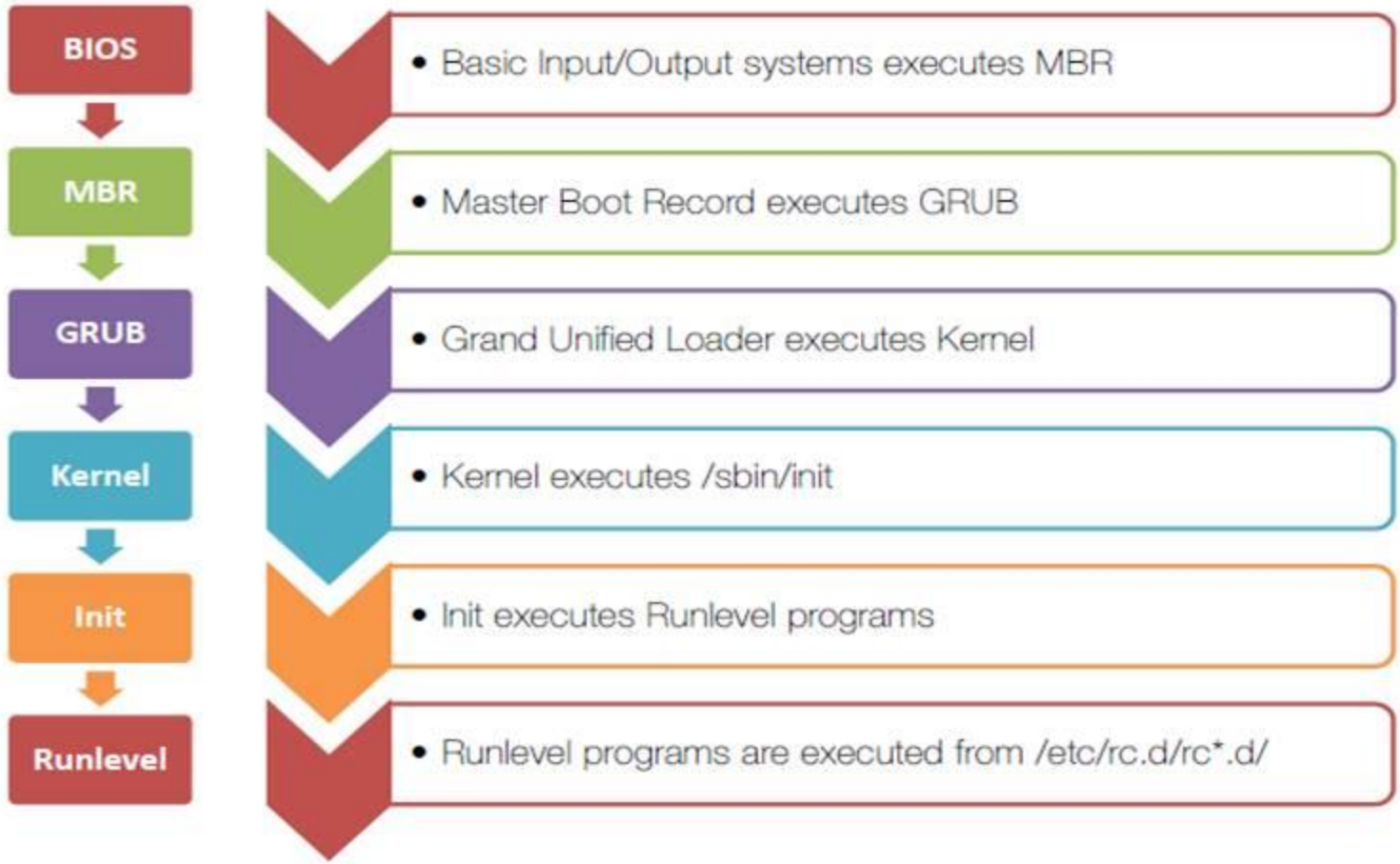
What is Kernel

- Kernel is the program that controls the resources of the computer and allocates them to the user.
- Kernel basically consists of
  - code for process and memory management
  - code for I/O from hardware
  - code for handling filesystem
- Unix kernel is set of essential routines and data structures that are always stored in main memory when the system is running.
- Rest of the OS is stored on disk, to be paged in as necessary.

Functions of Kernel

- Process Management
  - Allocation of resources to a process
  - Synchronization among processes
- Memory Management
  - Allocating and deallocating memory to programs
- Device Management
  - Controlling several devices attached to the system
- Storage Management
  - Disk Management
  - Disk scheduling

System Startup Sequence



System runlevels

- Runlevels act as a method to define what processes are started and stopped. RHEL (upto ver 6) uses 7 runlevels (0-6).

- In RHEL 7 runlevels are replaced by targets.

0 – Halt

1 – Single user text mode

2 – Not used (user – definable)

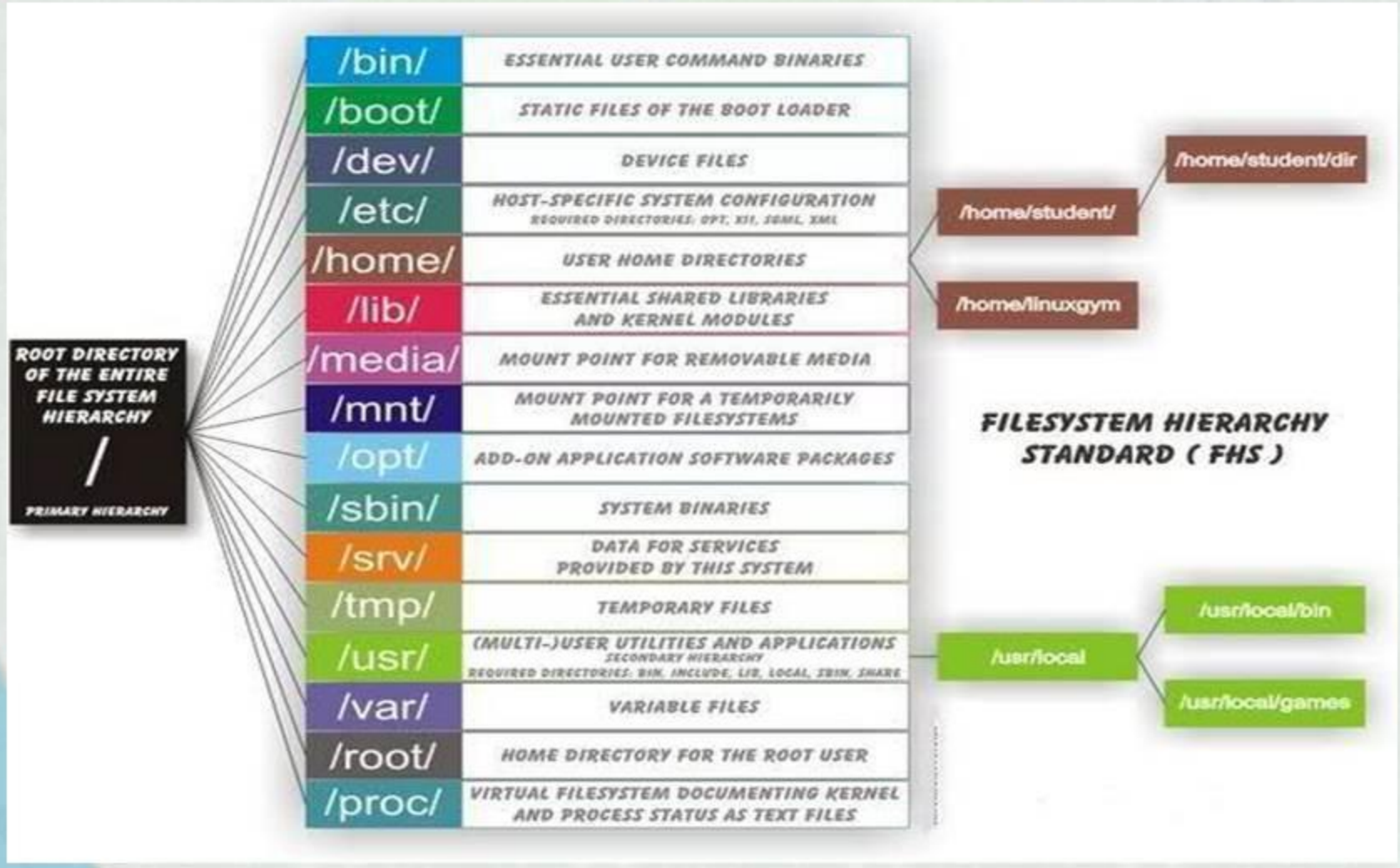
3 – Full multi-user text mode

4 – Not used – (user – definable)

5 -- Full multi-user graphical mode (with an X-based login screen)

6 -- Reboot

Directory Hierarchy



Some important system files

/etc/fstab	•All mounted filesystems can be found here and new can be added.
/etc/resolve.conf	•To add DNS server entry here for resolving your hostname with it.
/etc/ssh/sshd_config	•For enabling remote access via SSH port 22
/etc/passwd @ /etc/group	•having password and shell entries •and having group entries
/etc/inittab	•Having details of all run levels
/etc/init.d	•Contains links to the run-level scripts. These scripts are linked from files in the /etc/rc?.d directories to have each service associated with a script started or stopped for the particular run level. The ? is replaced by the run-level number (0 through 6).
/etc/hosts	•Contains IP addresses and hostnames that you can reach from your computer. (Usually this file is used just to store names of computers on your LAN or small private network.)
/etc/hosts.allow	•Lists host computers that are allowed to use certain TCP/IP services from the local computer.
/etc/hosts.deny	•Lists host computers that are not allowed to use certain TCP/IP services from the local computer (doesn't exist by default).

Filesystems

- Organization method of data on a hard disk volume.
- Many filesystems are present e.g. FAT, NTFS, ext2, ext3, reiserfs, xfs etc. RHEL6 uses ext4 by default and RHEL7 supports XFS by default.

```
Filesystem                      1024-blocks      Used  Available     Mounted
on

/dev/mapper/rootvg-root_lv           999320    456096     490796     /
/dev/sda1                            245679     28026     204546     /boot
/dev/mapper/rootvg-home_lv           999320      1304     945588     /home
/dev/mapper/infravg-images_lv       3997376      8184    3779480     /images
/dev/mapper/rootvg-opt_lv           2031440    322312    1604284     /opt
/dev/mapper/rootvg-system_lv         122835      9053     107229     /system
/dev/mapper/rootvg-tmp_lv            999320      3544     943348     /tmp
/dev/mapper/rootvg-usr_lv           3997376    800080    2987584     /usr
/dev/mapper/rootvg-var_lv           3063568    291584    2614724     /var
```

Users

- Two types of users
  - Privilege user “root” (with uid=0). Root user has unrestricted privileges on a Unix system
  - Other Non-privieges users. These are normal users and cannot access OS configuration files and access is restricted.

Types of Files

- In unix/linux, everything is a treated as a file that can be read or written
- Three types of files
  - Directory
    - `drwxrwxrwt. 5 root root 4096 May 30 03:32 /tmp`
  - Special files (device/character/block files)
    - `brw-rw---- 1 root disk 8, 0 May 13 12:54 /dev/sda`
    - `crw-rw-rw- 1 root tty 5, 0 May 13 12:54 /dev/tty`
  - Ordinary files
    - `-rw------- 1 root root 1542 Mar 10 07:13 /etc/fstab`

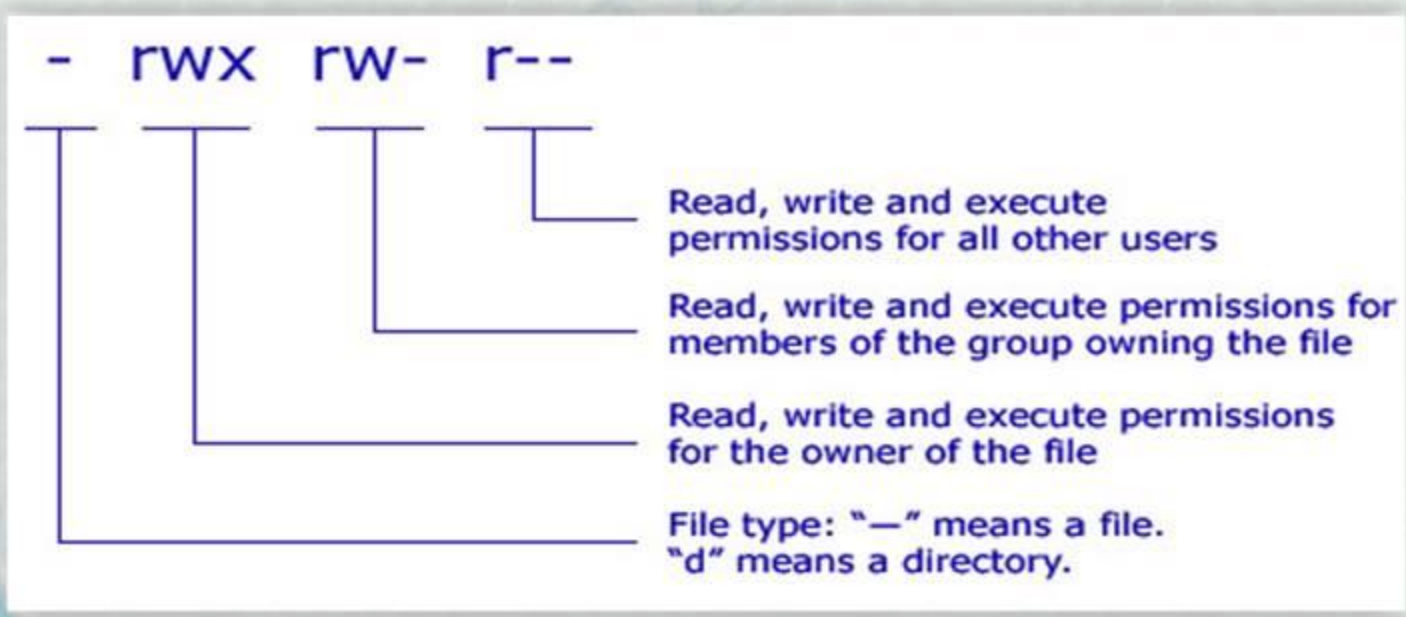
File Permission Basics

- Every user in Linux also belongs to a group associated to him.
- Every user is assigned a user id and a group id.
- Three categories of users
  - The Owner
  - The Group
  - Others
- For each category, the system keeps track of three sorts of permissions
  - read
  - write
  - execute

File Permissions

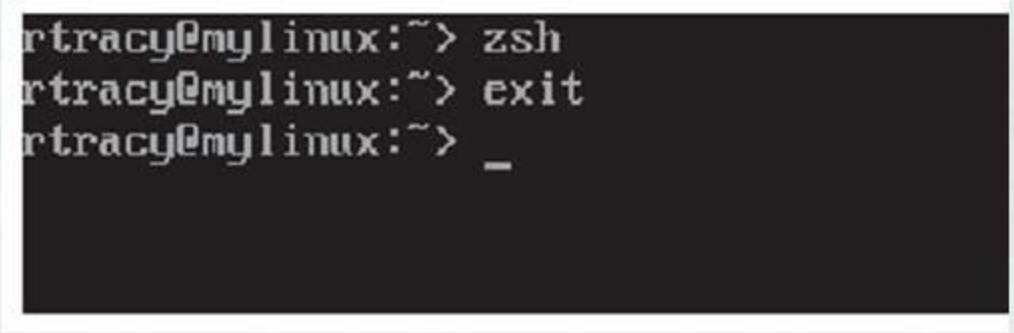
```
-rw-rw-r--    1 tclark   authors        360 Jan 13 17:48 preamble.txt
```

Permissions Owner Group



Introduction to Shell

- Linux shells: A shell is a command interpreter that allows you to type commands from the keyboard to interact with the operating system kernel.
- It's the Linux command line interface through which you can launch processes and applications
- Every user is assigned a default shell. Whenever you login to a Linux system, you get a shell prompt according to the type of default shell.



Types of shells

- sh (Bourne Shell): The sh shell was the earliest shell, being developed for UNIX back in the late 1970s.
- bash: (Bourne-Again Shell) The bash shell is an improved version of the sh shell and is one of the most popular shells today. It's the default shell used by most Linux distributions.
- csh (C Shell): The csh shell was originally developed for BSD UNIX. It uses a syntax that is very similar to C programming.
- zsh: (Z Shell) The Z Shell is an improved version of the bash shell.

Common commands

- uptime
  - In Linux uptime command shows since how long your system is running and the number of users are currently logged in and also displays load average for 1,5 and 15 minutes intervals.

```
# uptime
 09:59:50 up 16 days, 21:05,  1 user,  load average: 0.00, 0.00, 0.00
```

- id
  - Prints user and group information for the specified USERNAME or current user

```
# id
uid=0(root) gid=0(root) groups=0(root),600(staff)
# id osadmin
uid=249(osadmin) gid=249(osadmin) groups=249(osadmin),600(staff)
```

Common Commands

- who
  - It will displays users currently logged in and their process along-with shows load averages. also shows the login name, tty name, remote host, login time, idle time, JCPU, PCPU, command and processes.

```
# who
 10:15:05 up 16 days, 21:20,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     pts/0    172.20.20.98      08:57    0.00s  0.02s  0.00s w
```

Common Commands

- su
  - substitute user, using this command you can change and login to a new user.

```
# su - newuser
```

- whoami
  - whoami command print the name of current user. You can also use "who am i" command to display the current user. Use "who am i" command if you want to know the exact user logged in.

```
# whoami
osadmin

# who am i
root     pts/0        2016-05-30 08:57 (172.20.20.98)
```

Common Commands

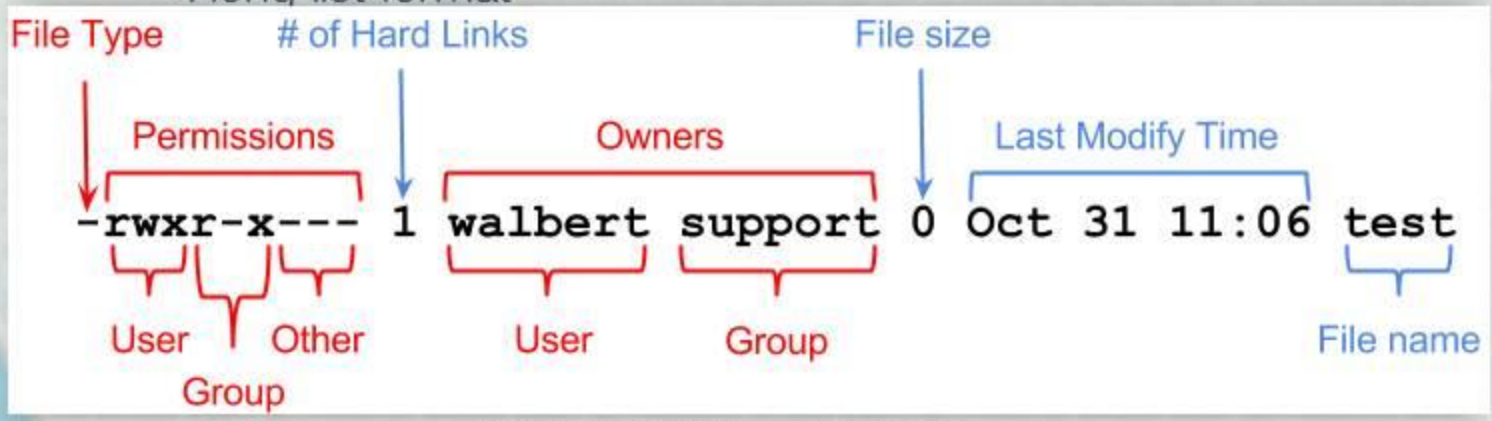
- ls
  - List contents

ls [-adlR] [pathname(s)]

-a all entries (hidden files too)

-d if argument is a directory list only the name (instead of the contents)

-l long list format



Common Commands

- cd
  - Change directory

```
# cd /tmp     [Change current directory to /tmp]
# cd ~        [Change to home directory]
# cd ..       [Change to parent directory]
# cd -        [Change to previous working directory]
```

- pwd
  - Print current working directory.

```
# pwd
/tmp
```

Common Commands

- more
  - Display output page wise one screen at a time. Doesn't allow backward movement.

```
# more filename
```

- less
  - More advanced version of more. Allows backward movement also. Faster than more for large files.

```
# less filename
```

Common Commands

- date
  - Display or change current date on system

```
# date
Mon May 30 10:57:57 CEST 2016
```

- man
  - To get help and manual pages for any command available.

```
# man <command>
```

Common Commands

- tail
  - Display last lines of a file

tail [-n] [*filename* ...] default last 10 lines

```
# tail -40 messages          [Display last 40 lines]
# tail -f messages           [Display and follow changes to file]
```

- head
  - Display first lines of a file.

head [-n] [filename ...] default first 10 lines

```
# tail -40 messages          [Display first 40 lines]
```

Common Commands

- mkdir
  - Make new directory

```
# mkdir /tmp/mydir
```

- rmdir
  - Remove empty directory

```
# rmdir /tmp/mydir
```

Common Commands

- df
  - Display all mounted filesystems along with space used and available

```
# df -hP
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/rootvg-root_lv      976M  446M  480M  49% /
tmpfs                           939M     0  939M   0% /dev/shm
/dev/sda1                       240M   28M  200M  13% /boot
/dev/mapper/rootvg-home_lv      976M  1.3M  924M   1% /home
/dev/mapper/infravg-images_lv   3.9G  8.0M  3.7G   1% /images
/dev/mapper/rootvg-opt_lv       2.0G  315M  1.6G  17% /opt
/dev/mapper/rootvg-system_lv    120M  8.9M  105M   8% /system
/dev/mapper/rootvg-tmp_lv       976M  3.5M  922M   1% /tmp
/dev/mapper/rootvg-usr_lv       3.9G  782M  2.9G  22% /usr
/dev/mapper/rootvg-var_lv       3.0G  285M  2.5G  11% /var
```

Common Commands

- cp
  - Copy a file or directory

    cp [-i] file new_file

    cp [-i] file [file ...] dest_dir

    cp -r [-i] dir [dir ...] dest_dir

    Example :

```
$ cp Report.log Report.log.bck
$ cp Report.log Report.log.bck Reports/
$ cp -r Reports Reports_Old
```

-i is interactive (wait for user confirmation)

-r recursively, used for directories

Common Commands

- rm
  - Removes a file or directory

    rm [-if] *filename* [*filename ...*]

    rm -r[if] *dirname*

    Example :

    $ rm rm.log

    $ rm –r xyz.dir

-i is interactive (wait for user confirmation)

-f force remove

-r recursively, used for directories

Common Commands

- mv
  - Moves and renames a file or directory

    mv [-if] *oldfile newfile*

    mv [-if] *olddir newdir*

    Example :

    $ mv *oldfile newfile*

-i is interactive (wait for user confirmation before overwrite)

-f force and do not prompt

Features of the Shell

- Command line interpretation
- Metacharacters and substitution
- Input – output redirection
- Pipeline connection
- Background/foreground processes
- command grouping
- Shell environment variables

VI Editor

- 'vi' is a a screen-oriented text editor. It was designed to be terminal independent, command have been mapped to almost every key of the standard keyboard.
  - Command mode : keystrokes are interpreted as commands ESC – puts you in command mode
  - Input mode : keystrokes are entered into the file

VI Editor

w = move forward word by word

b = move backwards word by word

^ = go to the beginning of the current line

$ = go to the end of the current line

G = go to the end of the file

CTRL + g = Reports the current line number

CTRL + b = Scroll back to previous window of text

CTRL + f = scroll forward to next window of text

H = Go home (first line,first character of the screen)

:# = Go the specific line (I.e. :3, go to the line number #)

VI Editor

- If you like to see line numbers while you are editing your file you can enter the command :

  :set number

- You can disable line numbers with

  :set nonumber

- Input Mode: I, a, o

  a append new text after the cursor

  i insert new text before the cursor

  O open a line for text above the current line

  o open a line for text below the current line

  A append new text at end of the line

  I insert new text at beginning of the line

- Pressing ESC key concludes an input session (switch to vi command mode)

VI Editor

- Deleting Text: x, dw, dd, dG

x	**delete the character at the cursor**
dw	**delete the current word**
dd	**delete the current line**
dG	**delete through the last line of the file**
d$	**delete to the end of the line**
d^	**delete to the beginning of the line**

- You might delete or modify something that was not intended to be deleted or modified. The undo command will undoubtedly come to your rescue:

u	**undo** the last modification
U	**undo all** modifications to current line

VI Editor

- Moving Text: p,P
  - p paste contents of the buffer back into the text after the cursor
  - P paste contents of the buffer into the text before the cursor
- Copying Text: yw, yy
  - yw yank the current word
  - yy yank the current line
  - yG yank through the last line
  - y$ yank to the end of the line
  - y^ yank to the beginning of the line
- Changing Text: r, R, cw,
  - r character replaces the character at the current cursor
  - R replaces all characters
  - cw change the current word
  - cc change the current line entirely
  - cG change through the last line of the file
  - c$ change to the end of the line
  - c^ change to the beginning of the line

Hostname Settings

- hostnamectl is the utility to check the status and rename host name related settings in RHEL/CENTOS 7.x
- uname -a
- hostnamectl -h
- hostnamectl set-hostname thinknyx
- hostnamectl status
- uname -a

Note: Run bash on the same session to get the name reflected in the same session.

Networking

- /sbin/ip address is the main command used for Networking tasks in RHEL/CENTOS 7.x

```
[student@desktopX ~]$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,❶UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
 1000
    ❷link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff
    ❸inet 172.25.0.10/24 brd ❹172.25.0.255 scope global eth0
       valid_lft forever preferred_lft forever
    ❺inet6 fe80::5054:ff:fe00:b/64 scope link
       valid_lft forever preferred_lft forever
```

❶ An active interface has the status of **UP**.

❷ The link line specifies the hardware (MAC) address of the device.

❸ The inet line shows the IPv4 address and prefix.

❹ The broadcast address, scope, and device name are also on this line.

❺ The inet6 line shows IPv6 information.

Networking

- Display the IP address and netmask for all interfaces

ip addr

- Display the statistics for ens33 interface

ip -s link show ens33

- Display routing information

ip route

- Verify the accessibility of the router

ping -c3 <gateway>

Networking

- Check the IP information

[root@thinknyx ~]# ip addr show ens33

- Restart the network services

[root@thinknyx ~]# systemctl restart network

- Login with the new IP

Disk Management

- What is a Filesystem: In simple term you can say a FS is a medium to access your files/directories in any Unix Flavour. In Unix each and every thing is File or Directory. In order to standardize Files or Directories structure Filesystem are used.

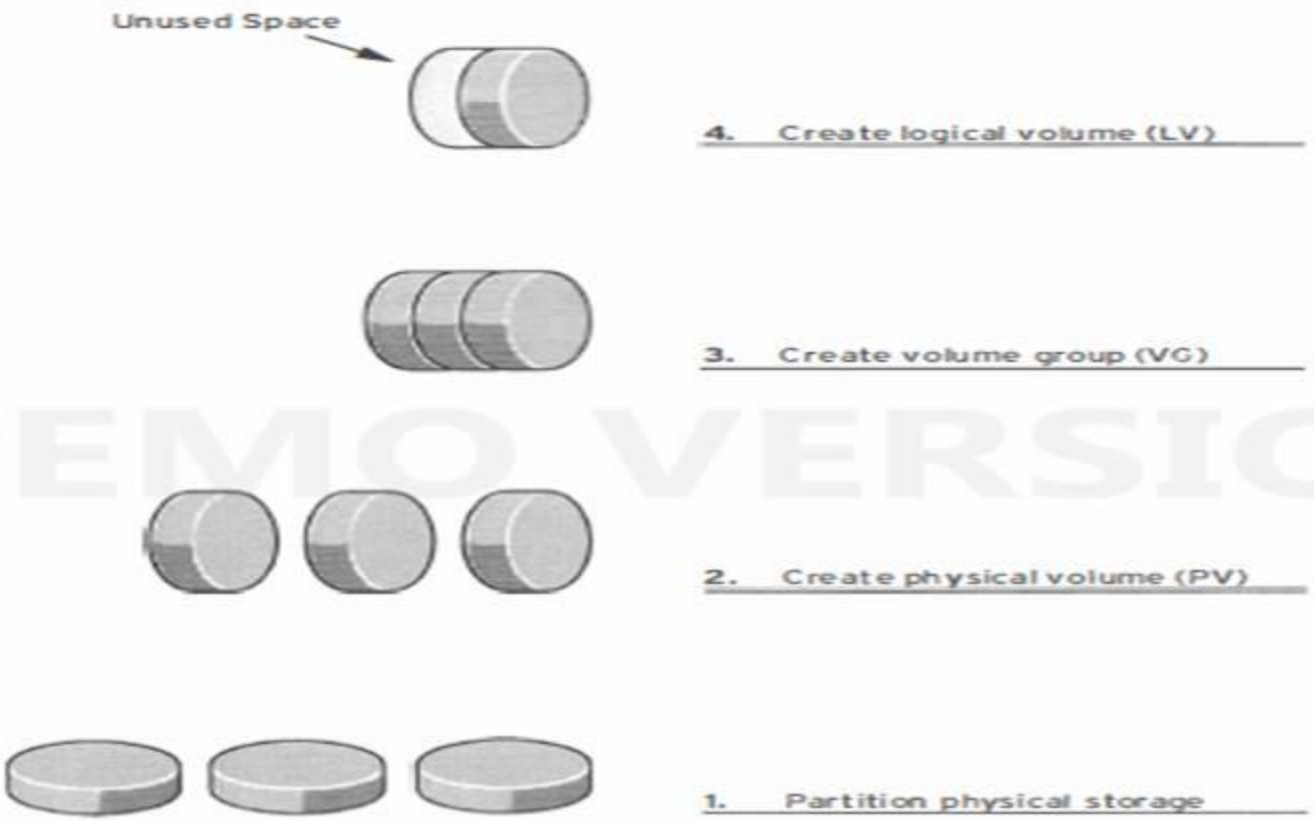
- Now in depth you can say a FS is a portion of your harddisk (or complete harddisk) which is used to create files and directories which are world-wide accessible.

Linux Filesystem routemap

- **Identify Device** –> which may be your internal harddisk or external storage. (/dev/sd{a,b,c,....} –> SCSI , /dev/hd{a,b,c,...} –> IDE)
- **Partition Device** –> Disks are partitioned to make separate Filesystems as per requirement.
- **Make Filesystem** –> Partitions are used to make Filesystems so that they can be recognized by the OS.
- **Mountpoint** –> Mountpoints are nothing but just are simple directories where Filesystems are mapped. Its easy to remember simple directory names as compare to complete FS name.
- **Make FS permanent after reboot** –> The FS entry should be made under a special file called /etc/fstab which is used by init phase to mount all the filesystems listed under /etc/fstab.

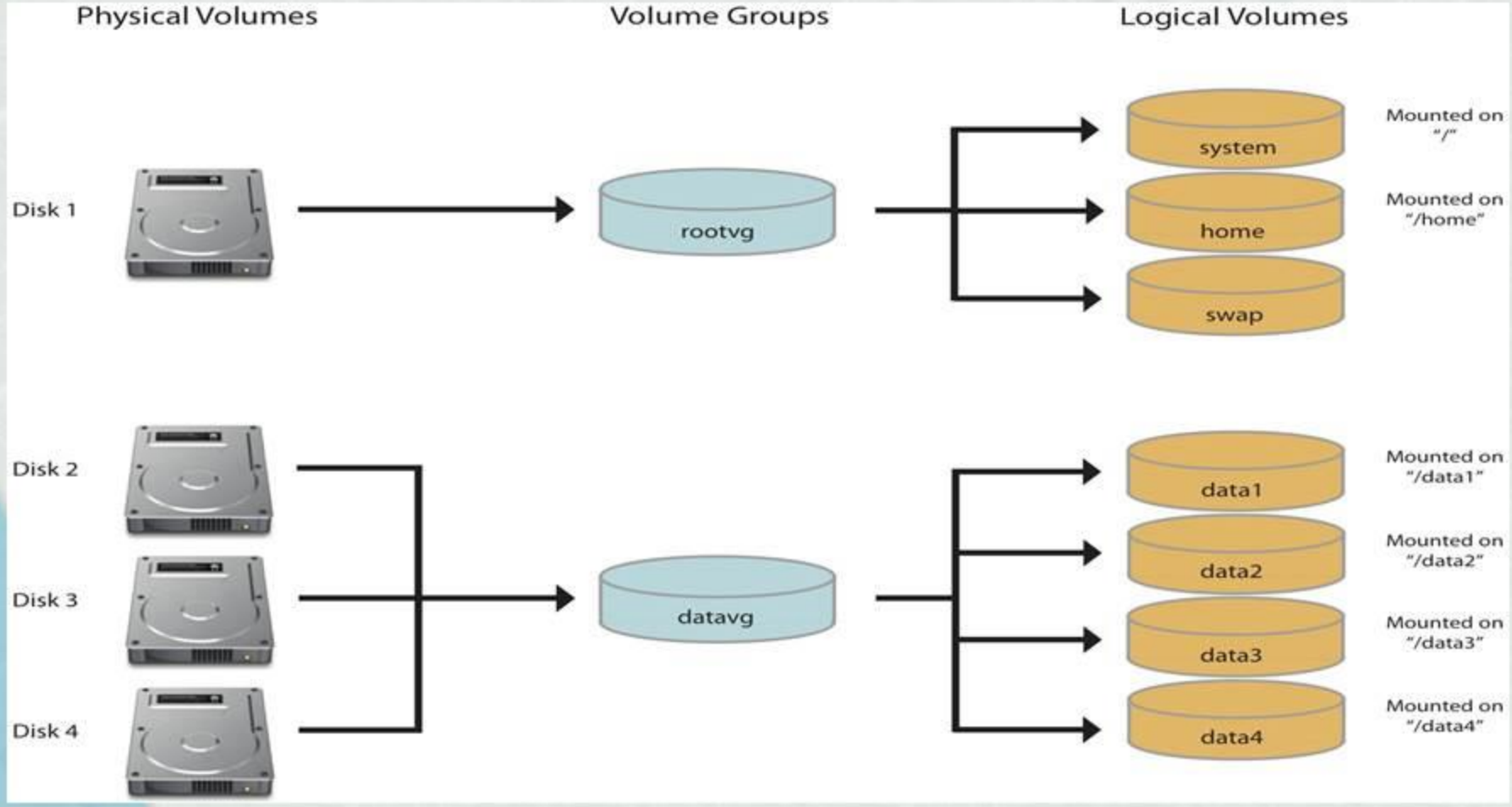
Linux LVM

- LVM stands for Linux Volume Manager.



Linux LVM

- LVM stands for Linux Volume Manager.



Linux LVM

- **Harddrive/Storage Luns** —> The physical media where are read and writes occurred.
- **Physical Volume** —> The actual hard drive or a partition of hard drive can be assigned to Physical Volume. Its the smallest building unit in Linux LVM.
- **Volume group** —> The combination of Physical Volume is used to make a Volumegroup. You can think Volume group as a Pool of harddrives/partitions.
- **Logical Volume** —> We can create Logical Volume on any Volumegroup which indeed are created on Physical Volume which indeed are on Physical disks.
- **Mountpoint** —> The Filesystem is created on Logical Volume and is mounted on mountpoint which is only a simple directory to make the Filesystem open to world for read write operations.

Package Management

- **Yellowdog Updater, Modified (yum)** is an open-source command-line package-management utility for computers running the Linux operating system
- RPM Software Packages and Yum - RPM packages re grouped into yum repositories, provide a uniform method of tracking software installation and updates.
- Yum is used to install, remove and update software packages.
- Repositories for yum are configured in the **/etc/yum.repos.d** directory
- **yum-config-manager** is a dedicated utility in RHEL/CENTOS 7.x to manage software repositories.
- Log file is **/var/log/yum.log**

Package Management

- Packages can be located, installed, updated and removed by the name or by package groups.

Task:	Command:
List installed and available packages by name	`yum list [NAME-PATTERN]`
List installed and available groups	`yum grouplist`
Search for a package by keyword	`yum search KEYWORD`
Show details of a package	`yum info PACKAGENAME`
Install a package	`yum install PACKAGENAME`
Install a package group	`yum groupinstall "GROUPNAME"`
Update all packages	`yum update`
Remove a package	`yum remove PACKAGENAME`
Display transaction history	`yum history`

Package Management

- Packages outside YUM repositories can be located, installed, updated and removed using RPM utility.

Syntax	Description	Example(s)
rpm -ivh {rpm-file}	Install the package	rpm -ivh mozilla-mail-1.7.5-17.i586.rpm rpm -ivh --test mozilla-mail-1.7.5-17.i586.rpm
rpm -Uvh {rpm-file}	Upgrade package	rpm -Uvh mozilla-mail-1.7.6-12.i586.rpm rpm -Uvh --test mozilla-mail-1.7.6-12.i586.rpm
rpm -ev {package}	Erase/remove/ an installed package	rpm -ev mozilla-mail
rpm -ev --nodeps {package}	Erase/remove/ an installed package without checking for dependencies	rpm -ev --nodeps mozilla-mail
rpm -qa	Display list all installed packages	rpm -qa rpm -qa | less
rpm -qi {package}	Display installed information along with package version and short description	rpm -qi mozilla-mail
rpm -qf {/path/to/file}	Find out what package a file belongs to i.e. find what package owns the file	rpm -qf /etc/passwd rpm -qf /bin/bash
rpm -qc {pacakge-name}	Display list of configuration file(s) for a package	rpm -qc httpd
rpm -qcf {/path/to/file}	Display list of configuration files for a command	rpm -qcf /usr/X11R6/bin/xeyes
rpm -qa --last	Display list of all recently installed RPMs	rpm -qa --last rpm -qa --last | less
rpm -qpR {.rpm-file} rpm -qR {package}	Find out what dependencies a rpm file has	rpm -qpR mediawiki-1.4rc1-4.i586.rpm rpm -qR bash

Services and Daemons

- A History - For many years, process ID 1 of Linux/Unix Systems has been the init process.
- This init process (or PID 1) was responsible for activating other services on the system.
- Most of the required daemons and services were started on the system at book time with System V and other LSB (Linux Standard Base) init scripts.
- Less frequently used daemons were started on demand by another services, such as initd or xinetd.
- These legacy systems have several limitations, which addressed with latest "systemd" in latest Linux flavors.

Services and Daemons

- In Redhat/CentOS 7.x, process ID 1 is systemd - "the new init system".
- Parallelization capabilities, which will increase the boot speed of a system.
- On-demand starting of a daemons without requiring a separate service.
- Automatic service dependency management, which prevent long timeouts which was a big challenge earlier.
- systemctl utility is used to manage different types of systemd objects, called units.
- A list of available unit types can be displayed with systemctl -t help.

Services States

- The state of a service can be viewed with "systemctl status name.type. If the unit type is not given, systemctl wil show the status of a service unit, if exists.

```
[root@thinknyx~]#systemctl status ntpd
● ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
```

```
[root@thinknyx~]#systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-09-11 03:05:25 PDT; 18min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 1082 (sshd)
   CGroup: /system.slice/sshd.service
           └─1082 /usr/sbin/sshd

Sep 11 03:05:25 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 11 03:05:25 localhost.localdomain systemd[1]: PID file /var/run/sshd.pid not readable (yet?) after start.
Sep 11 03:05:25 localhost.localdomain sshd[1082]: Server listening on 0.0.0.0 port 22.
Sep 11 03:05:25 localhost.localdomain sshd[1082]: Server listening on :: port 22.
Sep 11 03:05:25 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
Sep 11 03:17:00 localhost.localdomain sshd[22395]: Accepted password for root from 192.168.74.1 port 59689 ssh2
[root@thinknyx~]#
```

Services States

- Systemctl status <services-name> has replaced service <service-name> status.
- Various status with systemctl are given below:

Keyword:	Description:
loaded	Unit configuration file has been processed.
active (running)	Running with one or more continuing processes.
active (exited)	Successfully completed a one-time configuration.
active (waiting)	Running but waiting for an event.
inactive	Not running.
enabled	Will be started at boot time.
disabled	Will not be started at boot time.
static	Can not be enabled, but may be started by an enabled unit automatically.

Services States

- Query the state of all units:

systemctl

- Query the state of only the service units

systemctl --type=service

- Troubleshooting with long listing of the status of the units:

systemctl status ntpd -l

- To check the unit is active, enabled at boot or not:

systemctl is-active ntpd
systemctl is-enabled ntpd

- List the active state of all loaded units. "--all" will add inactive units also.

systemctl list-units --type=service
systemctl list-units --type=service --all

- View only failed services

systemctl --failed --type=service

Controlling System Services

- Whenever we change application or database or OS configuration files, maximum time we have to restart the services to make the changes take effect. Lets perform some of the tasks to understand it.

- View the status of a service and verify that the process is running.

systemctl status sshd.service
ps -up PID

- Stop and verify the service

systemctl stop sshd.service
systemctl status sshd.service

- Start and view the status

systemctl start sshd.service
systemctl status sshd.service

Controlling System Services

- Stop and start, the service with one command

systemctl restart sshd.service
systemctl status sshd.service

- Read and relaod the configuration file without complete stop and start. This time the process ID will not change

systemctl reload sshd.service
systemctl status sshd.service

- Special Note:

[root@thinknyx~]#systemctl list-dependencies sshd

Controlling System Services

- Disable the service and view the status, note: disabling doesn't mean service will get stop .

systemctl disable ntpd
systemctl status ntpd

- Enable a service and view the status. note: enabling doesn't mean service will fet started.

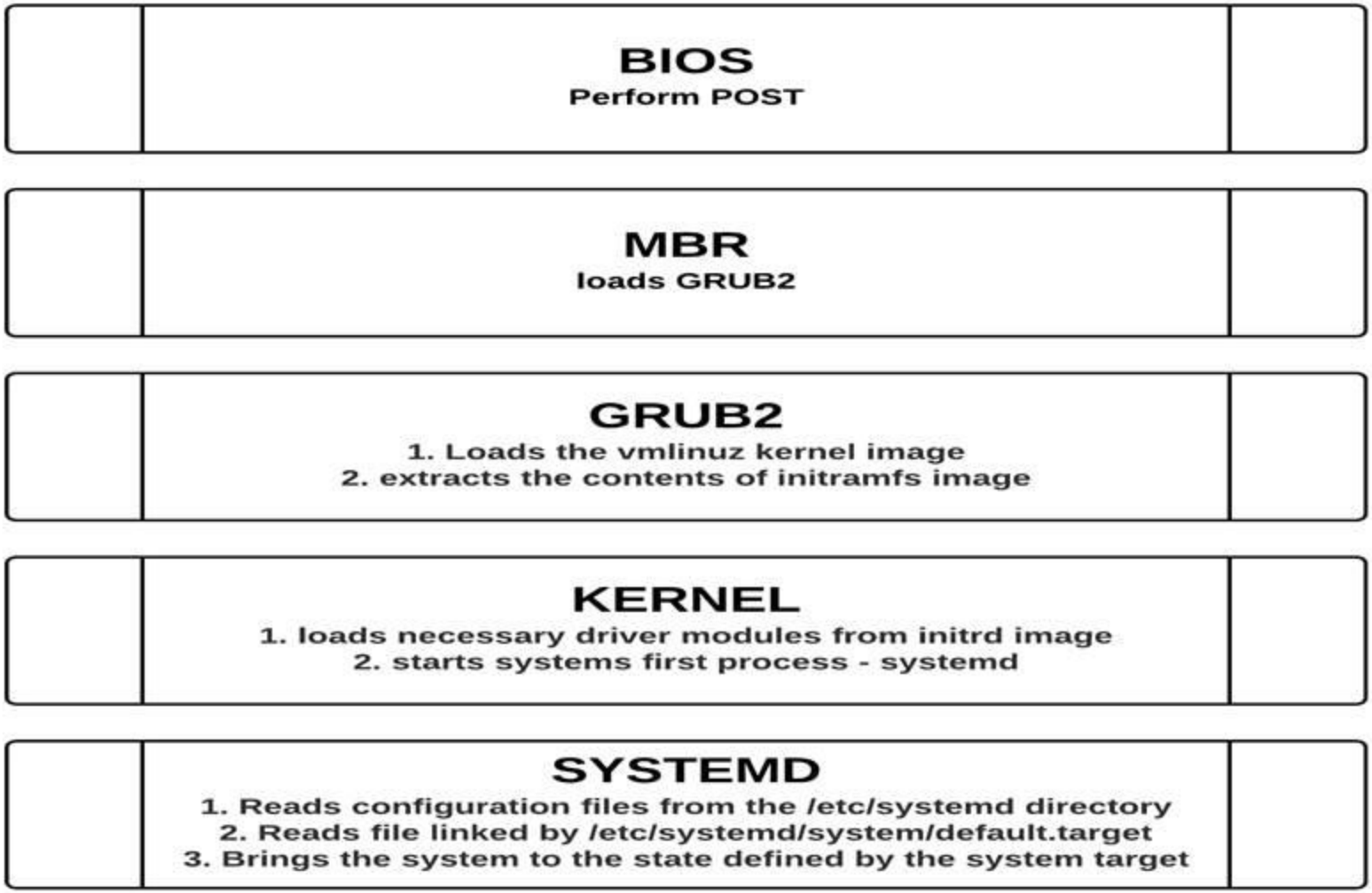
systemctl enable ntpd
systemctl status ntpd

Controlling System Services

- Summary of systemctl commands:

Task:	Command:
View detailed information about a unit state.	`systemctl status UNIT`
Stop a service on a running system.	`systemctl stop UNIT`
Start a service on a running system.	`systemctl start UNIT`
Restart a service on a running system.	`systemctl restart UNIT`
Reload configuration file of a running service.	`systemctl reload UNIT`
Completely disable a service from being started, both manually and at boot.	`systemctl mask UNIT`
Make a masked service available.	`systemctl unmask UNIT`
Configure a service to start at boot time.	`systemctl enable UNIT`
Disable a service from starting at boot time.	`systemctl disable UNIT`
List units which are required and wanted by the specified unit.	`systemctl list-dependencies UNIT`

RedHat/CentOS 7.x Boot Process



Questions & Answers

THANK YOU

For any queries or questions, please contact:

support@thinknyx.com
yogesh.raheja@thinknyx.com
yogeshraheja07@gmail.com