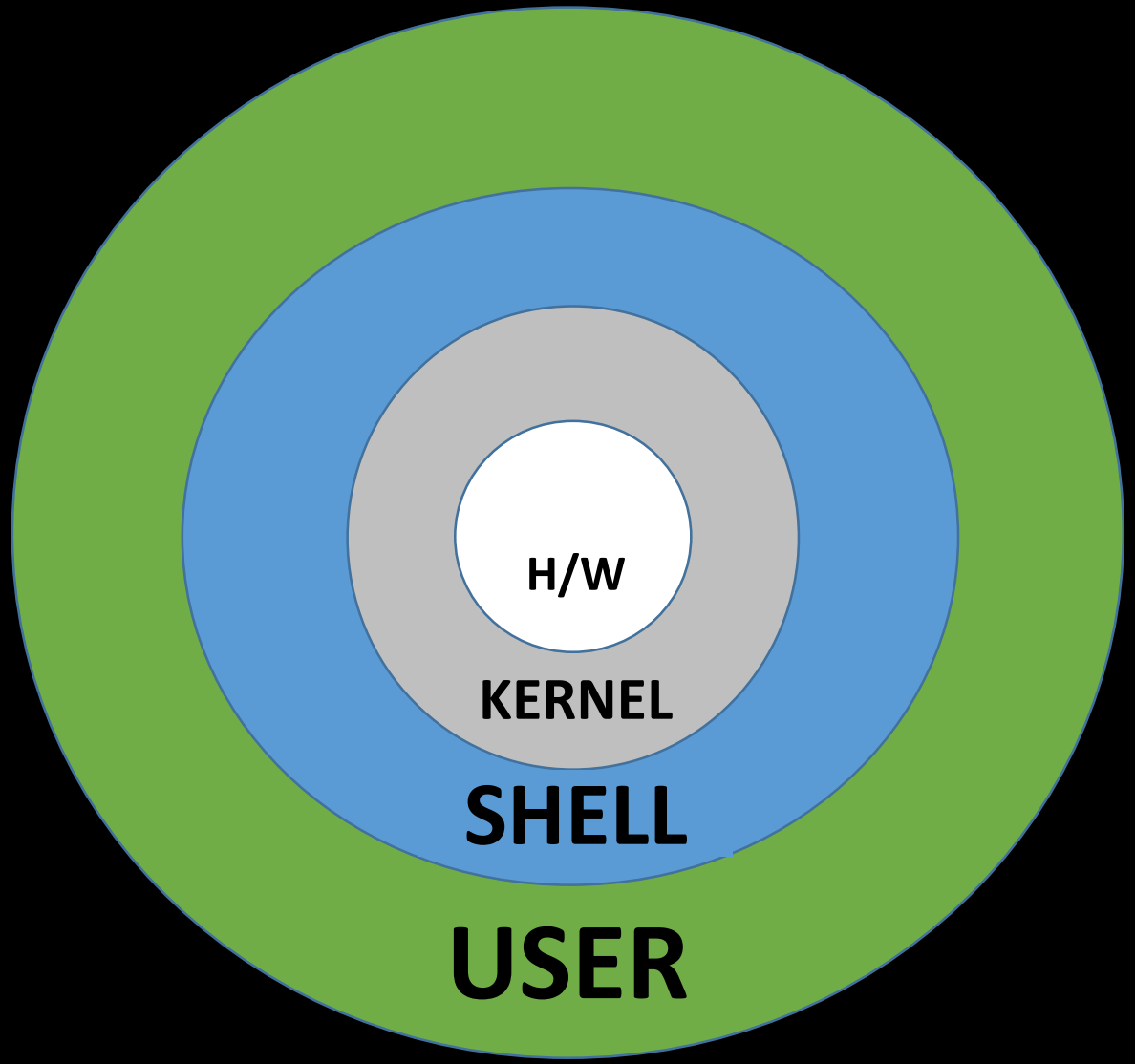
STRUCTURE OF OS



Introduction of Linux

- Linux was developed by Mr. Linus Torvalds in 17th Sep. 1991. But Red Hat launch first OS of Linux in 1993.
- Linux is a open source OS.
- Linux is a dedicated network OS.
- Dedication is Linux use in server end & Windows use in front end.
- Linux was developed under GPL(General Public License).
- Linux is a case sensitive OS.
- Linux use its own KERNEL & kernel is core part of OS.
- Linux use a shell name “BASH” & it’s a default shell of RHEL-6.
- SHELL- Shell is a platform which is used by the user to providing instructions to the hardware via kernel.
- BASH- Bourne Again Shell.

- To install any OS, It required a installer. Linux use a installer named “ANACONDA”.
- To boot any system, It required a boot loader. And RHEL-6 use a boot loader named “GRUB(Grand Unified Boot loader)”.
- Linux is multiuser multitasking OS.
- Linux operate in CLI (Command Line Interface) and GUI (Graphical User Interface) both.
- In Linux GUI is called X-Windows.

- X-Windows has two types-

a) GNOME (Global Network Object Model Environment)

b) KDE (Kool Desktop Environment)

- KDE: The KDE GUI is equipped with everything users typically need, including a file manager, window manager, help tool and system configuration tool.
- In Linux everything is file.
- Whenever we create a file in Linux, system provide a unique number to each & every file for system identification is called “I-nod number” or “Index number”.
- The super user of Linux is “Root”.
- The parent directory of Linux is “/” & directly or indirectly all the files & folders are related with “/”.
- Linux was developed by using two programs: a) C Language

  b) Unix script

INSTALATION OF REDHAT

Recommendations :

- Recommended space for Red Hat is 10 GB.
- “/” is root partition.
- “/boot” is second partition (100-200 MB required).
- “swap” is third partition (swap= 2 x RAM).
- Ext3 or Ext4 is format in RHEL-6.

TO CREATE PARTITIONS

- Go to create -> Standard Partition
- Click on “Create”
- In mount select “/”. File system ext4
- Size & Create.
- * If we want to check memory info press: Alt+ctrl+F2, And type:

cat /proc/meminfo :more

- Come back to GUI mode: Alt+ctrl+F6
- Create SWAP partition: If u have 3GB RAM, than give size 6GB (2 x RAM).
- DONE !!!!!

EXTRA

- **NOTE** : If hdd is SATA or SCSI then it gives partition in “sda”, And if hdd is IDE then it gives in “hda”.
- **File system** : Mechanism to how to store in hdd.
- **Virtual Memory** : A part of hard disk act as a RAM is called Virtual Memory.
- For basic install packages:

 * If you need graphics, click on “Desktop”.

 * Click on base “Customize”.

Basic Commands

1. Pwd : Present working directory.
2. Cd / : Change directory.
3. Clear : clear screen or Ctrl+L .
4. ls : List of the directory contain.

5. mkdir name : make a directory.
6. touch filename : To create a blank file.
7. touch /filename : Specific directory.
8. cat> filename : To create a file with text, Ctrl+D to save file.
9. cat filename : To show a file with text.
10. cat >> filename : To append the data in existing file.
11. eject : To eject the CD drive.

13. eject –t : Take in back CD drive.
14. date : Show the date & time.
15. cal : Show the calendar (cal 2013 & cal 11 2013)
16. Copy:

- File to File : cp sourcefile destinationfile
- File to Folder : cp –rf file1 destinationfolder
- Folder to Folder : cp –rf sourcefolder destinationfolder

17. Move:

- File to File : mv sourcefile desti.file
- Folder to Folder : mv sourcefolder desti.folder

18. Remove :

❖#rm filename

NOTE: For forcefully delete the file : #rm –f filename.

❖#rm –rf foldername

19. Rename :

#rename oldname newname oldname

20. whoami : To show the user login.

21. whatis command : one line information of command.

22. info command : Detail info of cmd [Press ‘q’ foe exit].

23. man command : Complete info of cmd.

ROOT STRUCTURE HIERARCHY

1. /bin : It stands for binary & this Linux directory contain the information of only those commands which is executable by normal user .
2. /boot : Its separate partition which contain the information about booting configuration files.
3. /dev : It stands for ‘devices’ and it contain the information of all system devices.
4. /etc : This Linux directory contain all the system services & system configurations related file.
5. /home : This is home directory of normal user.
6. /lib : It stands for library & this Linux directory contain all the program information in Linux.
7. Lost+found : It contain the all swappable data.
8. /media, /mnt, /opt, /misc : All these Linux directories are by default created by Linux but they are empty because all these directory are provided to user to store these customize data.

9. /net : This Linux directory is used to access the share data on network.

10. /proc : This Linux directory contain all the system information including process information.

11. /root : This is the home directory of super user.

12. /sbin : It stands super user binary & this Linux directory contain the information of only these commands which is executable by only superuser.

13. /usr : This is the most critical directory in Linux which contain all the system commands.

14. /selinux : It stands for security enhanced Linux. It’s a security feature of Linux. Which provide the file base security and all the rules & policy of a selinux is contain by the ‘/selinux’ directory.

15. /opt : Contain all installation & download (It is separate directory).

16. /srv : It stands for services and this Linux directory contain the information of third party services.

17. /sys : It stands for system & this Linux directory contain all the system driver database.

18. /tmp : It stands for temporary & this Linux directory contain the information of temporary files.

19. /var : It stands for variable & this Linux directory contain all the Linux variable data.

TERMINAL

[root@station1 ~]#

User name

Hostname

root home directory

AND

/sbin & /bin

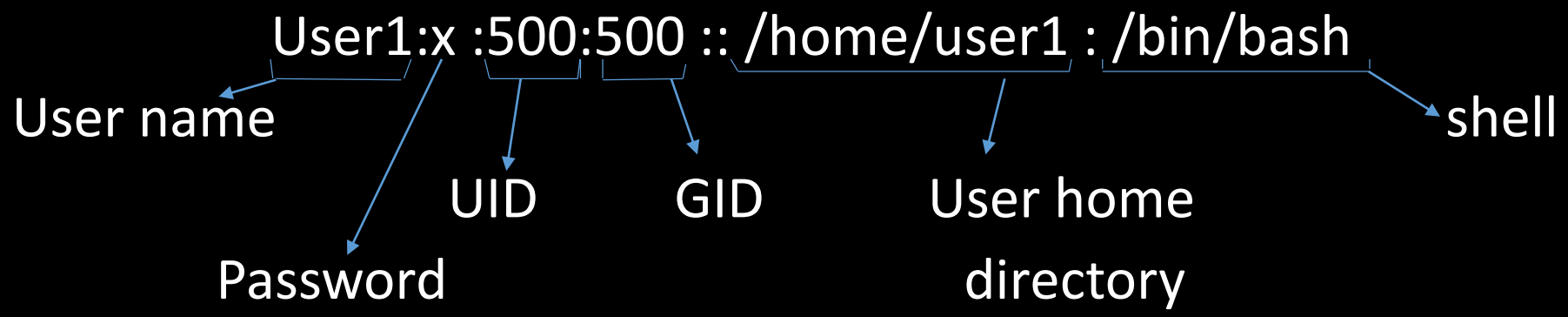
/usr (This is shortcut of /sbin & /bin)

Basic User & Group Management

USER :

1. User is an smallest object of any OS, which is created for login purpose.
2. Whenever we create a user in Linux, it provide an unique ID to each & every user known UID.
3. Whenever we create a user in Linux, our system create a group having the same name of that user & that user is primary member of that group.
4. Whenever we create a user in Linux, It required a password & password information stored into '/etc/shadow'.

FORMATE :

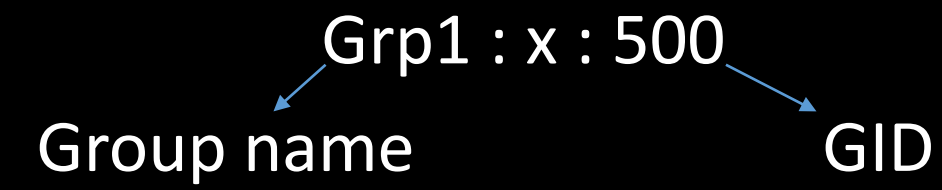


5. Whenever we create a user in Linux our system stored its information into ‘/etc/passwd’.

GROUP :

1. Group is a container object which contain the multiple users & it’s used to assign the permission to the multiple users by using single object.
2. Whenever we create group in Linux our system provide a unique ID to each & every group known as ‘GID’.
3. Whenever we create a group in Linux our system never create a user automatically.
4. Whenever we create a group in Linux it stored the group information into ‘/etc/group’.

FORMATE :



LAB

- **Create user** :

#useradd username

#adduser username

Password:

#passwd username

User info :

#cat /etc/passwd

Password info :

#cat /etc/shadow

Create group :

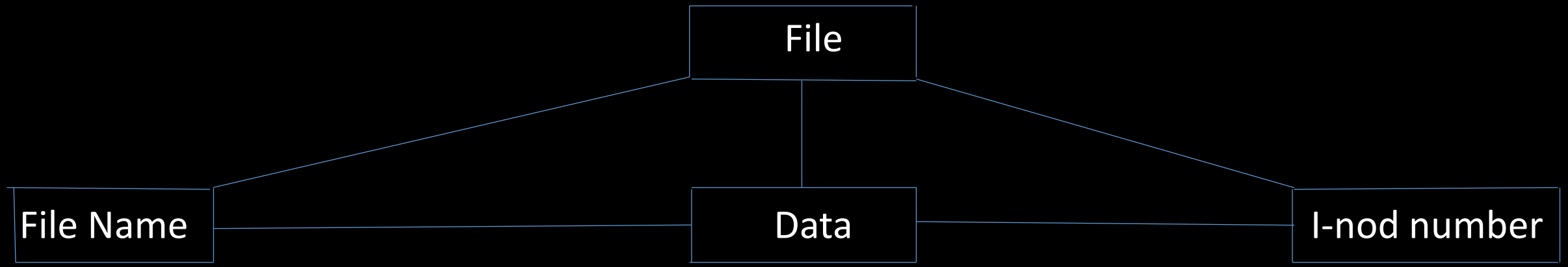
#groupadd groupname

Group info :

#cat /etc/group

File Types

- File : File is a collection of data & it combination of filename & I-nod number.



- Types :
  - Regular
  - Directory
  - Character

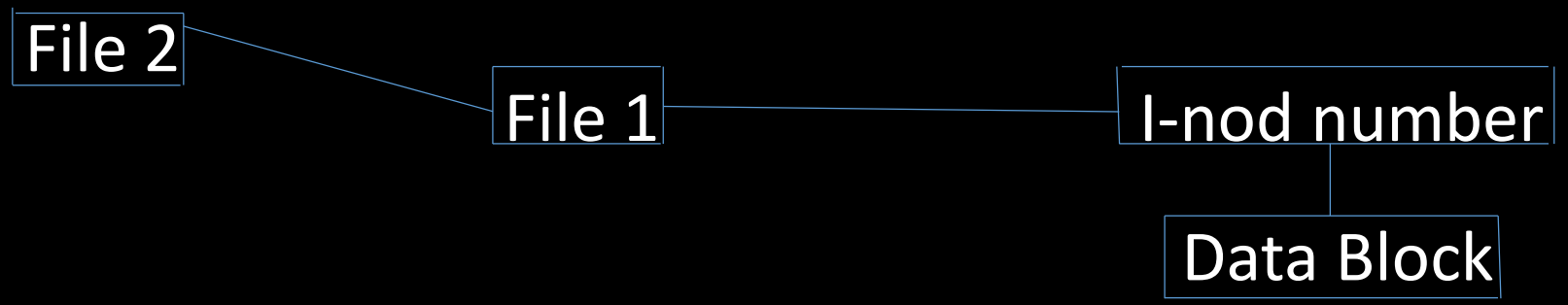
- Block
- Pipe or Socket
- Links

- File identification :
- \- => Regular
- d => Directory
- c => Character
- b => Block
- pors => Pipe or Socket
- l => Link

Link File

- Two types of Link :-

A. Soft Link : Soft link is also known as symbolic link which is work as a shortcut in Linux. Whenever we create a soft link of any file, our system create a link with file name.



B. Hard Link : It's a special type of link available in Linux. Which is used for data redundancy purpose. Whenever we create a hard link our system create a link with I-nod number instead of filename.
* When hard link is create, then all link have same I-nod number.
#cat >hcl (create file)

#ln hcl ibm

LAB

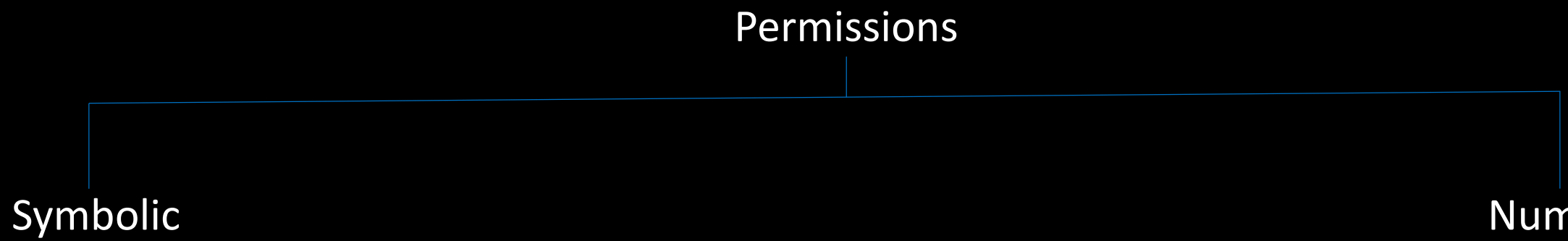
- After create a file :
  #ls –li

  12100 – rw-r- -r- -1 root root

  I-nod number

  File type

Permissions



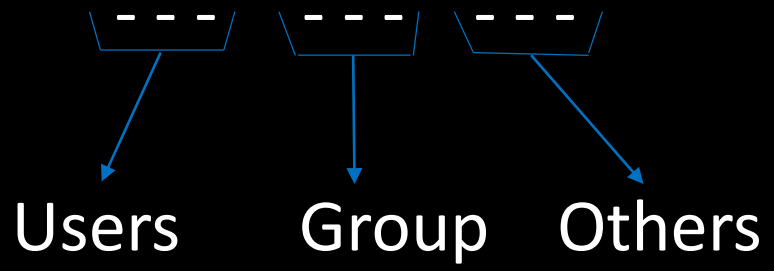
SYMBOLIC :

u = Users
g = Groups
o = others

r = Read
w = Write
x = Execute

+ = Assign permission
- = Remove permission

- Permission Area :



- Default permission of file in Linux :

- Read / Write -> User (6)
- Read / Read -> Group (4)
- Read -> Other (4)

- Default permission for folder in Linux :

- Read / Write / Execute -> User (7)
- Read / Execute -> Group (5)
- Read / Execute -> Other (5)

LAB

To give permission to file :

#chmod u+rwx ram

#ls –li

Remove permission :

chmod u-x ram

Give permission to U,G&O :

#chmod u+rw,g+rx,o-x ram

- Numeric Method :

Read (r) => 4

Write (w) => 2

Execute (x) => 1

Full permission => 7

- Give permission ->

#chmod 721 ram

** I-nod contain only 2Kb.

Shell Types

1. Sh
2. Ksh
3. Csh
4. Tcsh
5. BASH
6. Nologin

1. Sh : It stands for shell & it's a traditional shell of Linux, which is use to execute the commands or give the instruction to the hardware.
2. Ksh : It stands for 'Korn shell'. Which is a dedicated shell of UNIX. Which is used for executing the UNIX script only.
3. Csh : It stands for C language shell, which is use to execute normal C programming.

4. Tcsh : It stands for “Turbo C shell”. Which is use to execute advance C programming.

5. BASH : It stands for “Bourne Again Shell”. It’s a default shell of RHEL-6. It’s made by the combination of ‘ksh’ & ‘tcsh’. So by using BASH, we can execute any kind of program or commands.

6. Nologin shell : It’s used by the system for allowing or disallowing the user to login.

VIM Editor

- VIM editor : Editor is a tool available in all OS, Which is used for modifying the text of the file.
- Mode of VIM editor : Three types of modes available in VIM editor.

i. Command Line Mode : It is a default mode of VIM editor. Which is used for executing the all commands of VIM editor. In VIM editor all commands execute in command mode but effect display on insert mode.

ii. Insert mode : In this mode we type all the text of the file.

iii. Last line mode : In this mode we perform all the last action like- save, quit, save & quit etc….

- i : To go to the insert mode.
- Esc : Return to the cmd mode.
- a : Jump to the next character in insert mode.
- o : Jump to the next line in insert mode
- O : Go to the insert mode by adjusting on the current line into next line.
- x : Delete the character.
- A : Go to the end of the line in insert mode.
- I : Go to the start of the line in insert mode.
- j : for down arrow.
- k : for up arrow.
- l :for right arrow.
- h : for left arrow.

- yy : for copy (yanked) the current line.
- nyy : for copy ‘n’ number of lines. (n= 1,2,3,4,….,n)
- dd : for delete or cut the line.
- ndd : for delete or cut ‘n’ lines.
- p : for paste.
- /word : for find the word.
- :%s/original word/new word : replace word
- :set nu : for set line number.
- :set nonu : for remove the line number.
- :line number : to reach the line number.
- :w : for save.
- :q : for quit.
- :wq : save & quit
- :wq! : save & quit forcefully.

Job Scheduling

- Whenever we want to execute any command not yet but in future is known as Job scheduling.
- And to perform job scheduling in Linux we must use a service of RHEL-6 known as ‘crond’. To use this service, we use a utility ‘crontab’.
- How to schedule a Job :

a. Whenever we want to schedule a job, we must specify six different things, which are given an below:

* Minute (0-59)
*hour (0-23)
*date (1-31)
*month (1-12)
*day (0-6)
*Full path of command

LAB

1. Identify console :
   #tty
2. Check time :
   #date
3. #crontab –e
4. 50 10 16 10 02 mkdir /JOB>/dev/pts/0
5. :wq
6. #service crond restart
7. #chkconfig crond on
8. #crontab -l

Backup & Restore

- Backup : Backup is a duplicate copy of original data which is used in case of data lost.
- Restore : To extract original data from backed up data, is called Restore.

LAB

- To create data :

```
#mkdir /data
#cd /data
#mkdir ram
#cat >iiht
#touch baba
```

- To backup of /data :

  #tar –cvf /data.tar /data
- Check backup :

  #tar –tvf data.tar
- Original data delete & restore :

  #tar –xvf /data.tar

How to search data in Linux

Syntax :

#find / -name passwd

#find / -size 2k

#find / -perm 755

#find / -inum 12345

Run Levels

- **Run Level** : Run levels are the specific running mode & working criteria available in Linux. Each run level define how our system will work & all the run levels control by an ‘initialization script’ known as ‘init’.
- Available run levels of RHEL-6 :
  A. init 0 – System shutdown mode.
  B. init 1 – Single user mode or troubleshooting mode.
  C. init 2 – Multiuser with CLI.
  D. init 3 – Multiuser with CLI with Networking.
  E. init 4 – Available for developers.
  F. init 5 – Multiuser with graphic with networking.
  G. init 6 – System restart mode.

- To change run levels : [Carefully] [Its work on booting time.]
  #vim /etc/inittab
  you see:

  id:5:initdefault

  change it.

Recover root Password

- To recover your root password, Reboot your system in single user mode.
- For rebooting in ‘init 1’ restart your system and go to the ‘grub screen’.
- For grub screen we press arrow key continuously when system is restart.
- After going to grub screen, select your Red Hat OS & press ‘e’ then select your kernel version & press ‘e’.
- Then press ‘spacebar’ & type ‘1’ then press ‘enter’ , Then press ‘b’.
- After performing whole these task, your system automatically start in Single User Mode.

- When you got the shell, type these commands:
  #setenforce 0
  #passwd –d root
  #passwd root
- Give the root password & reboot your system.
- Now, try to login with your new password & you are able to login.

How to use Removable media

- To mount removable device :
  #mount /dev/dvd /mnt
  #cd /mnt
  #ls
- To mount pen drive :
  *Check media (device)name
  #fdisk -l
  got : /dev/sdb1

  #mount /dev/sdb1 /media name

Advance User & Group management

- **Customize user properties** :
  *If you create a new user with fix UID :-
  #useradd -u 666 user2

  *Exciting user to modify UID :-
  #usermod -u 599 user1
- **Customize group properties** :
  *Existing group change GID :-
  #groupadd -g 599 group1
- **Create a user with comment** :
  *On user1:-
  #usermod -c “NetworkAdmin” user1

* To check :-
#cat /etc/passwd
*To switch user:-
#su – username

- To check hidden files in Linux :
#ls -a
*To hidden file in Linux put ‘.’ in starting of the file name.
- To change shell :
* Check user shell name:-
#cat /etc/passwd
*Change shell:-
#usermod -s /bin/nologin user1
#cat /etc/passwd

User Membership :

a) Primary member :-

#usermod –g groupname username

b) Secondary member :-

#usermod -G groupname username

Advance Permission

1. Set UID = 4
2. Set GID = 2
3. Sticky bit = 1

Set UID (Script): It is a special permission in Linux, Which is only applicable in system script. Which is by default use by the system. If any script having the set UID permission, then the system will check for the UID for user, who is executing the script.

If the user having the UID b/w 0 to 499 then the system will execute with full function otherwise limited function.

Set GID (Folder) : There is a special permission available in Linux. Which is only applicable on the folder. Which is use to inherit the group ownership from parent object to its child object.

Sticky bit: It is also a special permission available in Linux, Which is use to prevent the user to delete the file of other users.

Only owner of that file having rights to delete there files.

- CHANGE OWNER :

#chown ram admin

User — Make owner of admin

#chgrp grp1 admin

Group — Admin ownership

- Check: #ls –l /usr/bin/passwd

  got: -rwsr-xr-x. 1 root root ……..

  Permission

- Symbolic: #chmod u+s /usr/bin/passwd

- Numeric: #chmod 4755 /usr/bin/passwd

- Set GID: Inherit group ownership

  Symbolic: #chmod g+s /admin/

  Numeric: #chmod 2770 /admin/

- Restriction for folder :

  #chmod o+t /admin/
  #chmod 1770 /admin/

got permission as “T” , Sticky bit is apply.

LAB

```
#mkdir  /admin
#ls  –li  /admin
#groupadd  grp1
#chgrp  grp1  /admin
#ls  –li  /admin
#chmod  770  /admin
#ls  -li
#useradd  -G  grp1  u1
#useradd  -G  grp1  u2
```

```
#cat  /etc/group
#su  -  u1
$cd  /admin
$mkdir  u1data
$ls  -li
$exit

#su  -  u2
$cd  /admin
$mkdir  u2data
$ls  -li
$exit
```

```
#chmod  u+s  /admin
or
#chmod  2770  /admin
#ls  -li  /admin
#su  -  u1
$cd  /admin
$mkdir  u1newdata
$ls  -li
$exit
#su  -  u2
$cd  /admin
$mkdir  /u2newdata
$ls  -li
$exit
```

```
#su - u1
$cd /admin
$rm -rf u2data
$exit
#su - u2
$cd /admin
$rm -rf u1data
```

Partitions

- ‘/’ in primary.
- ‘/boot’ in primary.
- ‘swap’ in extended.
- Common utility is ‘fdisk’.
- To check list of partition ‘fdisk –l’.

✓ To create partition ‘fdisk /dev/sda’
then: press ‘p’ and enter

✓ ‘p’ is used for list of partition table.

✓ ‘n’ for create a new partition.

✓ First option is ‘First cylinder’

✓ Type end cylinder size of your last partition.

✓ ‘+500’ for give size & enter.

- ✓Then press again ‘p’ to list, it shows your next partition.
- ✓For assign level, Press ‘’
  * ID’s of partition:
  a. 7 -> NTFS
  b. 83 -> Linux Partition
  c. 82 -> Swap
  d. 5 -> Extended
  e. 8e -> LVM
  f. fd -> Raid
- ✓<u>To delete any partition</u> :
- ➢Press ‘d’
- ➢Asking a number of partition which you want to delete.
- ➢Give number like: /dev/sda6
- ➢To save partition in table press ‘w’ & for quit ‘q’.

- ✓After save a partition, you must reboot your system:-
- ✓After reboot your system, you have to format your partition & mount it:
  #mk2fs.ext4 /dev/sda6
- ✓After format, make a directory for mount the partition:
  #mkdir /new

Mount

1. Temporary (Work when system is ON)
2. Permanent

Permanent: [Config a sysfile] CAREFULLY

- ➢#vim /etc/fstab
  We have many system entries. Go to last line in insert mode & enter.
- ➢/dev/sda6 /ram ext4 defaults 0 0

  for check partition for mount booting time.

  :wq

- ✓After mounting permanently
  #mount -a

ACL

- It stands for ‘Access Control List’, Which is a partition parameter of Linux. By using ACL we can configure our system to provide the permission to the users for accessing any object but without changing any ownership & without assigning any permission to the other.
- **LAB**:

➢ If u have File1 .

➢ #vim /etc/fstab

➢ Check partition have ‘/’ and after ‘default,acl’
:wq

➢ #mount

➢ #mount –o remount,rw /

➢ Set ACL to file:

➢ #setfacl -m u:username:rwx /file1

➢ #getfacl /File1 [To check permission to user]

Dynamic data storage tech.

- This technique is special storage technique which store the data into volumes. And volumes are made by the partitions, The basic difference b/w volumes and partitions are give an below:

i. Volumes are resizable in nature where partitions are fixed in size.

ii. Volumes are fast in speed compare than partition, Due to the data block size. The data block size of volume are 4MB. Where partitions data block size is 512 bytes.

- In dynamic data storage tech. we use two types of mechanism for creating a volume:

i. **LVM** : LVM stands for 'Logical Volume Manager'. And it is use to create volume of resizable nature.

LAB

1. Create a partition as per your requirement then change its type into LVM, for changing its type we press 't' in fdisk environment, then give '8e'.
   Then save the partition & reboot the system.

2. Now change the basic partition into physical volume for this, we perform these commands:
#pvcreate /dev/sda6
To check
#pvdisplay

3. Now create volume group
#vgcreate /dev/vg00 /dev/sda6
#vgdisplay

4. To customize data block size

#vgcreate -s 8Mb /dev/vg00 /dev/sda6
#vgdisplay

5. Now create LVM

a. Bases on size

b. Bases on LE [Logical extend]

a. Bases on size:
   #lvcreate -L 100M -n /dev/vg00/lv00 /dev/vg00

b. Bases on LV :
   #lvcreate -l 10 -n /dev/vg00/lv00 /dev/vg00

- Format the LVM :
  #mkfs.ext4 /dev/vg00/lv00
  #mkdir /ram

  #vim /etc/fstab
  /dev/vg00/lv00 /ram ext4 defaults 0 0
  :wq

  #mount -a

- Remove:
  #vgremove /dev/vg00
  #lvremove /dev/lv00

LVM increase size

- First unmount the partition
  #umount /dev/vg00/lv00
  check
  #df -h
- Check filesyatem
  #e2fsck -f /dev/vg00/lv00
  #lvextend -L +100M /dev/vg00/lv00
  #resize2fs /dev/vg00/lv00
  or
  #resize2fs -f /dev/vg00/lv00

LVM reduce

- #umount /dev/vg00/lv00
- #df -h
- #e2fsck -f /dev/vgoo/lv00
- #resize2fs /dev/vg00/lv00 100M
- #lvreduce -L 100M /dev/vg00/lv00
- #mount -a
- #df -h
- #lvdisplay

XFS Extend / Reduce

[+] Extend:

```
# lvextend -L +1G /dev/vg00/lv01 -r
# xfs_growfs /dev/vg00/lv01
```

[+] Reduce:

```
# xfsdump -f /tmp/test.dump /lvm2
# umount /dev/vg00/lv01
```

2) Reduce The Partition Size

```
# lvreduce -L 400M /dev/vg00/lv01
# mkfs.xfs -f /dev/vg00/lv01
# mount /dev/vg00/lv01 /lvm2
```

4) Restore The Data

```
# xfsrestore -f /tmp/test.dump /lvm2
```

Package Manager

In Linux package is the basically a s/w in compile format for managing package, red hat provide two package manager name :

1. RPM (Red Hat Package manager)

2. YUM (Yellow Dog Update Manager)

By using these package manager, we can perform some actions like: install, uninstall, query, update with packages.

RPM :

i. It’s stands for Red Hat packet manager. It’s an old packet manager in the industries.

ii. It’s entirely replaced by new package manager YUM.

iii. RPM syntax is complex compare than YUM.

iv. Whenever we want to install any package through RPM, we must required that package on local machine physically overall we can say that RPM is never work in server client environment or network environment.

v. RPM never resolve dependency.

vi. RPM doesn't support helping & searching.

LAB

1. Query: #rpm -qa vsftpd
2. Install: #mount /dev/dvd /mnt
   #cd /mnt
   #ls
   #cd Packages
   #rpm -ivh vsftpd[tab]
   #rpm -qa vsftpd
3. Uninstall: #rpm -e vsftpd[tab]
   #rpm -qa vsftpd[tab]
4. Update: (On other version of Linux like Ubuntu)
   #rpm -uvh packagename

YUM

- It’s stand for “YellowDog update manager”. It’s a new package manager in the industries ‘YUM’ syntax is easy compare than ‘RPM’.
- YUM works in server-client environment, whenever we want to install any package using YUM we don’t required that packages on local machine.
- YUM server is a machine having the complete dump(copy) of your private package database.
- YUM client is a machine having the information about the YUM server and location of package directory.
- YUM resolves dependency.
- YUM server is configured by using either “ftp” or “http”.
- YUM support helping & searching.

How to make “YUM” server

- #mount /dev/dvd /mnt
- #cd /mnt/Packages
- #rpm -ivh vsftpd[tab]
- #service vsftpd restart
- #chkconfig vsftpd on
- #rpm -ivh createrepo[tab]

It asking for the dependency so install all the dependency which is required for “createrepo”.

- #rpm -ivh deltarpm-[tab]
- #rpm -ivh python-delta[tab]
- #cd
- #cp -rvf /mnt/Package/* /var/ftp/pub
- #umount /mnt
- #cd /var/ftp/pub
- #createrepo -v Packages

How to make a client

- #vim /etc/yum.repos.d/client.repo
  type: [Packages]
  name=server
  baseurl=ftp://192.168.1.254/pub/Packages
  gpgcheck=0
  If we put ‘gpgcheck=0’ then no need for key.
  If we put ‘gpgcheck=1’ then need for key.
- If make user self machine as client & server-
- Type: [Packages]
  name=server
  baseurl=file:///var/ftp/pub/Packages
  gpgcheck=0
- Query: #yum list vsftpd
- Install: #yum install vsftpd*
- Uninstall: #yum remove vsftpd
- Update: #yum update vsftpd

Mount Image

- #mount -t iso9660 -o loop /ram/rhel6.2/mnt

Basic Networking

- To give IP:
- #setup -> network configuration -> Device configuration -> Select a device -> If give DHCP then mark [*] -> save & quit
- #service network restart
  #chkconfig network restart
- #ifconfig [To check mac & IP]
- #ping 192.168.1.10
- If need limited packets
  #ping –c 4 192.168.1.10
- To change hostname
  #vim /etc/sysconfig/network
  change hostname
  :wq
  #init 6 [must]

- To check hostname
  #hostname
- Any change on network
  #vim /etc/sysconfig/network-scripts/ifcfg-eth0
- To make DNS client
  #vim /etc/resolv.conf
  nameserver 192.168.0.254 [Its primary DNS]
  :wq
- Enable & Disable LAN card
  #ifdown eth0
  #ifup eth0

Telnet server

- Telnet: Telnet is an universal protocol which is available in all OS and telnet is use to manage the remote machine in command line interface mode.
- Port: 23
- Package: Telnet*
- Service or daemon: xinetd
- Config file: #vim /etc/xinetd.d/telnet
- Config steps: 1. config your machine as YUM client.
  2.#yum install telnet*
  3.#vim /etc/xinetd.d/telnet
  change “yes” into
  disable=no
  :wq
  4.#service xinetd restart
  #chkconfig xinetd on
- Note: By default root user is not allow to login via telnet. So, we create normal user.

- Client end: 1.Config your machine as YUM client.
  2.#yum install telnet*
  3.#telnet serverIP
- It asking for the username & password. Give & able to login.
- NOTE: If you want to work as a root then first of all login with user & type
  #su -
  it asking for password , give root password & able to login as root.

Linux Booting Process

1. POST (Power On Self Test)
2. BIOS initialization:- *Memory testing
   *I/O standard (All devices)
   * Boot Sequence
3. Boot Loader:- *MBR(Master boot record)
   *GRUB
4. Kernel Initialization:- Device detection & Drivers
5. INIT:- Common file service- /etc/rc.d/rc.sysinit
   Specific Services- /etc/rc.d/rc.?d

   Run Levels

   /etc/rc.d/rc.local
6. Xdm-> Xdisplay manager

SSH

- It is a Secure Shell & it is dedicated UNIX based protocol, which is used for remote management.
- But ‘ssh’ is a service which was designed for full filling the requirements of both ‘telnet’ & ‘ftp’ & in secure manner.

SSH SERVER

- Configure your machine as YUM client.
  #yum -y install ssh*
- Port no. = 22
- Service or Daemon = sshd
- Package = openssh* [Already installed]
- Config file = /etc/ssh/sshd_config
- #vim /etc/ssh/sshd_config
  :13
  Uncomment this line
  :42
  uncomment it also.
  :wq
- #service sshd restart
  #chkconfig sshd on

- To access with IP
  #ssh 192.168.0.16
- To allow ssh:
  #vim /etc/hosts.allow
  Go to end of file & write
  Remove uncomment end line & type
  sshd: 192.168.0.0/255.255.255.0
  :wq
- To deny ssh:
  #vim /etc/hosts.deny
  go to end & type
  sshd: 172.24.0.0/255.255.0.0
  :wq
- #service sshd restart
  #chkconfig sshd on

IPTABLES

The most popular firewall package running on Linux & it was ipchains, but it had a number of shortcomings. To rectify this, the Netfilter organization decided to create a new product called iptables.(IANA provides ip addresses)

1)Iptables are used to allow or deny a specific port which can access our system .

2)The way to enter in a systems is through protocols, And protocols are use ports to transmit data or control data flow over the network.

3)The main protocols which are used in industries are:- ssh, telnet, mail server etc….

The ports of the protocols which can be allow or deny through iptables are:-

Ssh-22

telnet-23

Mail server-25

Understanding Firewall:-

There are total 4 chains:

1) **INPUT** - The default chain is used for packets addressed to the system. Use this to open or close incoming ports (such as 80,25, and 110 etc) and ip addresses / subnet (such as 202.54.1.20/29).
2) **OUTPUT** - The default chain is used when packets are generating from the system. Use this open or close outgoing ports and ip addresses / subnets.
3) **FORWARD** - The default chains is used when packets send through another interface. Usually used when you setup Linux as router. For example, eth0 connected to ADSL/Cable modem and eth1 is connected to local LAN. Use FORWARD chain to send and receive traffic from LAN to the Internet.

Target meanings:-

*The target **ACCEPT** means allow packet.

*The target **REJECT** means to drop the packet and send an error message to remote host.

*The target **DROP** means drop the packet and do not send an error message to remote host or sending host.

#Iptables –L = it is used to list the iptables entries .

#Iptables –F = it is used to flush the iptables entries.

#service iptables save = it is used to save the iptables entries after flushing it

#chkconfig iptables off/on = it is used to on or off iptables permanently

- If we want to block a particular IP :

#iptables -A INPUT -s 192.168.10.1 -j DROP

- To delete blocked IP entry :

#iptables -D INPUT -s 192.168.10.1 -j DROP

Shell Script

- A shell script is a program written for BASH. The simplest shell script is a sequence of Linux commands.
- Shell scripts are similar to DOS batch files (Those files that end in ".bat").

SHELL PROGRAMMING IN BASH SHELL :

- In BASH shell the first line is always
  #!/bin/bash
- Here by using the "chmod" we can set the file permissions for the file.

- **Example** :
- #vim scriptname.sh

  #!/bin/bash
  echo “Hello….Root”
  :wq

- #chmod 777 scriptname.sh
- #./scriptname.sh
  OR
  #sh scriptname.sh
- OUTPUT: Hello….Root
- This script execute on shell is “BASH” & “echo” command to print the message.

SHELL VARIABLES

- Variables are referenced by “$name”.
- **EXAMPLE** :
- #vim script.sh

```
#!/bin/bash
echo “Enter the value”
read variable
echo variable=$variable
```

:wq

- #chmod 777 script.sh
- ‘echo’ command is to print the message & ‘read’ command to get input & $variable to show the output.

ARITHMETIC EXPANSION

- Arithmetic expansion is in the form of $((Expression))
- **EXAMPLE** :
- #vim script.sh

```
#!/bin/bash
echo $((5+5+5))
 echo $((1*2*3))
 echo $((8/2))
 echo $((9-1))
```

  :wq

- #chmod 777 script.sh
- #./script.sh
- OUTPUT: 15

  .

  .

CONDITIONS IN SHELL

- Command : condition ‘if’ (for BASH shell)
- The conditional ‘if’ statement is present in BASH shell & also in CSH shell & also in SH shell.
- SYNTAX :
  ‘if’ condition1
  ‘then’ command list ‘if’ condition1 is true
  ‘elif’ condition2
  ‘then’ command list ‘if’ condition2 is true
  ‘else’
  command list if condition is false
  ‘fi’ to finish the syntax

- EXAMPLE :-
- #vim script.sh

```
#!/bin/bash
if [ $1 == “1” ]
then
echo “The first choice is nice.”
elif [ $1 == “2” ]
then
echo “The second choice is just as nice.”
else
echo “Selection is wrong…!!”
echo “type valid number”
fi
```

:wq

- #chmod 777 script.sh

- OUTPUT : #sh script.sh 1
  The first choice is nice.
- #sh script.sh 2
  The second choice is just as nice.
- #sh script.sh 9
  Selection is wrong…!!
  type valid number

FILE DESCRIPTORS

- The file descriptor has two standard file descriptors:-
  a. stdin (Standard input to the program)
  b. stdout (Standard output to the program)
- File redirections are listed:
  > : OUTPUT redirect

  < : INPUT redirect
- **EXAMPLE** : #vim file1
  redhat
  redhat
  :wq
  #grub-md5-crypt < file1 > file2

- #vim file2
  encrypted password
  And copy the encrypted password & paste anywhere you want.
  :wq

CREATING A SHELL SCRIPT

- To create a shell script, use a text editor and enter your Linux commands.
- #vim script.sh

```
cd  /tmp
echo  “Removing  temp  files…..”
ls  -al
rm  -rfv  /tmp/*

:wq
```

- #chmod 777 script.sh
- #./script.sh
- ALL COMMANDS ARE EXECUTES.

TCPdump with Wireshark

- Popular network debugging tool
- Used to intercept and display packets transmitted/received on a network

- 1. Capture only udp (User Datagram Protocol) packets
- #tcpdump udp
- 2. Capture only tcp (Transmission Control Protocol) packets
- #tcpdump tcp

1. Capture only UDP packets with destination port 53 (DNS requests)

- # tcpdump udp dst port 53

2. Capture only UDP packets with source port 53 (DNS replies)

- #tcpdump udp src port 53

3. Capture only UDP packets with source or destination port 53 (DNS requests and replies)

- #tcpdump udp port 53

- Capture packet data in file-
  #tcpdump –w filename.pcap
- To read packet capture file-
  #tcpdump –r filename.pcap