LINUX BASIC ADMINISTRATION

- **OS and Linux**

- PRESENTATION BY V.SANTOSH

UNIX HISTORY

- 1965 Bell Laboratories joins with MIT and General Electric in the development effort for the new operating system, Multics.
- 1969 AT&T was unhappy with the progress and drops out of the Multics project. Some of the Bell Labs programmers who had worked on this project, Ken Thompson, Dennis Ritchie, Rudd Canaday, and Doug McIlroy designed and implemented the first version of the UNIX File System on a Programmed Data Processor (PDP-7) system.
- 1973 UNIX is re-written in C, new programming language developed by Dennis Ritchie.
- 1989, AT&T and Sun Microsystems joined together and developed system V release 4 (SVR4)

DIFFERENT FLAVOURS OF UNIX

- Different flavours of UNIX are:

BSD	1977	Univ. Of California, Berkley
Xenix	1980	Microsoft, but later discontinued
Sun OS	1980	Sun Microsystems (Oracle)
AIX	1985	IBM
HP/UX	1985	HP
SCO UNIX	1988	SCO
Mac OS X	1999	Apple Computers

OPEN SOURCE

- Open source: Software and source code available to all
- The freedom to distribute software and source code
- The ability to modify and create derived works
- Integrity of author's code
- The Free Software Foundation and the Four Freedoms

LINUX HISTORY

- 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 DISTRIBUTIONS

- Linux distributions:
  - Are built on top of the Linux kernel
  - Are full operating systems plus more
  - Include compiled binaries and source code
- There are hundreds of Linux distributions.
  - Commercially-backed distributions
  - Linux community–driven distributions
- Example:
  - Oracle Linux, Debian, Fedora, Red Hat Enterprise Linux (RHEL), SUSE,Ubuntu, Canonica ,CentOS etc

UNIX AND LINUX DIFFERENCE

- UNIX is copyrighted name only few of the companies are allowed it i.e. IBM AIX and Sun Solaris and HP-UX. **UNIX is a Propreitary OS (Source code is not shared)**
- This quote from Official Linux kernel README file:
  Linux is a UNIX clone written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net. It aims towards POSIX compliance**.**
- **Linux is an open Source OS (Source code is shared)**
- For more differences , refer to the comparison chart (Linux and UNIX Comparison Chart.doc)

DIFFERENT VERSIONS OF RHEL

- Red Hat Enterprise Linux 3
- Red Hat Enterprise Linux 4
- Red Hat Enterprise Linux 5
- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 7
- Red Hat Enterprise Linux 8
- Red Hat Enterprise Linux 9

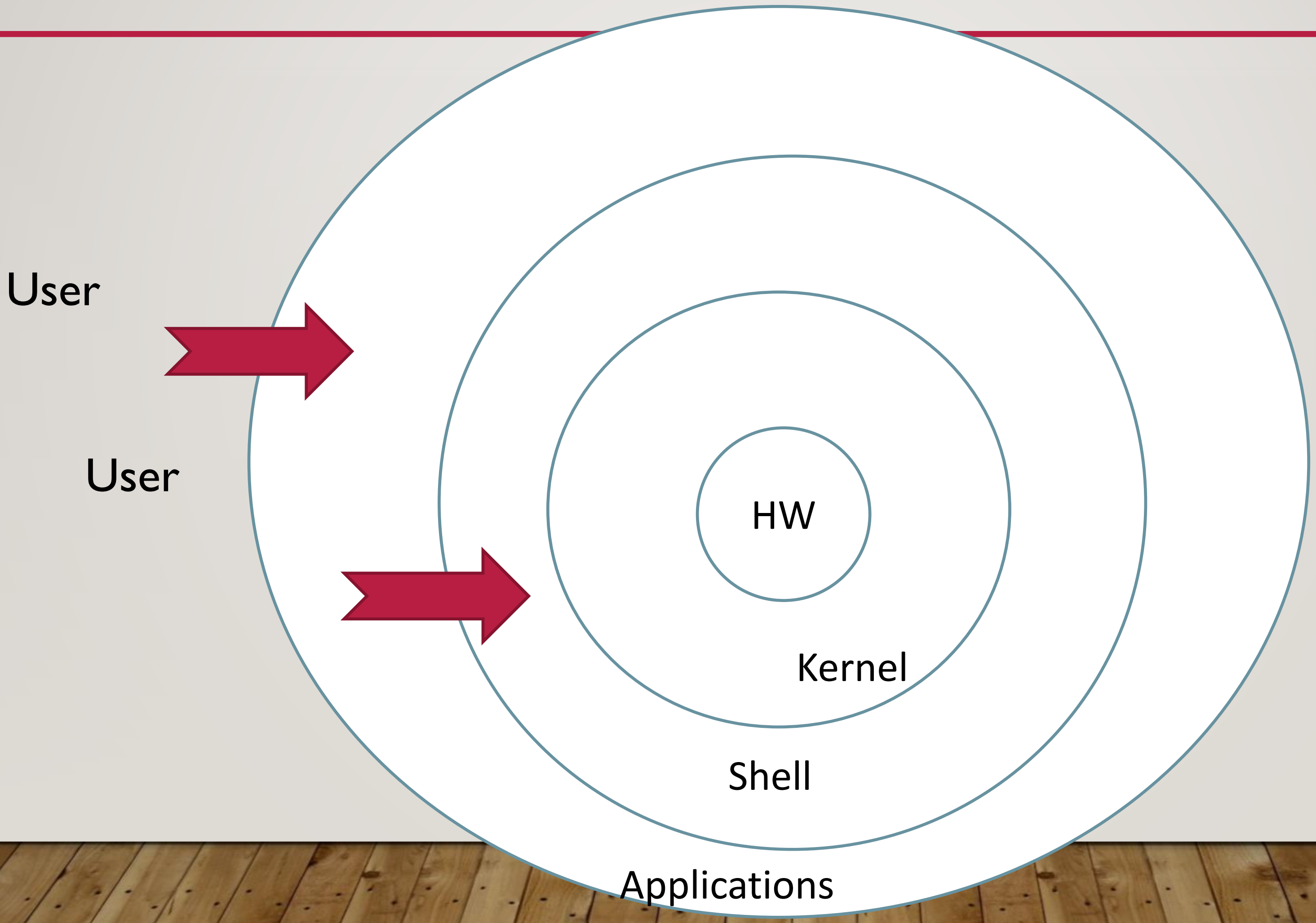
LINUX

- Linux is a :
- Multiuser
- Multitasking
- Stable and robust OS
- Secure OS
- Less Prone to virus attacks

LINUX

- Linux is a :
- Multiuser
- multitasking
- Stable and robust

LAYERED ARCHITECTURE OF LINUX OS



LINUX, UNIX 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

FILE SYSTEM HIERARCHY

- Everything is a file in Linux .
- The file system is hierarchical.
- Like any OS, the Linux File System is an

  Inverted tree structure

LINUX FILE HIERARCHY CONCEPTS

- Files and directories are organized into a single-rooted inverted tree structure
- Filesystem begins at the *root* directory, represented by a lone / (forward slash) character.
- Names are case-sensitive
- Paths are delimited by /

SOME IMPORTANT DIRECTORIES

- Home Directories: /root,/home/*username*
- User Executables: /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

LINUX ARCHITECTURE

- Monolithic kernel
  - Contains modular components, however
- UNIX-like or UNIX-based operating system
- Six primary subsystems:
  - Process management
  - Interprocess communication
  - Memory management
  - File system management
    - VFS: provides a single interface to multiple file systems
  - I/O management
  - Networking

RHEL7 INSTALLATION

HW Requirements for RHEL7 Installation

1) Processor = 64 bit
2) RAM = 4 GB or Higher
3) HDD = 10 GB or Higher
4) NW card
5) Keyboard, Monitor, Mouse etc standard Accessories

RHEL9 INSTALLATION

HW Requirements for RHEL 9 Installation

1) Processor = 64 bit
2) RAM = 4 GB or Higher
3) HDD = 30 GB or Higher
4) NW card
5) Keyboard, Monitor, Mouse etc standard Accessories

SW REQUIREMENT

- RHEL 9.2 server or higher iso image or DVD

INSTALLATION METHODS

- 1) Locally – From DVD ,Pen Drive or HDD
- 2) Over the Network – NFS, FTP, HTTP
- 3) Kickstart Installation
- 4) PXE Installation

- Can be installed in GUI or Text

PLANNING AN INSTALLATION

- Write down the make, model, and size of your hardware before you install.
- Red Hat 9.2 will automatically configure all supported hardware.
- Hardware compatibility list can be found at www.redhat.com.

DIFFERENT INSTALLATION METHODS

- **Local Installation :**

1. Install from local CD / DVD – Using RHEL CD / DVD in local drive.

2. Install from local Harddisk – One of the mount points of harddisk partitions may be the source of the ISO image of RHEL, from where the Operating system can be installed.

INSTALLATION METHODS (CONTINUED)

- Network Installation :

RHEL can be installed on multiple systems at the same time using Network installation through :

1. NFS
2. FTP
3. HTTP

INSTALLATION METHODS (CONTINUED)

- Kickstart Installation
- PXE Installation

KICKSTART INSTALLATION

- Kickstart Installation : Unattended or automated Installation
- To install on multiple systems simultaneously

with minimal user intervention.

PERFORM A KICKSTART INSTALLATION

- Kickstart installations can be performed using a local DVD, a local hard drive, or via NFS, FTP, or HTTP.
- To use kickstart, you must:

1. Create a kickstart file.
2. Create a boot media with the kickstart file or make the kickstart file available on the network.
3. Make the installation tree available.
4. Start the kickstart installation.

CREATING THE KICKSTART FILE

- The kickstart file is a simple text file, containing a list of items, each identified by a keyword. You can
- create it by using the **Kickstart Configurator application, or writing it from scratch. RHEL** installation program also creates a sample kickstart file based on the options that you

selected during installation. It is written to the file **/root/anaconda-ks.cfg. You should be able to**

edit it with any text editor or word processor that can save files as ASCII text.

SECTIONS IN KICKSTART FILE

- **Sections must be specified *in order.***
- *Items within the sections do not have to be in a specific order*
- unless otherwise specified.
- The section order is:
- Command section
- Partition section
- The %packages section
- The %pre and %post sections — These two sections can be in any order and are not required.

CLEAR PART IN KICKSTART FILE

- **--all —** Erases all partitions from the system.
- • **--drives= —** Specifies which drives to clear partitions from. For example, the following clears
- all the partitions on the first two drives on the primary IDE controller:
- **clearpart --drives=** hda,hdb --all
- • **--initlabel** — Initializes the disk label to the default for your architecture (for example
- msdos for x86). It is useful so that the installation program does not ask if it should initialize the disk label if installing to a brand new hard drive.
- • **--linux —** Erases all Linux partitions.
- • **--none (default)** — Do not remove any partitions.

LINUX KS OPTIONS AT BOOT: FOR KICKSTART INSTALLATION

- If the kickstart file is on a boot CD-ROM

linux ks=cdrom:/ks.cfg

- If kickstart file is on an nfs server

ks=nfs:<*server*>:/<*path*>

The installation program looks for the kickstart file on the NFS server <*server*>*, as file <path>.*

LINUX KS OPTIONS AT BOOT: FOR KICKSTART INSTALLATION (CONTD.)

- If the kickstart file is on a http server

ks=http://<*server*>/<*path*>

The installation program looks for the kickstart file on the HTTP server <*server*>*, as file path*>.

ks

- If ks is used alone, the installation program configures the Ethernet card to use DHCP**. The**
- kickstart file is read from the "bootServer" from the DHCP response as if it is an NFS server
- sharing the kickstart file

INSTALLING FROM NFS SERVER

- **nfs — Install from the NFS server specified.**
- • **--server=**
- Server from which to install (hostname or IP).
- • **--dir=**
- Directory containing the ***variant directory of the installation tree.***
- • **--opts=**
- Mount options to use for mounting the NFS export. (optional)
- For example:
- nfs --server=nfsserver.example.com --dir=/tmp/install-tree
- •

INSTALL FROM AN INSTALLATION TREE ON A REMOTE SERVER VIA FTP OR HTTP

- **url — Install from an installation tree on a remote server via FTP or HTTP.**
- Kickstart Options
- For example:

```
url --url http://<server>/<dir>
```

or:

```
url --url ftp://<username>:<password>@<server>/<dir>
```

OPTIONS AT THE BOOT: PROMPT DURING INSTALLATION

- To use a **specific display resolution**, enter **resolution=*setting as a boot option***. For example, to

set the display resolution to 1024×768, enter:

linux resolution=*1024x768*

- To run the installation process in

text mode, enter:

linux text

OPTIONS AT THE BOOT: PROMPT (CONTD.)

- You can install Red Hat Enterprise Linux with a newer version of the **anaconda installation program** than the one supplied on your installation media.

Enter The boot option

boot: linux updates

- To load the **anaconda updates from a network location instead, use:**

linux updates= followed by the URL for the location where the updates are stored.

OPTIONS AT THE BOOT: PROMPT (CONTD.)

- **Specifying the Installation Method**
- Use the **askmethod option to display additional menus that enable you to specify the installation** method and network settings.

boot: linux askmethod

- You may also configure the installation method and network settings at the boot: prompt itself as follows:
- **Installation method Option format**
- DVD drive **repo=cdrom:*device***
- Hard Drive **repo=hd:*device/path***
- HTTP Server **repo=http://*host/path***
- FTP Server **repo=ftp://*username:password@host/path***
- NFS Server **repo=nfs:*server:/path***

PARTITION CONSIDERATIONS

- **/ partition** – root partition is the starting point of entire directory structure of Linux. All the system and users directories are created under it.
- **swap partition** – Swap partition is used when there is not enough RAM and is used to support virtual memory.

- **boot partition** – This partition is used to keep the boot information. A disk space of 100-300 MB will be sufficient for this partition.

- The number can be increased as per the requirement. If the system is going to be a printer server or mail server then **/var** directory should have dedicated partition. For users home directory a separate partition **/home** partition can be created to implement disk quotas etc.

PARTITION CONSIDERATIONS(CONTINUED)

- / must contain /root,/dev/,/lib,/etc,/bin,/sbin

These should never be separate partitions

RHEL BOOTING PROCESS

1.BIOS

2.MBR (Master Boot Record)

3.GRUB2 Bootloader

4.Kernel

5.Systemd

6.Runlevel-Target

BIOS

- BIOS – Basic Input/Output System is a firmware interface that controls not only booting process and also provides all the control of low-level interface to attached peripheral devices.
- Performs POST and RAM Test
- Initializes the system Hardware components.
- After successful POST process, it will load the MBR (Master Boot Record) for the further boot process.

MBR

- Master boot Record placed in the first sector of the Linux boot Hard Drive and this information pre-loads into ROM (Read Only Memory) by BIOS.
- The MBR is only 512 bytes in size and it contains the machine code instructions for booting the Operating System, it’s called a boot loader, along with the partition table
- MBR loads and executes the GRUB2 bootloader

BOOT LOADER

- Bootloader: GRUB2 (Grand Unified Bootloader version2)
- GRUB2 is the default bootloader program in all latest version of like Red Hat/CentOS 7 and also Ubuntu from version 9.10.
- It has been replaced by GRUB bootloader also known as GRUB legacy.
- GRUB2 configuration file located in /boot/grub2/grub.cfg
- The boot loader (GRUB2 for RHEL 7) starts the RHEL 7 kernel and initial RAM disk(initrd)
- GRUB 2 is installed in the boot sector of your server’s hard drive and is configured to load a Linux kernel and the initramfs and the initrd is an initial root file system that will mount prior to the real root file system on Linux system.

KERNEL

- Linux Kernel is the central core of the OS and it is the first program loaded on the system starts up.
- While system starting kernel loads all the necessary Kernel Modules and Drives from initrd.img to load system first process systemd in Linux 7.

RUNLEVEL-TARGET

- systemd uses ‘targets’ instead of runlevels. By default, there are two main targets:
- multi-user.target: analogous to runlevel 3
- graphical.target: analogous to runlevel 5

RUN LEVELS : TARGETS IN RHEL7

- runlevel0.target -> poweroff.target
- runlevel1.target -> rescue.target
- runlevel2.target -> multi-user.target
- runlevel3.target -> multi-user.target
- runlevel4.target -> multi-user.target
- runlevel5.target -> graphical.target
- runlevel6.target -> reboot.target

LOGIN SCRIPTS

- Scripts executed when we login
- The /etc/profile File
- The ~/.bashrc
- The ~/.bash_profile

THE /ETC/PROFILE FILE

profile

- Stored in /etc/profile (global) and ~/.bash_profile (user)
- Run for login shells only

Used for

- Setting environment variables
- Running commands (eg mail-checker script)

BASHRC

- Stored in /etc/bashrc (global) and ~/.bashrc (user)
- Run for all shells
- Used for
- Setting local variables
- Defining aliases ,umasks, functions

LOGGING IN TO THE FILE SYSTEM

- Two types of login screens: virtual consoles (text-based) and graphical logins (called display managers)
- In RHEL7 , default GUI Terminal=Ctrl +Alt +F1
- In RHEL7, default CLI terminals =Ctrl+Alt+F2-F6
- Login using login name and password
- Each user has a home directory for personal file storage

LOGGING IN USING THE COMMAND LINE

- Ctrl +Alt +F2 = terminal 2

Upto

Ctrl+Alt+F6 =terminal 6 are all CLI terminals

You will get the following screen

Login:

Enter login name and you will be prompted for

Password: Enter your password

DESKTOP ENVIRONMENTS

- GNOME 3
- GNOME 3 is the default desktop environment on university Linux desktops. It is the most feature complete and useful desktop and as such as we highly recommend using it.

- GNOME 3, like many recent desktop environments, does not use the classic Windows style task bar model, or the Mac OS X style dock model. It can however be heavily customised by the GNOME Tweak Tool and GNOME Extensions.

COMMANDS FOR FILE OPERATIONS

ls	→ list files in a directory
touch	→ modify time stamps for file(s)
cat	**->** Concatenation (create and view files)
chmod	→ change permissions
cp	→ copy files
mv	→ rename files
ln	→ create links
rm	→ remove file
mkdir	→ make directory
rmdir	→ remove directory
umask	→ mask file permissions
find	→ find files matching the criteria

BASIC COMMAND SYNTAX IN LINUX

- Commands have the following syntax:
- **command *options arguments***
- Each item is separated by a space
- Options modify a command's behavior
- Single-letter options usually preceded by **-**
- Can be passed as **-a -b -c** or **-abc**
- Full-word options usually preceded by **--**
- Example: **--help**
- Arguments are filenames or other data needed by the command
- Multiple commands can be separated by **;**

CAT

- The cat command stands for concatenate, reads files either from standard input or those specified on command line and prints the contents on the screen.
- It can be used to
  - view contents of a file
  - concatenate contents of multiple files
  - create a file
- **cat** always sends its output to the standard output (by default console)

GETTING HELP IN LINUX

- Different Levels of help
- **whatis**
- ***command* --help**
- **man**
- **info**
- /usr/share/doc/
- Red Hat documentation

WHATIS COMMAND

- Displays short descriptions of commands
- Uses a database that is updated nightly
- Often not available immediately after install
- **$ whatis cal** cal (1) - displays a calendar

COMMAND **--HELP**

- Command **--help** Option
- Displays usage summary and argument list
- Used by most, but not all, commands
- **$ date --help**

MAN COMMAND

- Provides documentation for commands
- Almost every command has a man "page"
- Pages are grouped into "chapters"
- Collectively referred to as the Linux Manual
- **man [<chapter>] <command>**

NAVIGATING IN A MAN PAGE

- While viewing a man page
- Navigate with arrows, **PgUp**, **PgDn**
- ***/text*** searches for text
- **n**/**N** goes to next/previous match
- **q** quits
- Searching the Manual
- **man -k *keyword*** lists all matching pages
- Uses **whatis** database

THE **INFO** COMMAND

- Similar to **man**, but often more in-depth
- Run **info** without args to list all page
- **info** pages are structured like a web site
- Each page is divided into "nodes"
- Links to nodes are preceded by *
- **info [*command*]**

NAVIGATING IN INFO PAGES

- While viewing an **info** page
- Navigate with arrows, **PgUp**, **PgDn**
- **Tab** moves to next link
- **Enter** follows the selected link
- **n**/**p** /**u** goes to the next/previous/up-one node
- **s *text*** searches for text (default: last search)
- **q** quits **info**

THE /USR/SHARE/DOC DIRECTORY

- The /usr/share/doc directory contains

  extended information
- Subdirectories for most installed packages
- Location of docs that do not fit elsewhere
- Example configuration files
- HTML/PDF/PS documentation
- License details

RED HAT DOCUMENTATION

- Available on docs CD or Red Hat website

(www.redhat.com/docs)

- Installation Guide
- Deployment Guide
- Virtualization Guide

FILE AND DIRECTORY NAMES

- Names may be up to 255 characters
- All characters are valid, except the forward-slash
- It may be unwise to use certain special characters in file or directory names
- Some characters should be protected with quotes when referencing them
- Names are case-sensitive
- Example: MAIL, Mail, mail, and mAiL
- Again, possible, but may not be wise

ABSOLUTE AND RELATIVE PATHNAMES

- Absolute pathnames
- Begin with a forward slash
- Complete "road map" to file location
- Can be used anytime you wish to specify a file name
- Relative pathnames
- Do not begin with a slash
- Specify location relative to your current working directory
- Can be used as a shorter way to specify a file name

CURRENT WORKING DIRECTORY

- Each shell and system process has a *current working directory*(cwd)
- **pwd**
- Displays the absolute path to the shell's cwd

LS COMMAND

- **ls** command lets list files and directories

- Frequently used options

-l	list in long format
-a	list all file including those beginning with a dot
-i	list inode number of file in first column
-s	reports disk blocks occupied by file
-R	recursively list all sub directories
-F	mark type of each file
-C	display files in columns
-d	list directory entry, not its contents

It is possible to provide more than one option.

CREATE A FILE USING CAT

- # cat > newfile

This is a new file

Ctrl +d (to save and exit)

# cat newfile

To display the contents of the file you created

CAT COMMAND (CONTINUED)

Some of the commonly used options with cat

Command are:

-n	display line number
-b	display line number for non-blank lines only
-s	squeeze multiple adjacent blank lines

CP – FILE COPY

- **cp** command is used to copy files

```
$ cp    [options]  sourcefile    destfile
```

```
$ cp    [options]  sourcefile1  soucefile2  …   directory
```

Options can be

-i		→ interactive, warn before overwriting
-p		→ preserve mode
-R	or -r	→ recursively copy directories, if any

MV – MOVE OR RENAME FILES

- **mv** command is used to rename a file or move a file into another directory

```
        mv     [options]  sourcefile    destfile

    mv    [options]  sourcefile ...   directory
```

Commonly used options

-i → interactive, warn before overwriting

RM – REMOVE FILE

- **rm** command is used to remove or unlink a file

```
rm  [options]   file  ...
```

Options can be

-i		→ interactive
-f		→ don't prompt if file does not exist
-R	or –r	→ recursively remove contents of directories

DIRECTORY COMMANDS

- mkdir command creates a directory
- rmdir command removes directory

```
mkdir    [options]    directory_name

rmdir    [options]    directory_name
```

Commonly used options:-

-p	→ for mkdir, create parent directory, if it does not exist for rmdir, remove the directory along with subdirectories

SHELL METCHARACTERS

- File Globbing
- Globbing is wildcard expansion:
- ***** - matches zero or more characters
- **?** - matches any single character
- **[0-9]** - matches a range of numbers
- **[abc]** - matches any of the character in the list
- **[^abc]** - matches all except the characters in the list

DISK PARTITIONING

- Overview: Adding New Filesystems to the Filesystem Tree
- Identify Device
- Partition Device
- Make Filesystem
- Label Filesystem
- Create entry in /etc/fstab
- Mount New Filesystem

DEVICE RECOGNITION

- Master Boot Record (MBR) contains:
- Executable code to load operating system
- Space for partition table information, including:
- Partition id or type
- Starting cylinder for partition
- Number of cylinders for partition

DISK PARTITIONING

- An extended partition points to additional partition descriptors
- Total maximum number of partitions supported by the kernel:
- 63 for IDE drives
- 15 for SCSI drives
- Why partition drives?
- containment, performance, quotas, recovery

MANAGING PARTITIONS

- Create partitions using:
- **fdisk**
- **sfdisk**
- GNU **parted -** advanced partition manipulation (create, copy, resize, etc.)
- **partprobe -** reinitializes the kernel's in-memory version of the partition table

DIFFERENT FILE SYSTEMS IN LINUX

- ext2
- ext3
- ext4
- Xfs

Refer the doc difference between different Linux FileSystems

LINUX FILE SYSTEM

- Red Hat Linux File System
- ext2,ext3,ext4 are the different file systems

Supported in Linux

DIFFERENCE BETWEEN EXT2,EXT3,EXT4

ext2	ext3	ext4
Second extended fs	Third extended fs	Fourth extended fs
	Backward compatible with ext2	Default in RHEL6
	Default in RHEL 3,4,5	Backward compatible with ext3
	Ext3 = ext2 + journalling	Scalable compare to ext3
	Upto 32000 subdirs	Upto 64000 subdirs

FEATURES

- Journalling : File system metadata is backed up.
- Metadata= Information about the file like inode no., permissions etc.
- Scalable : Supports larger file size, larger partition size, We can create more no. of subdirectories etc.

MAKING FILESYSTEMS

- **mkfs**
- **mkfs.ext2**, **mkfs.ext3**, **mkfs.msdos**
- Specific filesystem utilities can be called directly
- **mke2fs [*options*] *device***

FILESYSTEM LABELS

- Alternate way to refer to devices
- Device independent
- **e2label *special_dev_file [fslabel]***
- **mount *[options]* LABEL=*fslabel mount_point***
- **blkid** can be used to see labels and filesystem type of all devices.

MOUNTING FILESYSTEMS WITH **MOUNT**

- **mount [options] *device mount_point***
- -t *vfstype* (normally not needed)
- -o options
- Default options: rw, suid, dev, exec, acl, and async

UNMOUNTING FILESYSTEMS

- **umount [*options*] *device* | *mount_point***
- You cannot unmount a filesystem that is in use
- Use **fuser** to check and/or kill processes
- Use the **remount** option to change a mounted filesystem's options atomically
- **mount -o remount,ro /data**

MOUNT POINTS AND /ETC/FSTAB

- Configuration of the filesystem hierarchy
- Used by **mount**, **fsck**, and other programs
- Maintains the hierarchy between system reboots
- May use filesystem volume labels in the device field
- The **mount -a** command can be used to mount all filesystems listed in the /etc/fstab

TO CONVERT A ROOT EXT2 FILE SYSTEM TO EXT3

1. Use the following command with the block device corresponding to the root file system:

```
# tune2fs -j device. ...
```

2. Run the mount command to determine the device that is currently mounted as the root file system

E2FSCK

- Unix-like operating system command
- Description: The system utility fsck is a tool for checking the consistency of a file system in Unix and Unix-like operating systems, such as Linux, macOS, and FreeBSD.
- A similar command, CHKDSK exists in Microsoft Windows
- Stands for: Filesystem check
- Function: Check and repair filesystems
- Syntax: e2fsck <options><device>

BASIC AND DYNAMIC DISKS

- The differences between basic and dynamic storage disks. ... Basic storage uses normal partition tables supported by all versions of Windows, MS-DOS®, and Windows NT. A disk initialized for basic storage is called a basic disk. It can hold primary partitions, extended partitions, and logical drives
- Dynamic disks have a lot of advantages over basic disks. They allow you to improve performance with a striped volume across multiple disks and give you the option to extend a volume to include unused space on other dynamic disks within the system
- Linux supports only basic disks

SEVEN FUNDAMENTAL FILE TYPES

- 1st bit of ls –l display
- -   Regular file
- d directory
- l symbolic link or soft link
- b block special file Ex: /dev/sda
- c character special file Ex: /dev/ttyS0,/dev/lp0
- p named pipe = used to pass data between processes
- s socket= End Point for communication

- file <filename> gives file type with more information

NFS

- NFS stands for Network File System, a file system developed by Sun Microsystems, Inc.
- It is a client/server system that allows users to access files across a network and treat them as if they resided in a local file directory.
- These differences are transparent to the NFS application, and thus, the user

DIFFERENCE BETWEEN SW RAID AND HW RAID

- Hardware RAID was the initial type of RAID available, where a specially built RAID controller handles the drives so that the processes are almost transparent to the host computer.
- Software RAID is a newer type of RAID where no specialized hardware is needed, and the host computer is responsible for the drives.
- SW RAID is implemented by the OS

DIFFERENCE BETWEEN SW RAID AND HW RAID CONTD

1. Unlike software RAID, Hardware RAID requires specialized hardware to handle the drives.

2. Software RAID is considerably cheaper than hardware RAID.

3. Unlike hardware RAID, Software RAID takes up a portion of the host processor.

4. Hardware RAID is more reliable compared to software RAID.

WHAT IS SOFTWARE RAID?

- Multiple disks grouped together into "arrays" to provide better performance, redundancy or both.
- **mdadm -** provides the administration interface to software RAID.
- "RAID Levels" supported in RHEL6 : RAID O, 1 , 5 and 6
- Spare disks add extra redundancy
- RAID devices are named, /dev/md0, /dev/md1, /dev/md2, /dev/md3 and so on.

SOFTWARE RAID CONFIGURATION

- Create and define RAID devices using **mdadm**
- **mdadm -C /dev/md0 -a yes -l 1 -n 2 -x 1 /dev/sda1 /dev/sdb1 /dev/sdc1**
- Format each RAID device with a filesystem
- **mke2fs -j /dev/md0**
- Test the RAID devices
- **mdadm** allows you to check the status of your RAID devices
- **mdadm --detail /dev/md0**

SOFTWARE RAID TESTING AND RECOVERY

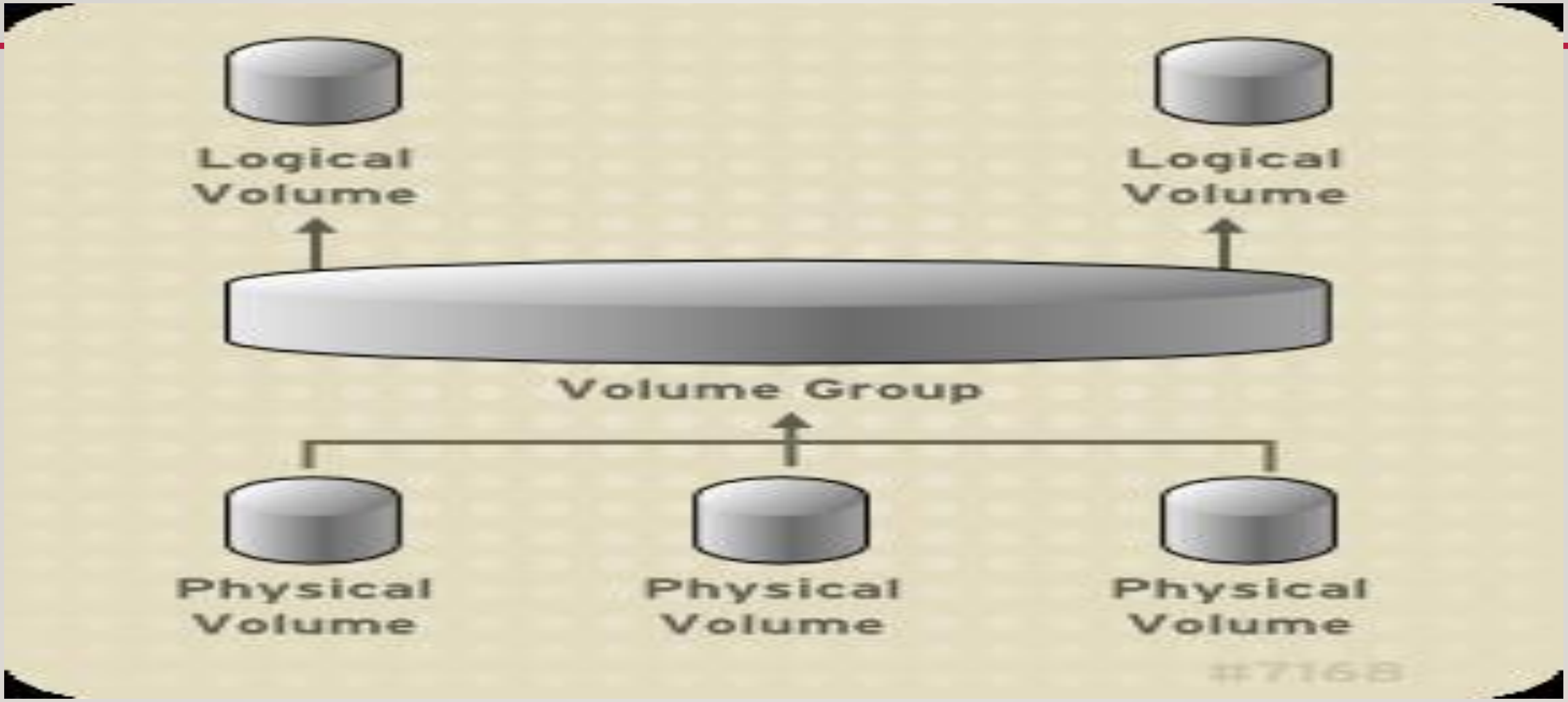
- Simulating disk failures
- **mdadm /dev/md0 -f /dev/sda1**
- Recovering from a software RAID disk failure
- replace the failed hard drive and power on
- reconstruct partitions on the replacement drive
- **mdadm /dev/md0 -a /dev/sda1**
- **mdadm**, /proc/mdstat, and syslog messages

MANAGE THE LOGICAL VOLUMES

Agenda:

- General LVM Concepts and Terms
- Displaying Current LVM Usage
- Initial LVM Deployment
- Extending a Logical Volume
- Removing a Physical Volume

FIG: LOGICAL VOLUME COMPONENTS



KEY POINTS

- Physical Storage (type 0x8e)
- Physical Volume (PV)= Disk or partition marked as usable space for the LVM
- Volume Group (VG)= Collection of PVs can think of it as a virtual disk drive
- Logical Volume (LV)=A piece of a VG, can be

Considered as a virtual partition

STEPS FOR LVM CREATION

1. Create physical partition(s)
2. Create Physical Volumes
3. Create Volume Group
4. Create Logical Volume(s)
5. Format the logical volume
6. Create a mount point
7. Mount the Logical Volume to this mount point

LAB STEPS

1. **Create new partition**

- Use disk utility create a new partition without a file system
- Edit the partition and change the type to

Linux LVM(0x8e)

2. **Initialize the new partition as a Physical**

Volume

Click on System -> Administration->Logical

Volume Management

LAB STEPS (CONTINUED)

3. Create a new VG using the PV just created
4. Create a LV within the new VG
5. Create a custom mount point
6. Format the LVM
7. Mount the LVM

EXTENDING THE LVM

Steps:

1. Create a new partition with id (0x8e)

without a file system.

2. Initialize the Partition as a physical volume.

3. Assign the new physical volume to the existing volume group

4. Extend the volume group

5. Extend the logical volume and its file system

By growing the logical volume

TASKS YOU WILL BE ABLE TO ACCOMPLISH

When you complete this unit you will be able to:

- Implement LVM Storage which is more

Flexible than standard partitions

- Extend the logical volumes and their file systems to provide additional space

PACKAGES & RPM PACKAGE MANAGER

Red hat Package Manager in short RPM is one of the most popular package management tool among Linux platforms.

rpm command is used to query, install, update and remove package.

In any rpm package name there are four parts of the name, those are package name, version, release and Architecture etc. sometimes, only the package name, version and release.

For example: samba-client-3.0.33-3.28.el5

Here, samba-client is the package name,
    3.0.33 is the version   and
    3.28.el5 is the release

File extension of rpm package is .rpm.

To install or update a package root privilege is required.

DIFFERENCE BETWEEN BINARY AND SOURCE RPMS

- **Binary** releases contain computer readable version **of** the application, meaning it is compiled.
- **Source** releases contain human readable version **of** the application, meaning it has to be compiled before it can be used.

PACKAGE ADMINISTRATION - RPM (CONTD.).

Installing a RPM package

```
#  rpm –ivh samba-client-3.0.33-3.28.el5.rpm
```

-i	stands for install
-v	stands for verbose
-h	stands for hash

The above command will install samba-client and while installing it will display # on progress.

Query on all the RPM Packages installed in the System.

```
# rpm –qa
desktop-backgrounds-basic-2.0-37
man-pages-2.39-15.el5
popt-1.10.2.3-18.el5
……….
………….
```

PACKAGE ADMINISTRATION - RPM (CONTD.).

<u>Query a Particular RPM Package with detailed information</u>

```
#rpm –qi samba-client-3.0.33-3.28.el5
```

-q   stands for query
-i   information

It will display every details and description of the package.

<u>Query of package through file</u>

```
#rpm –qf /usr/bin/mysqlaccess
```

It will display the name of the source package of this file.

PACKAGE ADMINISTRATION – RPM (CONTD.).

To display the manual pages of any package

```
#rpm –qdf  /usr/bin/mysqlaccess
```

-d stands for documents

To list all the files in a package

```
rpm –qlp  samba-client-3.0.33-3.28.el5.rpm
```

Here q stands for query, l stands for list and p stands for package

To view the dependency Package using rpm

```
# rpm –qpR  samba-client-3.0.33-3.28.el5.rpm
```

To Veryfy a particular rpm package

```
#rpm –Vp samba-client-3.0.33-3.28.el5.rpm
```

It compares the installed package with its source metadata and displays the mismatches (if any)

PACKAGE ADMINISTRATION – RPM (CONTD.).

To Upgrade a RPM package

```
# rpm –Uvh samba-client-3.0.33-3.28.el5.rpm
```

Erasing or removing RPM package

```
# rpm –ev samba-client
```

YUM - Yellowdog Update, Modified

- YUM (Yellodog Update,Modified) is software installation tool for Red hat linux and Fedora Linux. It is a complete software management system.
- The yum installer have been designed to use over network/internet. Duke University created the package manager YUM to improve the installation of Red hat packages.
- YUM alleviate dependency issues by searching many repositories for packages and their dependency packages and installing all of them together.
- YUM automatically computes dependencies and figures out what things should occur to install packages. It makes it easier to maintain groups of machines without having to manually update each one using rpm. Yum has a plugin interface for adding simple features

YUM CONFIGURATION

The configuration file /etc/yum.conf consists of:

- **main section –** Mandatory – This section can be used to configure yum options having global effects

- **One or more repository section[s] –** This section[s] used to set options which are repository-specific

➢ The recommendation is to define repositories in an existing or new **.repo** files in the directory **/etc/yum.repos.d**.

YUM CONFIGURATION (CONTD.).

Sample /etc/yum.conf file – main section

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit = 3
# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

YUM CONFIGURATION (CONTD.).

Sample /etc/yum.conf file – main section

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit = 3
# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

YUM CONFIGURATION (CONTD.).

Sample /etc/yum.conf file – repository section:

```
[repository]
 name=repository_name
baseurl=repository_url
```

YUM SERVER CONFIGURATION

Configuring the repository:

1. **Copy the required packages to /var/ftp/pub/Server**

```
# cp /root/Desktop/dumps/Server/zsh*  /var/ftp/pub/Server
```

2. Install createrepo package:

```
#rpm -ivh /root/Desktop/dumps/RHEL5/Server/createrepo*
```

3. Specifying the location of repository to yum

```
[root@linuxsrv1 ~]# cd /var/ftp/pub/Server
[root@linuxsrv1 Server]# createrepo -v .
```

YUM SERVER CONFIGURATION (CONTD.).

3. Create a file with repo as an extension and specify the YUM details in /etc/yum.repos.d directory – You may remove existing ones

```
[root@linuxsrv1 Server]# cd /etc/yum.repos.d
[root@linuxsrv1 yum.repos.d]# vi rhel.repo
        [rhelrepo]
        name=rhelrepo
        baseurl=file:///var/ftp/pub/Server
        gpgcheck=0
```

4. Clear the yum metadata.

```
[root@linuxsrv1] yum.repos.d]# yum clean all
```

5. Ensure that vsftpd service is up

PACKAGE ADMINISTRATION – YUM

Installing the packages:

```
[root@linuxsrv1~]# yum install zsh
```

See the information of the package:

```
[root@linuxsrv1 ~]# yum info zsh
```

Uninstalling the packages:

```
[root@linuxsrv1 ~]# yum remove zsh

[root@linuxsrv1 ~]# rpm -q zsh
package zsh is not installed
```

YUM CLIENT CONFIGURATION

1. **Remove/Move the existing repository file from /etc/yum.repos.d directory.**

```
[root@linuxsrv2 ~]# mkdir    /yum.repos.old
[root@linuxsrv2 ~]# mv /etc/yum.repos.d/*    /yum.repos.old
```

2. Create server.repo file in /etc/yum.repos.d directory

```
[root@linuxsrv2 ~]# cd /etc/yum.repos.d
[root@linuxsrv2 yum.repos.d]# vi rhel5_server.repo
[rhel5_server_repo]
name=rhel5_server_repository
baseurl=ftp://linuxsrv1/pub/Server
gpgcheck=0
```

YUM CLIENT CONFIGURATION (CONTD.).

3. Clear the yum metadata.

```
[root@linuxsrv2 yum.repos.d]# yum clean all
```

4. Install/get the info/uninstall zsh package from the server

```
[root@linuxsrv2 yum.repos.d]# yum install zsh
```

LINUX PATCHES

- Patches are updates that incorporate changes in source code.
- They can be applied to the Linux kernel or to applications and other systems code running on a Linux server.
- Patch management is basically the process of acquiring, testing and in- stalling multiple code changes (patches) to systems software and applications.

COMMON USES OF PATCHES

- Security Fixes
- Service Upgrades
- Bug Fixes
- Kernel Patches
- Kernel Upgrades

USER ADMINISTRATION

- User administration is a key function of the

  system administrator

- User administration requires a lot of planning

and preparation

- Users may need to login to systems in the

  same campus or to systems in their offices across the globe.

TYPES OF USER ACCOUNTS

1.Administrator or superuser

2. Local users and groups

3. Network users and groups

LOCAL USERS AND GROUPS

- Local users and groups are authenticated

  by the local system they login to.

- Types of Local users and groups

1. Root (Administrative account in Linux) ,uid=0
2. System Users and groups, uid=1-499 Ex: ftp, samba etc

(created by the system and when specific services are added)

3. Regular Users and groups, uid=500 onwards

Note: Human reference for user and group is

by name, Computer reference for user

and group is uid,gid.(Userid,Groupid)

FILES WHICH PROVIDE DEFAULT PROPERTIES FOR USER

1. /etc/login.defs

- Displays where your mailbox will reside /var/spool/mail/username
- Min pw age, Max pw age, pw warning (how many days before expiration)
- (Min and Max pw age are also in days)
- Create Home dir when user is created= yes or no
- Uid/gid for regular users etc.

2. /etc/default/useradd

- Default home dir, default shell,
- /etc/skel=dir which contains default profile, basrhc etc for users
- Account Expiration

Note: Account Expiration, pw min, max, warning days etc. will apply for all

users you create if updated in these files

We can use chage command to update these values for induvidual user.

DIFFERENT TYPES OF GROUPS

Groups are created to group users with similar

Rights and privileges into one group.

There are 2 types of groups :

1. Primary group = When user logins he is assigned to a group called primary group

All files and dirs, created by the user are owned

By the user and his primary group

2. Secondary group= For additional access

PRIVATE GROUP

- If username and primary group is same

  it is called a private group.
- If private groups are created then,

  files , dirs cannot be shared with other users
- Primary groups are recommended so that

  user can share files with members of his group (ex users of one dept are grouped into

One group)
- Private groups can be used with sensitive accounts

  Like MD,CEO or managers of the company.

CREATING USERS AND GROUPS USING COMMAND LINE

- Useradd or adduser <options> <username> creates an user, if no options are mentioned, user is created with default values for shell, homedir etc.

Options:

-g for primary group

-G for secondary group

-s for shell (bash,bourne,korn etc)

-d to specify home dir etc

For other options refer to the man pages

Note : 1. groups should exist before adding user to group

Groups are created with groupadd <groupname> command

2. Groups are not named primary or secondary

Only users are primary members or secondary members

3. Directory should exist before creating an user and assigning

him a home dir other than default

MODIFYING AN EXISTING USER, DELETING AN USER

- usermod [options] <username>

Options similar to that of useradd

- userdel <username> = deletes an user but

Home dir still exists for that user

userdel -r <username> deletes an user and

  user’s home dir

DISABLING AN USER

- Disabling an user in CLI :

Type * at the start of the password field, user’s account is disabled

or

usermod –s /sbin/nologin <username> disables

The user

- Disabling the user in GUI

system->administration->users and groups

Or expire the user account

ADDING, MODIFYING, DELETING GROUPS

- groupadd < groupname> creates a group
- groupmod <options><groupname> to

modify a group

- groupdel <groupname> deletes a group

EXAMPLE 1

You are the administrator of xyz company.

- You are asked to create groups prod, sales
- You are asked to create users ranjit, lokesh, imran with the following criteria:
- ranjit ,lokesh are primary members of prod,

  sec members of sales

imran belongs to his own private group and

Should get korn shell

SOLUTION TO EXAMPLE 1

```
# groupadd prod

#groupadd sales

# useradd –g prod –G sales ranjith

# useradd –g prod –G sales lokesh

#useradd –s /bin/ksh imran
```

ASSIGNMENT

- Create an user aravind with a comment team leader
- Create an user anil with account expiration

date of 29th February 2012.

- Create an user sam and assign him the home dir /xyz/sam.

FILES WHICH ARE UPDATED WHEN WE CREATE OR MODIFY AN USER OR GROUP

- /etc/passwd
- /etc/group
- /etc/shadow
- /etc/gshadow

/ETC/PASSWD

- /etc/passwd = User account database
- It has 7 fields
- Field 1 : Login name or account name
- Field 2: User passwd if x indicates it is

  written to shadow file for security
- Field 3 : uid
- Filed 4: User's primary group gid
- Field 5 : Comment or full name
- Field 6: Home dir of the user
- Field 7 : Default shell

/ETC/GROUP

- It is the group account database
- It has 4 fields :

  Field 1 : groupname

  Field 2 : group password

  Field 3 : gid

  Field 4: Additional or secondary members of the group

/ETC/SHADOW

- Contains encrypted user passwords

  It has 9 fields:

Field 1: username or login name

Field 2: Encrypted user passwords

- Last password change (lastchanged): Days since Jan 1, 1970 that password was last changed
- Minimum: The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password
- Maximum: The maximum number of days the password is valid (after that user is forced to change his/her password)
- Warn : The number of days before password is to expire that user is warned

/ETC/GSHADOW

- Contains secure group information
- It has 4 fields :
- Field 1 : groupname
- Field 2: encrypted Group passwd
- Field 3: (Comma seperated) group administrators
- Field 4: (comma seperated)group members

CREATING USERS AND GROUPS USING GUI

- 1. Click *System* then select *Administration* and click *Users and Groups*. This will launch the **User Manager** window.
- 2. Specify the password for root when asked.
- 3. In the **User Manager** window, click *Add User*. This will show the **Create New User** window.
- 4. In the **Create New User** window, fill in the *User Name*, *Full Name*, *Password* and *Confirm Password*. Click *Ok* when you’re done.
- 5. Done, you have created a new user.
- To remove a user, select the user you want to remove and click *Delete*.

BEST PRACTICES IN USER CREATION

- For Security, employee name and loginname should be different
- Local usernames and remote or network usernames

  should be different or else your logged using

Local account rather than network account

. Use lower case usernames instead of upper

Or mixed case user names to avoid login problems

etc.

PASSWORD MANAGEMENT

- Syntax: passwd < username > to change password for user (You will be asked to enter a password and

Confirm the password)

chage: the chage command lets you specify

the user account and password expiration

Syntax : chage [Options] <username>

Options :

-l list the current password expiration

-m <days> minimum days between pw cahnges

-M <days> maximum days user can use the pw

PASSWORD GUIDLINES

- As a general guideline, passwords should consist of 6 to 8 characters including one or more from each of following sets:
- Lower case alphabetics
- Upper case alphabetics
- Digits 0 thru 9
- Special character
- Passwords should be alpanumeric and should

  contain a special character
- Password should be a dictionary word
- Ex: a1b2c3_15,m$itd@123

ASSIGNMENT

- Create an user xyz assign him a password ‘ a’ as root login as that user and change his password
- Create an user andrew, he should be

able to login without password

SU COMMAND- SUBSTITUTE USER IDENTITY

- **Syntax:** su [*options*]... [*user* [*arg*]...]
- **Description:** The su Substitute User command allows you to become other existing users on the system.
- For example you can temporarily become root and execute commands as the super-user root.
- It is recommeded you limit the person allowed to su to the root account.

SU OPTIONS

- **Options**
- **-**, **-l**, **--login** Go through the entire login sequence (i.e., change to *user*'s environment).
- **-c** *command*, **--command**=*command* Execute *command* in the new shell and then exit immediately. If *command* is more than one word, it should be enclosed in quotes. For example:
- **Su -c ‘ls –l ‘ student where student is the username**

WHAT IS THE DIFFERENCE BETWEEN "SU -" AND "SU" ?

- The main difference between su - and su is that the former makes the shell a login shell.
- This is very important especially if the user is going to su from a regular user account to a root (superuser) account.
- Normal users do not usually have /sbin/ and /usr/sbin/ in their search path. Therefore if a normal user wants to execute the command ifconfig, for example, after doing su, he usually gets the error message:

```
bash: ifconfig: command not found
```

SUDO COMMAND

The sudo utility allows users defined in the /etc/sudoers configuration file to have temporary access to run commands they would not normally be able to due to file permission restrictions. The commands can be run as user "root" or as any other user defined in the /etc/sudoers configuration file. The privileged command you want to run must first begin with the word sudo followed by the command's regular syntax. When running the command with the sudo prefix,

SUDO COMMAND CONTINUED

- You may run other privileged commands using sudo within a five-minute period without being re-prompted for a password. All commands run as sudo are logged in the log file /var/log/messages. In order to use sudo we first need to configure the sudoers file.
- Note: With sudo we are configuring

co-administrators

OTHER REASONS FOR USING SUDO COMMANDS

- If a server needs to be administered by a number of people it is normally not a good idea for them all to use the root account. This is because it becomes difficult to determine exactly who did what, when and where if everyone logs in with the same credentials. The sudo utility was designed to overcome this difficulty.

SUDO EXAMPLES (EXAMPLE 2)

- **sudoers user can access the root and also can reset the password of root account.**
- **#useradd admin**
- #passwd admin
- #visudo add following entry,

admin ALL=(ALL) NOPASSWD:ALL

Note : We can directly edit /etc/sudoers or use visudo command

Recommended : use visudo as it displays the syntax errors when you save the file

EXAMPLE 2 :TO ALLOW AN USER TO START SERVICES ON THE SYSTEM

- Alias can be used to
  group no. of users , commands
- Ex:Visudo
- # User alias specification

```
User_Alias  ADMINS=student,new
 # Cmnd alias specification
Cmnd_Alias SERVICES  = /sbin/service,/sbin/chkconfig
 # User privilege specification
 admin ALL=SERVICES
```

ASSIGNMENT

- Create users student,new
- They should be able to start the services on your system
- For ex: service sshd restart

  chkconfig sshd on

WORKING WITH MULTIPLE GROUPS

- To add an user to mutiple groups say sales,

prod, hr

Ex: useradd –G sales, prod, hr hari

Where hari is username

Creates user hari with secondary membership of sales, prod,hr groups

Note: User can be a member of only one primary group and no. of secondary groups

SPECIAL PERMISSIONS

The three types of special permissions are:

1. Set User id
2. Set Group id
3. Sticky bit

1. SUID: SET USER ID

- Set for user(owner) so that all users can execute the
- Command as root.
- Syntax: chmod u+s <command> or
- chmod 4775 <command>
- Ex: passwd, ping.

2. SGID: SET GROUP ID

- Set for a dir for a group so that group owner of files created within the dir are owned by the groupowner
- Of the dir.
- Syntax: chmod g+s <dir> or chmod 2775 <dir>
- Collaborative dir: No. of users are made secondary
- Members of a group. And sgid is set for the dir so
- That these users can access and modify each others
- Files in that dir for a particular project.

3. STICKY BIT

- Set for a dir where many users have rw access so that only the owner or root can modify or delete
- Files within it.
- Syntax: chmod o+t < dirname> or chmod 1777 <dir>
- Ex: /tmp

ACL

- Control access to files with access control lists (ACL)
- · Manage file security using POSIX access control lists.

ACCESS CONTROL LISTS (ACLS)

- Grant rwx access to files and directories for multiple users or groups
- **mount -o acl /*directory***
- **getfacl *file|directory***
- **setfacl -m u:hari:rwx *file|directory***
- **setfacl -m g:prod:rw *file|directory***
- **setfacl -m d:u:harry:rw *directory***
- **setfacl -x u:sam *file|directory***

DEFAULT ACLS

- By setting a default ACL, you'll determine the permissions that will be set for all new items that are created in the directory. But the permissions of existing files and subdirectories remains same.

- To create a default FACL on a directory :

- # setfacl -m default:u:john:rw /accounts

Redirectors

- < used to redirect input from file
- > used to redirect output to file
- 2> used to redirect error
- & to redirect error and output
- 2>&1 to redirect error and output=used with pipes

Examples for redirection

•tr ‘a-z’ ‘A-Z’ < <filename>= redirects input
From file instead of from keyboard.
tr= translates lower case to upper case
        and vice versa

•find /etc –name hosts > x will find all files
With the name hosts under /etcand redirects to
the file x (Syntax for find: find <path> <criteria>)

Pipes

- Pipes (the **|** character) can connect commands:
- *command1* | *command2* Sends STDOUT of command1 to STDIN of command2 instead of the screen.
- STDERR is *not* forwarded across pipes
- Multiple pipes : Used to combine the functionality of multiple tools
- *command1* | *command2* | *command3*... etc

Examples of redirection (contd).

- find /etc –name hosts 2> y =redirects all errors while executing the command to the file Y
- find /etc –name &> z = will redirect both op and error to the file z.
- find /etc –name 2>&1 | more=gives both output and error from find command to the Command more

Filters

- In Unix and Unix-like operating systems, a **filter** is a program that gets most of its data from its standard input (the main input stream) and writes its main results to its standard output (the main output stream). Unix filters are often used as elements of pipelines. The pipe operator ("|") on a command line signifies that the main output of the command to the left is passed as main input to the command on the right.

Unix, Linux Filter commands

.**cat** = displays file content
.**cut** = used to cut a field or character from a file or output of command
.expand = expands spaces to tab spaces in a file
. **pr** =format before printing
• **grep** = searches for a string / pattern and displays the entire line
• **head** = display first ten lines of file by default
•**sed** = stream line editor= search and replace strings or patterns
•**sort** = sorts a file or command output in ascending or descending
•order
•**tail** =displays the last ten lines of a file

Linux, Unix filters continued

- tac =concatenate and print files in reverse
- tee = used with mulitiple pipes writes to file and pipes
- tr = translates upper case to lower case and vice versa
- uniq =report or omit repeated lines
- wc = word count, no.of lines, no. of words, byte count from file or output of command

Examples of filters in Linux, Unix

- cat hello.txt | wc | mail -s "The count" xyz@example.com
- head hello.txt
- tail hello.txt
- wc hello.txt
- uniq hello.txt
- sort hello.txt

Filters

Filter	What it does
cat	Concatenates and displays files
less	Displays a file screenful at a time
more	Displays a file screenful at a time
head	Prints the first ten lines of a file by default

Filters (Contd.)

- **tail Tail displays the last Ten lines of a file by default**
- grep Searches for a pattern and displays
- The line which contains the pattern
- sort Sorts files

VI EDITOR

- Create and Edit text files with vim
- · Introduce the vim text editor, with which you can open, edit, and save text files.

VI EDITOR

vi=visual editor ,A powerful editor

vim=visual improved editor

<u>Three modes in vi</u>

1) Command mode= Default mode =copy(yank),cut, paste, delete, change

3 MODES IN VI CONTD.

2) Insert mode = Insert the text

i=insert before cursor

I=insert before line

a=insert after cursor

A=insert after line

3) Ex mode = :w to save

:wq to save and quit

:q to quit without saving

:q! to abandon changes

:wq! to save and quit forcibly

CURSOR MOVEMENT IN VI

- h moves cursor to left by one character
- l moves cursor to right by one character
- j moves cursor down by one line
- k moves cursor up by one line
- w next word
- b previous word

Command mode in vi

- Command mode is the default mode in vi
- The moment you open the file with
 vi <filename> , you are in command mode
- In command mode is for :

i. Cursor movement
ii. Copy (yank)
iii. Paste
iv. Cut or delete
v. Go to etc.

Cursor movement in vi

- Cursor movement is possible only in command mode in vi
- Arrow keys may work
- Standard keys for cursor movement are:

i. **h** - for left

ii. **l** - for right

iii. **j** - for down

iv. **k** - for up

Cursor movement (continued)

- **w** - to move one word forward
- **b** - to move one word backward
- **$** - takes you to the end of line
- <**enter**> takes the cursor the the beginning of next line
- - (Minus) – takes the cursor to the beginning of the current line

Cursor movement continued

- **)** - moves cursor to the next sentence
- **}** - move the cursor to the beginning of next paragraph
- **(**   - moves the cursor backward to the beginning of the current sentence
- **{** - moves the cursor backward to the beginning of the current paragraph
- **%** - moves the cursor to the matching parentheses

Manipulating text in Command Mode

	Change (replace)	Delete (cut)	Yank(copy)
•Letter	**cl**	**dl**	**yl**
•Word	**cw**	**dw**	**yw**
• Line	**cc**	**dd**	**yy**
•Sentence after	**c)**	**d)**	**y)**
•Sentence before	**c(**	**d(**	**y(**
•Paragraph •above	**c{**	**d{**	**y{**
•Paragraph	**c}**	**d}**	**y}**

Put in cursor mode

- **p** - Paste the yanked lines from buffer to the line below
- **P** - Paste the yanked lines from buffer to the line above

(the paste commands will also work after the **dd** or **ndd** command (cut a line and paste or cut no. of lines and paste respectively)

SCREEN MOVEMENT IN VI

- **L go to last line of screen**
- **M go to middle line of screen**
- **H go to first line of screen**

FILE MOVEMENT IN VI

- G go to last line
- 1G go to first line
- 5G go to fifth line

BASIC SYSTEM CONFIGURATION TOOLS

date command in linux displays the current date and time

Syntax: date

Displays the current date and time

```
date -s  " Mar  4 00:06:10  2019“
```

Sets the date and time

VARIOUS DATE COMMAND FORMATS

Format options	Purpose of Option	Output
date +%D	Displays Current Date; shown in MM/DD/YY	02/07/13
date +%F	Displays Date; shown in YYYY-MM-DD	2013-02-07
date +%H	Displays hour in (00..23) format	23
date +%I	Displays hour (01..12) format	11

LP COMMAND

- The lp command is used to print files on Unix and Linux systems.
- The name "lp" stands for "line printer".

	Type This on System V UNIX	Type This on Linux or BSD UNIX
Print file	lp textfile	lpr textfile
Print file on a named printer	lp -dprinter textfile	lpr -P printer textfile
Cancel a print job	cancel requestid	lprm jobnumber
Check the printer queue	lpstat -a all	lpq -a

CONFIGURE NETWORK CONNECTIONS

- nmcli con show = Shows the N/w Cards in the system
- nmcli dev status = Shows status of N/w Cards in the system
- nmcli con up dev = To activate the device
- nmcli con del dev = To delete the device N/w card
- nmcli con add type ethernet autoconnect yes ifname ens33

= To add the device N/w card

NETWORK CONFIGURATION IN RHEL7

- ip a
- ip r

ip a is a shortcut for ip address show, ip r a shortcut for ip route show.

We can configure the Ip address to the NW card using nmtui command

We can configure the ipaddress with the following command:

```
nmcli con add con-name ethernet-ens33 ifname ens33 type ethernet ip4 192.168.1.20/24
```

MODIFYING THE IPV4 AND IPV6 ADDRESSES

- We can modify using nmtui command
- We can modify using nmcli command as follows:

- nmcli con mod ethernet-ens33 ipv4.address 192.168.1.25/24
- nmcli con mod ethernet-ens33 ipv6.address 2001:ac81::1105/64
- nmcli con down ethernet-ens33
- nmcli con up ethernet-ens33

IMPORTANT NETWORK CONFIGURATION FILES

- /etc/sysconfig/network-scripts/ifcfg-ethernet-ens33 = To configure NW address
- /etc/resolv.conf = Points to the DNS Server

MANAGING SYSTEM SERVICES

- systemctl start <servicename>
- systemctl stop <servicename>
- systemctl restart <servicename>
- systemctl enable <servicename>
- systemctl disable <servicename>

TCP WRAPPERS

- TCP Wrapper is an open source host-based ACL (Access Control List) system, which is used to restrict the TCP network services based on the hostname, IP address, network address, and so on.
- SSHD and VSFTPD are the tcp wrapper services

COMPARISON OF THE SERVICE UTILITY WITH SYSTEMCTL

service	systemctl	Description
service *name* start	systemctl start *name*.service	Starts a service.
service *name* stop	systemctl stop *name*.service	Stops a service.
service *name* restart	systemctl restart *name*.service	Restarts a service.
service *name*condrestart	systemctl try-restart *name*.service	Restarts a service only if it is running.
service *name* reload	systemctl reload *name*.service	Reloads configuration.
service *name* status	systemctl status *name*.service systemctl is-active *name*.service	Checks if a service is running.
service --status-all	systemctl list-units --type service --all	Displays the status of all services.

SYSTEMCTL

- systemctl list-units --type service
- By default, the systemctl list-units command displays only active units. If you want to list all loaded units regardless of their state, run this command with the --all or -a command line option:
- systemctl list-units --type service –all
- To list all units installed on the system,

systemctl list-unit-files

SYSTEMCTL COMMANDS

- systemctl is-active name.service
- systemctl is-enabled name.service

Displaying Service Status

systemctl status gdm.service

Displaying Services Ordered to Start Before a Service

- systemctl list-dependencies --after gdm.service

SECURE NETWORK SERVICES

- Activate and deactivate firewall
- Modify the firewall to allow access to trusted

Services

- Basic SELinux security concepts
- SELinux Modes
- Use the SELinux tool to change the SELinux modes
- Display SELinux concepts of processes and files

ACCESSING THE COMMAND LINE REMOTELY

- Remote Administration can be done on the servers using ssh and telnet

TELNET

- Terminal emulation
- Transient service controlled by superdaemon

xinetd

- Insecure
- Username, password is sent in plain text
- Main config file /etc/xinetd.d/telnet
- Uses port 23
- Cannot login as root by default in telnet

SSH

- Secure shell
- Standalone service
- Secure
- Username, password, data are sent through secure channel
- Main config file /etc/ssh/sshd_config
- By default we can login as root

INTERMEDIATE COMMAND LINE TOOLS

- Hardlinks , softlinks
- Compression tools =gzip, bzip2
- Backup tools=tar,dump

COMPRESSION TOOLS

- gzip <filename> compresses a file
- gunzip <filename> uncompresses a file
- Bzip2 compresses more than gzip
- bzip2<filename> to compress a file
- bunzip<filename>to uncompress a file
- Note: gzip and bzip2 can be used only for files

tar can be used for both file and dir names

In tar original file is not disturbed,in compression

Using gzip and bzip2 , original file is disturbed

COMBINING ARCHIVING AND COMPRESSION

- tar czvf <archive filename> <filename>

To compress and backup

C=create,v=verbose,f=filename,z=gzip compression

tar xzvf < archive filename> to restore and uncompress (x=extract)

tar cjvf <archive filename> <filename>

j=bzip2 compression

DUMP/RESTORE

- Back up and restore ext2/3 filesystems
- Does not work with other filesystems
- dump should only be used on unmounted filesystems or filesystems that are read-only.
- Can do full or incremental backups
- Syntax: dump [Options] <destination> <source>
- restore [Options] <backup>
- Examples:
- **dump -0u -f /dev/nst0 /dev/hda2**
- **restore -rf /dev/nst0**

SEVEN FUNDAMENTAL FILE TYPES

- **ls –l display**
- Symbol File Type
- - regular file
- d directory
- l symbolic link
- b block special file
- c character special file
- p named pipe
- s socket

CHECKING FREE SPACE

- **df** - Reports disk space usage
- Reports total kilobytes, kilobytes used, kilobytes free *per file system*
- **-h** and **-H** display sizes in easier to read units
- **du** - Reports disk space usage
- Reports kilobytes used *per directory*
- Includes subtotals for each subdirectory
- **-s** option only reports single directory summary
- Also takes **-h** and **-H** options
- Applications->System Tools->Disk Usage Analyzer or **baobab** - Reports disk space usage graphically

NETWORK CONFIGURATION AND TROUBLESHOOTING

Objective:

- Configure network settings
- troubleshoot network issues

DEVICE CONFIGURATION – NETWORK INTERFACE

- While booting the machine a script called **/etc/rc.d/init.d/network** is started.
- It initiates the network interface devices.
- It is done by invoking a set of scripts stored in **/etc/sysconfig/network-scripts** directory.
- Interface configuration file : **/etc/sysconfig/network-scripts /ifcfg-eth0**

DEVICE CONFIGURATION – NETWORK INTERFACE (CONTD.).

/etc/sysconfig/network Script stores Global network configuration information for each interface. A typical /etc/sysconfig/network looks like:

```
NETWORKING=yes
FORWARD_IPV4=‘yes’
HOSTNAME=‘xyz.testdomain.com’
GATEWAY=‘192.165.1.1’
GATEWAYDEV=‘eth2’
```

DEVICE CONFIGURATION – NETWORK INTERFACE (CONTD.)

Virtual IP Address

virtual IP address (**VIP** or **VIPA**) is an IP address that is not connected to a specific computer or network interface card (NIC) on a computer. Incoming packets are sent to the VIP address, but they are redirected to physical network interfaces.

VIPs are mostly used for connection redundancy; a VIP address may still be available if a computer or NIC fails, because an alternative computer or NIC replies to connections.

A virtual IP address eliminates a host's dependency upon individual network interfaces.

DEVICE CONFIGURATION – NETWORK INTERFACE (CONTD.).

Configuring Virtual IP address

Using the following command, we can configure Virtual IPs in the Linux box at run-time without re-booting the machine or re-starting the network.

```
ifconfig   eth0:1  172.19.1.5  netmask   255.255.255.0   up
ifconfig  eth0:3  172.19.1.6  netmask   255.255.255.0   up
```

The same interfaces can be made down by using the following command.

```
ifconfig   eth0:1  172.19.1.5  netmask   255.255.255.0   up
ifconfig  eth0:3  172.19.1.6   netmask   255.255.255.0   up
```