Unit I Introduction – 2hrs 7 marks long

Page No.
Date: 13/Oct/

Linux
- monolithic kernel - linux
- recursive acronym for

1. GNU – GNU's NOT Unix Linux // NU Collection of Foss?
2. GNOME – Interface Visual Desktop environment
   – GNU Network Object Model Environment
3. Kudzu – Hardware management
4. ROOT/SuperUser – Allows to edit bin files/Kerne the root most file .../
5. Kernel – Betwn H/w & S/w, middleware
   – a unique way of process mngmnt, system resource & H/w mngmnt, a part of OS
6. Binaries – combination of 0's & 1's
   most impt files in binaries. Everything in file system in UNIX

*Blue Screen of Death – full sys fail at windows kernel level

Linux – RHEL Not free
Ubuntu, ... Open source

Adv of Linux	Disadvantages
– Stability – Control – Open Src – Installation in many ~~os~~systems use	1– For too many distros – thousands of OS versions 2– Linux is perplexing to 3– Trustworthiness 4– Lack of commercial software Photoshop → Gimp, Premier Pro → Da Vinci Outlook → Thunderbird office → libre office Antivirus → Clam AV

⇒ History of Linux

Unix = Dennis Ritchie and Ken Thompson (C programming)
AT&T Bell Laboratory, 1969, release 1970

Command line

1963 - Richard Stallman, GNU (project) A Unix-like OS, GNU, GPL
For general people
1987 - Minix Andrew S. Tanenbaum
1990s - S/W required were readily available
(Hurd) - fail
Linus Torvalds - Linux 1991 on MINIX using GNU Compiler

Disadvantages of Linux

5- Drivers are not easily found.
6- Games and entertainment.

⇒ Some important distros (distributions)

1 ~~Divine~~ Debian GNU linux
This non-commercial distribution started in 1993.
and it is the most ~~important~~ popular distribution.
- It has the largest developer community.
- It uses package manager called APT
(Advance Packaging Tool)
- ~~Divine~~ Debian is a huge distribution and it takes time to install.

GURUKUL

2 Ubuntu Linux
- This ~~div~~ Debian ~~Divine~~ based distro was initially released in 2004. It is gaining popularity over the years.
- It uses apt-get to install packages

3. Fedora
- This ~~is~~ a distribution once successor to Fedora Core is RedHat Linux which is the Linux distribution from RedHat.
- Fedora Core 1 was released on November, 2003. It uses RedHat Package Manager (RPM)

4. Gentoo Linux
- This non-commercial source based distribution ~~was~~ first appeared in 2002. This installer provides some binary packages to get Linux going but the idea is to compile all source packages in the user's computer. It is time consuming to install and has core desktop environment.

5 Knoppix
- This is based on Debian. It was built by Klaus Knopper from Germany.
- Used as a recovery system.

6. Slackware
- Released in 1992.
- It is one of the oldest distribution.
- It uses tar file (like zip for compression) tape archive

- You do all s/w configuration by editing text files.

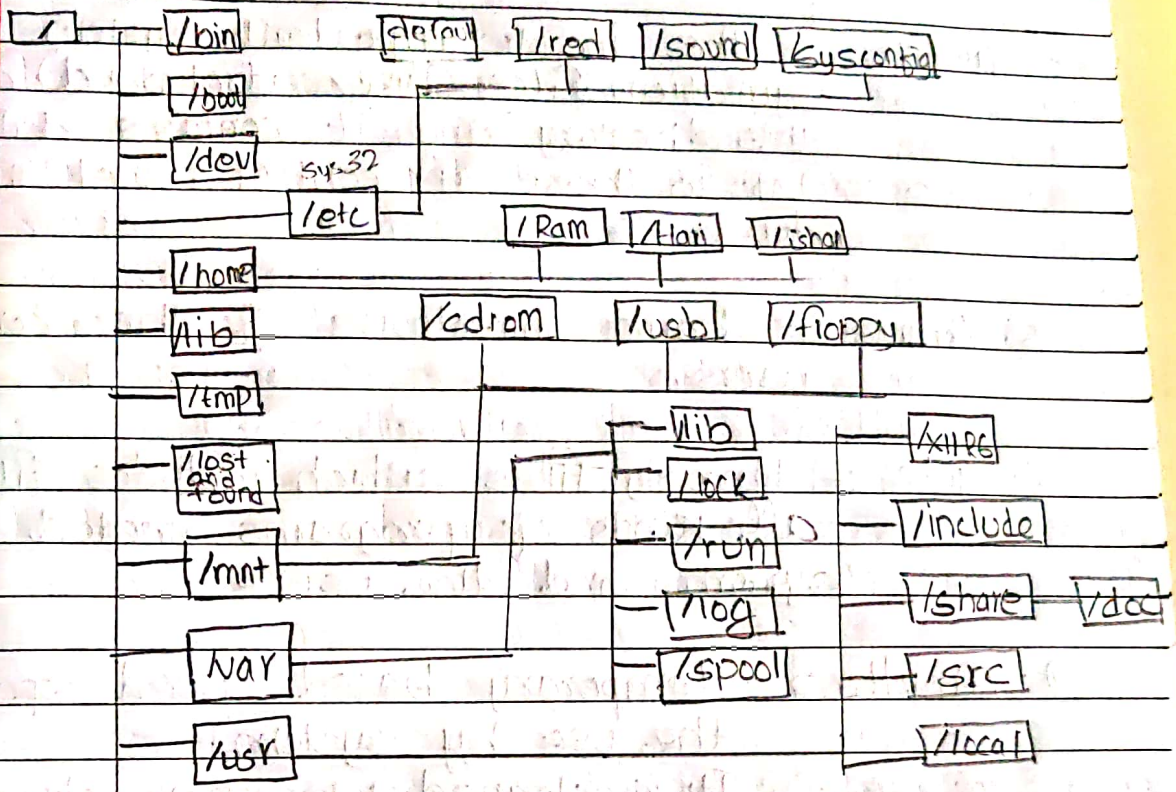
⇒ Making sense of version ~~com~~numbers

The version numbers of the Linux Kernel and Linux distributions are unrelated but each has a particular significance.
Version numbers are in the form of three integers separated by periods (.)

Major . Minor . Patch
0 . 0 . 1

Impt Directory Layout

Win - version
UNIX - flavour



1. /bin :- It contains common programms, shared by the system, these system administrator and the users.

2. /boot :- The startup files and the Kernel stored here. In some recent distribution, it also contains GRUB (Grand Unified Boot Loader) for multibooting.

3. /dev :- contains references to all the CPU peripheral hardware which are represented as files with special properties.

GURUKUL

Page No.
Date : / /

4. /etc :- the most important system configuration files (like control panel) are in this directory and it contains data similar to those in control panel in windows.

5. /home :- home directories of the common users.

6. /lib :- library files which includes files for all kinds of programs needed by the system and the users

7. ~~/tmp~~/tmp :- temporary ~~based~~ ~~used~~ space for the use by system.
It is cleaned up upon reboot. Here ~~de~~ don't use for saving any work.

8. /lost+found :- Every partition has lost+found in its upper directory. Files that were saved during failures are stored here.

9. /mnt :- mount :- Standard mount point for external file system. Ex : usb, cd-rom, floppy, etc.

10. /var :- storage for all variable files and temporary files created by users, such as log files, mail queue, the prints spooler area, space for temp-

*burn - write

Page No.
Date: / /

orary storage, a files downloaded from the internet, or to keep an image of a CD before burning it.

11. /usr :- programs / libraries / documentation for all user related programs, Bootable Pendrive

FAT, NTF, defrag, File Recovery
File Allocation System

Linux File System
- A file system is a structured collection of files on a disk drive or a partition
- A partition is a segment of memory that contains data, and every partition contains a file system.

A file system is a process that manages how and where data on a storage disk, typically a hard disk drive (HDD), is stored, accessed and managed. It is a logical disk component that manages a disk's internal operation.

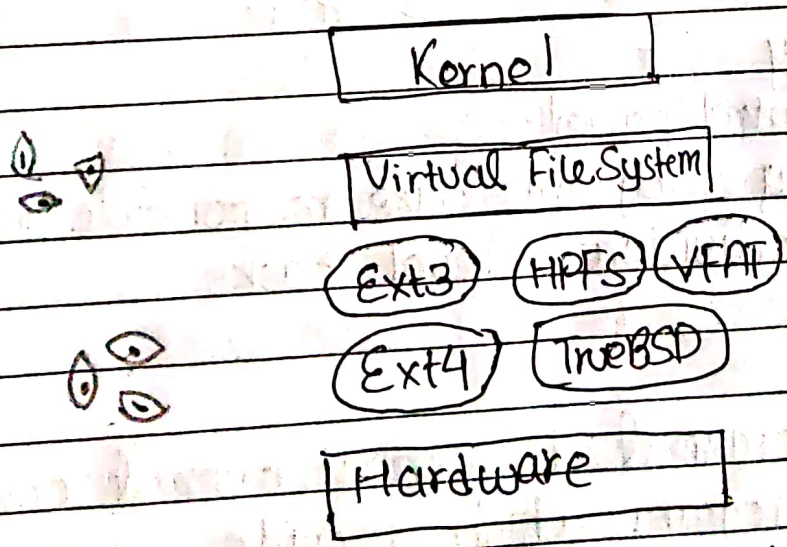


Fig: File System Architecture

GURUKUL

Page No.
Date : / /

Imp Q. Internal

Important Components of Linux
The important components of Linux are :-

i) Kernel
- Kernel is the core part of Linux which is responsible for all major activities of the operating system. Generally, Linux Kernel is monolithic (single language) It consists of various modules and interacts directly with underlying hardware. Kernel provides the required ~~ep~~ abstraction to hide low level hardware details to system or application programs.

ii) System Library
- System libraries are special ~~for~~ programs or functions using which application programs or system utilities access Kernel features. These Libraries implement most of the functionalities of the OS and do not require Kernel modules' code access rights.

iii) System Utility
- Things in control panel
- System utility programs are responsible to do specialized individual level tasks.

Imp iv) Shell
- Command Terminal to control overall computer
- Shell is a program which provides the interface between the user and Operating System.
- Shell has compiler

GURUKUL

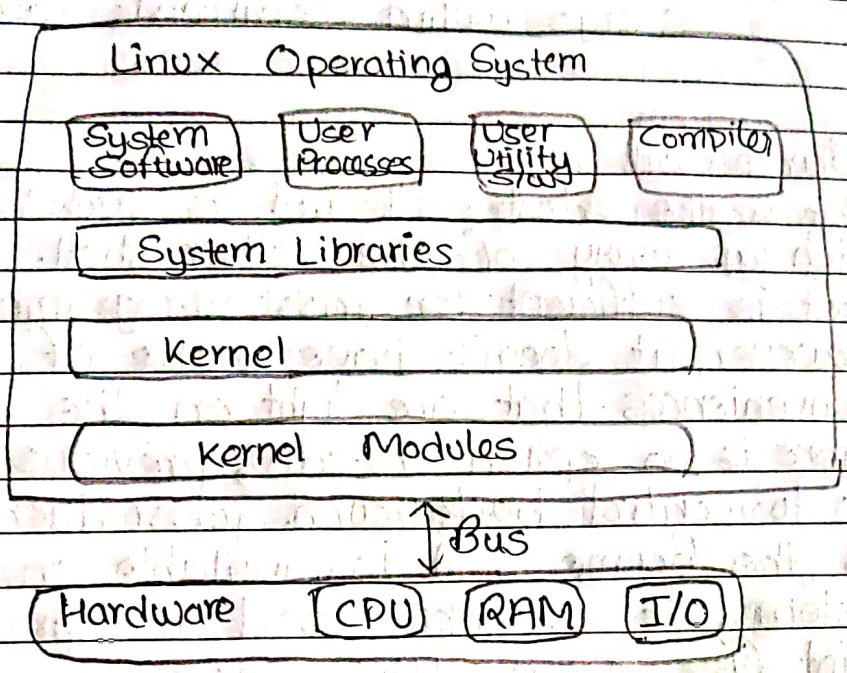


Fig: Linux Architecture

Types of Shell

1. Bourne Shell
2. C Shell
3. Korn Shell
4. Bash Shell
5. tcsh Shell (C Tener Shell)

Key Features of Shell :-

1. to allow shell scripts to be used.
2. To provide programmability using including control flow and variables.
3. Control overall input/output file descriptor.
4. Run program within a shell. For ex Vi, Vim
5. Run system level commands.

- daemons :-

Page No.
Date : / /

6. Control over signal handling within scripts
(args within commands handled)

1. Bourne Shell
- This is the oldest shell and as such has not as feature rich as many of the other shell. Its feature set is sufficient for most programming needs. However, it doesn't have some of the user conveniences that are light on the command line. There is no option re-edit previous commands or to control background jobs (daemons).
- As the Bourne shell is available on all UNIX systems, it is often used for programming script files as it offers maximum portability between different UNIX versions.
- Its default prompt character is $.

2. Bourne Again Shell (Bash)
- It is combination of Bourne Shell and C-Shell
- It has command line editor that allows using use of cursor keys (left, down, up, right).
- It facilitates view by giving a list of commands by typing the first few letters followed by the tab key.
- It is the ~~diffe~~ default shell on most Linux distros.
- Default prompt character is $ in Bash

Page No.
Date : / /

3. C Shell (Csh)
- The C Shell syntax is taken from the C programming language. The script syntax is similar to that of C programming for instance, conditional statements, operator priority, switch cases, memory allocation, etc. are similar to C programming.

4. Korn Shell
- It is based on Bourne shell. It has useful command line editing facility.
- You can recall and edit previous commands using VI (a text editor) or emacs keys.
- It works well across a network or using a physical terminal. It has powerful programming constructs (concepts) than the Bourne shell.

5. Tcsh (Tenex Shell)
- It emulates C Shell.
- It is more feature rich than Bash Shell.

Assign Types of file system in Linux (ext2, ext3, ext4)

ls - list
q - quit

ls -s size
--size
ls -S -t -b (first this, 2nd, 3rd)

-h = --help

Page No.
Date : / /

Command Syntax Structure (mostly based on bash)

$ "Command [option]" "<arguments>"
Bash eg: kill Pid

- The command, options and arguments are separated by blank spaces.
- A Linux command is usually an executable program residing on Linux disk.

Man Pages - Manual Pages

- It provides a detail manual for commands which explains the option in the command and their use
- Man pages also come in handy when you are using multiple ~~fo~~ flavours of UNIX or several Linux distros since options and parameters sometimes vary.

Press 'q' to exit the Man Page.

man $configfile

- Most configuration files have their own manual

⇒ 20 important Linux Commands

= Linux provides a CLI (Command Line Interface) to communicate with the OS. Here are the most basic of the Linux Commands.

~~/home~~ ~ = home

Page No.
Date: / /

1. pwd
This command displays the current working directory of the terminal.
Syntax: $ pwd

2. echo
This command writes its arguments to standard output.
Syn: $ echo "<text>"
Ex: $ echo "Hello World"
Hello World

3. su
This command is used to switch to root-user so that superuser permissions can be used to execute commands.
Syntax: $su
Ex: [user@localhost ~]$ su
Password:

4. su <username>
- This command executes only that command with root/superuser privileges.
syntax: $ sudo <command>
Ex:- sudo useradd <username> // Add a new user
sudo passwd <username> // set a password for the new user

sudo userdel <username>

6. clear
- To clear the terminal screen. Contents will be scrolled down only. ctrl + L for clearing the screen
Syntax:- $ clear

GURUKUL

Page No.
Date : / /

Linux Commands: Working with Files

7. cp

This command copies files and directories. A copy of the file/directory copied, still remains in the working directory.

Syntax:

$cp <flag> {filename} /pathname/

Example: $ cp testfile.txt /home/edure/Public

8. mv

It moves files and directories from one diretory to another.

Syntax: $ mv <flag> {filename} /pathname/

9. rm

- Remove files from a directory. By default, the rm command does not remove directories. Once removed, the contents of a file cannot be recovered.

Syntax: $ rm <flag> {filename}

Example: $ ls

$ rm testfile.txt

10. grep

To search for a particular string/word in a text file. Similar to Ctrl+F but in CLI

Syntax: $ grep <flag or element to search> {filename}

cd 'xx yy' - ' ' coz space in folder name

grep -i // returns results for case insensitive string

11. cat
- This command can read, modify or concatenate text files. It also displays file contents
Syntax: $ cat <flag> <filename>
Ex: $ cat cattest.txt
The content of cattest.txt
// Displays file content

12. ls - lists all the contents in the current directory.
Syn: $ ls <flag>

ls <pathname>
ls -l // lists all the contents along with its owner settings, permissions & time stamp (long format)
ls -a lists all the hidden contents in the specified directory

13. cd - to change the current working directory of the user
Syn: $ cd /pathname1
$cd /home/edureka/Public

cd ~ //changes the directory to home
cd / // " " " " root directory
cd .. // " " " " its parent directory
cd "xx yy"

14. sort
- sorts the results of a search either alphabetically or numerically.
- Files, file contents & directories can be sorted using this command.

syntax:- $ sort <flag> <filename>

Ex :- $ cat sorttest.txt
alias
cat
cd
rm
grep

:- $ sort sorttest.txt
alias
cat
cd
cp

sort -r results in reverse order
sort -n results as per numerical order

15. mkdir
- To create a new directory

$ mkdir <flag> <directoryname> /pathname/

mkdir -p <filename 1> / <f1, f2, f3>
// To create multiple subdirectories inside the new parent directory

16. rmdir
- Remove a specified directory.

Sy $ rmdir <flag> <directoryname>
Ex: $ rmdir newdirectory /home/edureka/

Page No.
Date: / /

Linux Commands: Working with user permissions

17. chmod
=to change the access permissions of files & directories.
Syntax:- $ chmod <permissions of user, group, others> <filename>

Ex: $ ./chmodtest.sh
Permission denied
$ chmod 777 chmodtest.sh
$ ./chmodtest.sh
This is file content.

Num	read	write	execute
0	-	-	-
1	-	-	Y
2	-	Y	-
3	-	Y	Y
4	Y	-	-
5	Y	-	Y
6	Y	Y	-
7	Y	Y	Y

Installing Packages

18. install packages
- For an RHEL based system
  Syntax: sudo yum install package-name
- For a Debian based system
  Syntax:- $ sudo apt-get install package-name

Page No.
Date : / /

Linux: Working with Zipped files

19. tar
- To zip files using .tar format
Syntax: ~~$ tar~~
$ tar -cvf tar-filename source-folders-name

To unzip files of .tar format
Syntax: $ tar -xvf tar-file-name

Linux Commands: Working with Secure Shell for Remote Machine Access

20. ssh // Secure Shell
- For operating network servers securely over an unsecured network
- Securing ~~rem~~ N/W service with SSH

Access to the master from the slave node
$ ssh <master's ip>

Access to the slave's node from the master
$ ssh <slave's ip>

*

Date
Page
20

Page No.
Date : / /

Open Source

⇒ Free/software Vs. proprietary Software

A software being open source doesnot imply the access of the source code, they must follow ten commandments also known as open-source high definition. (simply)

1. Free redistribution
- The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programms from several different sources. The license shall not require royalty or other fee for such sale.

2. Source code
- The program must include source code and must allow distribution in source code as well as compiled form where some form of the product is not distributed with source code. there must be a well publicised means of obtaining the source code for no more than a reasonable reproduction cost preferably downloading via the internet.

3. Derived works
- The licensee must allow modification and derived works and must allow them to be distributed under the same terms as the license (GNU GPL and MIT) of the original software.

4. Integrity of the author's source code
- The license may restrict the source code from being distributed in modified form only if the license allows distribution of patch files with the source code for the purpose of modifying the programs and build time.
The license may require derived works to carry a different name or version number from the original software.

5. No discrimination against persons or groups

6 No discrimination against fields of endeavour
- For example, it may restrict the program from being used in a business or for being used for genetic research.

7. Distribution of license
- The rights attached to a program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License must not be specific to a product
(// Entire program must be open source)

9. License must not restrict other software
- The license must not place restriction on other software that is distribution alone with the license to software. for ex:- the license must not insist all other programs distributed on the same medium must be open src. software.

GURUKUL

10. MWD

Page No.
Date : / /

10. License must be technology ~~re~~ neutral

Culture of free software

- We have four essential freedom

i) Freedom 0
- The freedom to run the program as you wish for any purpose.

ii) Freedom 1
- The Freedom to study how the program works and change it so ~~that~~ it does computing as you wish. Access to source code is a precondition to this.

iii) Freedom 2
- The Freedom to redistribute copies so you can help others.

iv) Freedom 3
- The Freedom to distribute copies of your modified versions to others by doing this you can give the whole community a chance to benefit from your changes.
Access to source code is a precondition to this

Imp*

* Special characters in Linux

Character	Feature / Meaning
~	Home directory
`	Command Substitution
#	Comment
$	Variable expression
&	Background job
*	string Wildcard
(	Start Subshell
)	Close Subshell
\	Quote next character // forward slash
;	Shell Command Separator
/	Path name directory separator // Back slash
!	Pipeline logical NOT

↳ Text Editors in Linux
(GUI in server take a lot of processor power
So, we use ~~com~~ CLI in server in sever in
Shell / Terminal)

i) Vi
- It is the default editor that comes with UNIX like Operating System.
Vi = Visual Editor
- Alternative Editor :- Pico, emacs
- Vi editor has two modes of operation :
  a) Command Mode
  b) Insert - Command Mode commands which

GURUKUL

cause action to be taken on the file.

b) Insert Mode
- In which entered text is inserted into the file.

In the command mode, every character typed is a command that does something to the text file being edited. A character typed in the comm. and mode may even cause the Vi editor to enter the insert mode.

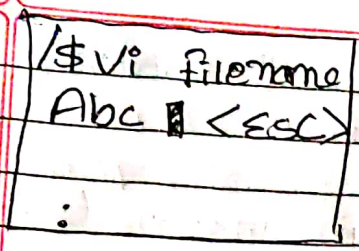
Command mode
- Insert
- Delete
- Edit

In the insert mode, every character ~~added~~ typed is added to the text in the file; press the ESC key turns ~~ALTP~~ OFF the insert mode.

Both UNIX and Vi are case sensitive.

Q How to start Vi ?
- Open the terminal.
Enter location : $ vi filename

Q How to exit Vi ?
- Usually, the new or modified file is saved when you leave Vi. However, ~~new or~~ it is also possible to quit Vi without saving the file. The cursor moves to the bottom of the screen whenever colon (:) is typed.



Exiting Modes

:x <Return>	Quit Vi writing out modified file to file name in original invocation
:wq<	Same as above))
:q quit vi editor	quit vi editor

Features of Vi

- Its memory footprint is very low.
- It is command centric. You can perform complex text related tasks with few commands.
- It is a default editor.
- It is highly configurable and uses simple text file to store its configuration.

* Moving the Cursor
- The mouse does not move the cursor within Vi editor screen. You must use the key commands given below:-

j	Cursor down one line
k	Move cursor up by one line
h	Move cursor left one character
l	Move cursor right one character
0 (zero)	Move cursor to the start of current line
$	Move cursor to the end of line
w	Move cursor to beginning of next word
b	Move cursor back to beginning of preceeding word
:num	Ex: ~~Color~~ :23 Move cursor to line 23
:$	Move cursor to the last line in file
:0	Column zero :0 to First line

~~String~~ screen Manipulation

^f	move forward one screen (page up)
^b	move backward one screen (pg down)
^d	move down one half screen
^u	move up one half screen

⇒ Types of file system in Linux

- The file system may be defined as the way of storing the data in a particular manner so that it could be located smoothly when required.

- In Linux OS, the mounting & unmounting of the hard drive is not possible while the OS is running. Therefore in the beginning, while installing the OS, you will get the chance to

choose the part of the disk that you will be using or that you want to be occupied by any file system.

Below are ~~the~~ some of the file systems in Linux:

1) Ext 4 [Extended File System]
- The Linux OS allows us to use the Ext 2, Ext 3, Ext4 file system, where Ex4 has been considered the modern distribution & the most efficient one. It is a very compatible option for the SSD (Solid state drive) disks & it is the default file system in Linux distribution.

2. TFS
- This file system is popular due to its simplicity & the capability to work according to the hard drive size. It stands for a journaled File System & has been developed by IBM. This file system uses very little processing power of the CPU. & due to its feature, it is preferred by ~~the~~ some of the system administrator.

3. Reiser FS File System
- Reiser FS File system is an alternative to the Ext3 file system. It has improved performance & advanced features. In the earlier time, the Reiser FS was used as the default file system in SUSE Linux, but later it has changed some policies, so SUSE returned to ext3

4. XFS File System
- XFS File system was considered as high speed EFS (IFS) which is developed for parallel I/O processing. NASA still using this file system with its high storage server (300+ Terabyte Server).

5. btrFs (Better / Butter / B-tree) File System
- It was developed in 2007. It provides many features such as snapshotting, drive pooling, data scrubbing, self-healing & online defragmentation. It is the default file system for Fedora workstation.

6. FAT 32 File System
- This is the most usual file system in all OS. This is because it provides the best way to store the files so the OS can find it easy to handle there files & directories.

Syntax :- Commands Options Parameter/Arg

Page No.
Date : / /

⇒ Common Commands in Linux

1) Sudo (super user do / Run as admin)
- It is shorthand for Superuser Do that lets you perform tasks that require administrative or root permissions.
- When using sudo the system'll prompt users to authenticate themselves with a password.
- Every root user by default can run sudo commands for 15 mins.

Options :-

- -K : AKA reset timestamp (Invalidates the time stamp files)
- -g : (Group) Runs commands as a specified group name or Id
- -h : Runs command on the host

2. pwd (Present Working Directory)
- It returns the full current path starting from the / (root) upto ~~the~~ your present working / current directory.

Options

- -L : (Logical) ~~Friends~~ Prints environment variable content including symbolic links
- -P : (Physical) Prints the actual path of the current directory

classmate

Page No.
Date: / /

3. cd (Change directory)
= It is used to navigate through the Linux files & directories, it requires either the full path or the directory name.
Running this command without an option will take you to the home folder.
Some shortcuts :-

- cd ~ (username)
  - Goes to other user's Home directory
- cd ..
  - Moves one directory up.
- cd -
  - Moves to your previous directory

4. ls (lists files and directories within a system)
- Running it without a flag or parameter will show the current working directory's content
Examples:-
ls /home/~~work~~ user 2/

Options :-
- ls -R : lists all the files except hidden files in the sub directory
- ls -a : lists all the hidden files in addition to visible ones
- ls -lh : shows the file sizes in easily readable formats such as MB, GB, TB.

GURUKUL

1 2 4 - user group other

5. cp (Copy)
- Copies files or directories and their contents
Ex:-
- cp filetext.txt /home/user/Document
- Condition you should be inside the directory containing filetext.txt

- cp filename.txt filename2.txt filename3.txt /home/users/Documents
- To ~~move~~ copy three files to document directory

- cp filename.txt filename2.txt
- copy content of one file to the another file

6. mv
- Move and rename files and directories.
- mv filename.txt filename2.txt /home/user/Documents
- to move multiple files

- mv filename.txt filename2.txt
- to move content of one file to another.
* uses ~~list~~ ls lists to see o/p.

7. mkdir (Make dir)
- To create one or multiple directories at once and set permissions for each of them.
- Ex:-
mkdir music
- Make directory Music

- mkdir Music/songs rmdir - removedir
- Make multiple

GURUKUL

* Verbose :- in detail (step by step)

Page No.
Date : / /

- Options

- -p :- parent Music / 2022 / songs
  Create a directory between two existing folders. For ex:-
  mkdir -p Music/2022/songs

- -m :- set the file permissions
  Ex:- mkdir -m 777 Music/songs
  1,2,4 get Read Write & Execute permission

- -v : Verbose output (In detail every step)
  Prints a message for each created directory

Permissions :-
Read Write ~~Execute~~

8. Boot
Boot sequence HDD, CD (whichever contains Boot file)
GPT partioning ~~not~~ Hard drive may not always support so MBR. VmWare, Virtual Practice Boot

swap
- Swap partitions are used to support virtual memory. In other words, data is written to a swap partition when there is not enough RAM to store the data your system is processing

GURUKUL

Amt of RAM in the system	Recommended swap space	Recommended swap if allowing for hibernation
less than 2GB	2 times the amt of RAM	3 times the amt of RAM
~~2GB~~ 2GB-8GB	Equal to the amt of RAM	2 times the amount of RAM
8GB -64GB	4GB to 0.5 times the amount of RAM	1.5 times the amount of RAM
More than 64 GB	Workload dependent (at least 4GB)	Hibernation not recommended

Installation type	Recommended minimum RAM
Local	

Sure Impt

LINUX Administration

Definition Impt

Assign Q What are your responsibilities as a Linux Administrator (10 mrks 2pg)

Linux Administrator

some major duties

- Plays a pivotal role in the org as he/she executes ^

⇒ ~~The~~ My responsibilities as a Linux Administrator are:-

1. Install, configure and maintain the Linux servers and workstations. ~~They~~ I am responsible for maintaining the network environment as well as the health of the network & servers.
Make sure ~~that~~ I to provide solutions by complying with the security standards of the company.

GURUKUL

2. Support the requests of the users & solve any problems related with the Linux servers & workstations. Set up and configure new systems, install and maintain the application software & coordinating the networking connectivity.

3. Be involved in designing, implementing, securing & maintaining the computer systems for its test, development and production environment.

4. Identify and analyze the issues that hamper the performance of the system, & to work in close coordination with the product development team and recommend the solutions for the issues.

5. Be involved in developing and overseeing the backup, replication, clustering & fail over strategies.

6. Participate in cross training activities to support the database. Train the subordinates in system administration & software deployment responsibilities. Monitor important services & server related issues & make sure that proper backup is provided for the N/W equipment configurations.

7. Conduct security audits on a routine basis as well as plan & expand the current offerings related to services.

8. Maintain proper documentation of all the procedures, configurations, programs. Update & upgrade the N/W hardware & software.

GURUKUL

9. Write & modify scripts for application deployments as well as system monitoring.

10. Maintain & monitor all patch releases & design various patch installation strategies and maintain all systems according to NIST standardization

11. Develop infrastructure to provide support to all business requirements. Perform regular trouble shoot & on system to resolve all issues

12. Ensure and evaluate availability & optimization of all server resources.

1. Server Admini
2. Programming & coding
3. Analytical skills
4. Information security
5. Communication skills
6. Problem-solving skills

Duties:- . Installing & configuring server
" " application s/w
Creating & maintaining user a/c
Backing up & restoring files
Monitoring & turning performance

sudo, sudoedit - execute a command as another user

Imp / Short Note

Pam - packet authentication module

GURUKUL

Page No.
Date : / /

Man in middle - PAM checks digital signature
su command
Available options :-

i) -c --command=command
Pass command to the shell with the -c optio

ii) -f, --fast
Past -f to the shell, which may or may not be useful, depending on the shell

iii) -g, --group=group
Specify the primary group. This option is available to the root user only.

iv) -G, --supp-group=group
Specify a supplimentary group. This option is available to the root user only. The first specified supplementary group is also used as a primary group if the option --group is not sperified.

v) -h, --help
Display help text and exit.

ISO → Ubuntu
burn in USB to make bootable using Rufus
Swap space
Sequence - USB <4GB - 8GB
everthy in Pendrive is deleted

20.0.4 x Issue

Pat ISO
Rufus
Virtual Box — Dowld ISO Ubuntu

⇒ Steps of Installing Linux (8 mrks)

Step 1) Make sure your computer has required specification Usually, 2 GHz processor, 2 GB of RAM, 5 GB Hard drive space (25 GB Recommended) and a DVD or USB port.

Step 2. Create a Bootable USB Drive or get an Installation DVD of your desired Linux OS.

Step 3. ~~Step 3~~ Partition computer's hard drive. The Ubuntu support page recommends 25 GB of Free Space.

Step 4. After you enter the Boot mode, you'll be prompted to select a language, desirably English.

Step 5. In this step, you can choose to try ~~Ubuntu~~ Ubuntu without Installing or actually install on your Hard ~~Drive~~ Drive or SSD.

Step 6. A new Dialog Box appears which is called the Installation ~~User~~ Wizard.

Step 7. After clicking Forward, it'll check the minimum requirements for running Linux. If everything is fine, we can see green colored tickmarks

GURUKUL

Step 8.

We can also select to download updates while installing and install some third-party software.

Step 8. Now, we choose either erase or use specific parts partition manually. ~~We~~ You can choose the first option if you want Linux to exist in your system else select the second option. Now, it will display the free space available in your PC. Select the free space and click on 'add' option to create a new partition and choose partition type as primary, size around 70% of the free space available or choose anything like 10,000 or 20,000 ~~MB~~ mb, use EXT3 or EXT4 Journaling File system or ~~EXT4~~ above ~~or~~ and select mount point as / forward slash.

- Now again select free space from the table & click add option. Now select size to be around 300 MB, use EXT3 Journalling file system & select Mount point as / slash boot.
- Now again select free space from the table & ~~select~~ click add option ~~mount point~~. Now select size to be around twice the size of RAM and select use as swap area and click OK.

Step 9:- Click 'install now' button and the Wizard will ask ~~en~~ us a location. Select our location and click forward.

Step 10. Now select the desired keyboard layout and click forward.

Step 11. Now fill in the details: filename, computer name, choose a username, and create a password. Then click forward and let your Linux copy all the essential files until Installation complete wizard appears.

Step 12:- Restart your PC and enjoy your Linux.

11 Nov Deadline Assignment

Q ⇒ Partions & partions numbering. Write the advantages & disadvantages of Disk partitions. You should mention the F Disk command and its role in disk partitioning.

↳ F Disk Command

F Disk also known as Format Disk is a dialog driven command in Linux used for creating & manipulating Disk partition Table. It is used for the view, creation, deletion, change, resizing, copying, and moving partition on a Hard drive using the Dialog driven Interface. F Disk allows you to create a maximum of four primary partition and the number of logical partion depends on the size of Hard disk.

It understands GPT, MBR (Master Boot Record) and BSD (Berkely) Partition Table:- Memorize at least 15 out of 20 options)

Options Allows user to :- • create space for new partitions
• Organize space for new drives
• Re-organize old drives
• Copying or Moving data to new disks/partitions

GURUKUL

Page No.
Date: / /

Syntax:-
fdisk <options> device

Examples:-
$sudo fdisk -l
View all disks partitions. This command is used to list the partitions on your system and see their /dev.

$ sudo fdisk /dev/sda
- View all fdisks commands (To see all the commands which are available under fdisk command you can use /dev/sda. This will prompt for a command (m for help). Type m for seeing all the operations which can perform on /dev/sda.

$sudo fdisk -h
- To see help message and listing of all options.

Options :-

i) -b
- Specify the sector size of the disk. Valid values are 512, 1024, 2040 and 4096

ii) -l
- lists all the partition table

iii) -V
- Display version information & exit.

iv) -L
- Colorize the output.

GURUKUL

⇒ root login -su

sda - SATA disk Array

v -x

- Similar to listing -L but provides more details.

vi -o

- Specify which output columns to print. It is used to create a default partition table in empty device.

su Vs sudo su - requires psud of the target a/c
sudo - ;)) " current user
great laptop virtual box hence, sudo is safer

Repeated RHEL Book

:- Check disk space ⇒ Linux command

1 ⇒ df command (disk free) - Shows the amount of disk space used and available on Linux file system

2 ⇒ du command (disk usage) - Display the amount of disk space used by the specified files & for each disk subdirectory

Assign

1. What are differences betwn du and df commands
2. What are the options available for du command?

df [option OPTION]... [FILE]... // df command syntax

df [-K] [-P|-t] [-de] [file...]

-K Use 1024-byte units, instead of the default 512 byte units, when writing space figures

-P Use standard portable, output format

GURUKUL

classmate

sudo - superuser / substitute user do

Page No.
Date: / /

-m Show output size in one-megabyte
-h Display in more easily human readable units such as KB, MB, GB or TB.
-T option to display the type of each file system listed such as btrfs, ext2
-t This command is used when you wish the disk usage information of the file systems having a particular type only
File Write the amount of the free space of the file system containing the specified file

Sample O/P

Filesystem	1K-blocks	Used	Available	Used %	Mounted on
/dev/sda	293026584	6940528	28595472	3%	/data

→ du command

- The du command can be used to track the files & directories which are consuming excessive amount of space on hard-disk drive.

Syntax:-
du [options] [file]

Ex:-
1. If we want to print sizes in human readable format (K, M, G) use -h option du -h /home/mandeep/test
2. Use -a option for printing all files including directories du
du -a -h /home/mandeep/test

GURUKUL

3. Use -c option to print total size
- du -c -h /home/sandeep/test

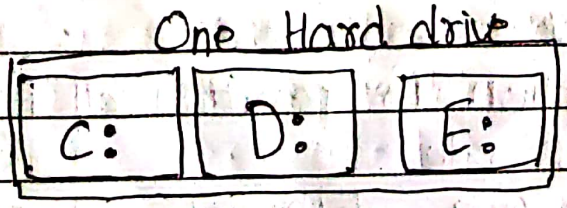
4. To print sizes till particular level, use -d option with level no.
du -d 1 /home/sandeep/test

⇒ Advantages and disadvantages of Disk partitioning

Advantages :-

1) Install multiple operating systems
- You can add a partition to a drive to install another operating system on it.
For example, you can run Linux alongside Windows but your computer cannot handle a virtual box. Here you can ~~install~~ create a new drive partition for Linux without touching your existing windows system.

2. Improved data security and protection
- If ransomware lands on your Windows partition, it may have a lesser chance of locking your personal files on another partition
- In addition, by partitioning the disk, you can ensure you have multiple ways of backing up files



3. Multiple File System
- ~~Partion~~ Partitioning drives let you store data on different file systems like NTFS, FAT32, exFAT, and EXT, which is free to manage your data, including updating or storing.

4. Well-organized
- If your drive has all sorts of files such as music, video, pdf, etc without partietioning the drive according to file types & preferences, its messy, tight and difficut and time-consuming to navigate & access the files.

5 Faster processes
- Without a partitioned hard drive, more processes & sequences have to be accessed, it will just But partitioning the drive saves you time & energy to navigate & access files & execute programs.

Other:- Backup is simpler

Disadvantages

1. Complicated Process & Chances of errors
- The complexity of having multiple partitions introduces more chances of mistakes. While formatting one partition, you might accidentally erase another

2. Juggling Partitions & Wasted space
- Multiple partitions sometimes leads to having cramped space for one partition but have plenty of free space for another.

↓↓
Space wastage

GURUKUL

Page No.
Date : / /

8. False sense of security
- While data loss in one partition may not affect data/entries on another partition, those partitions are all still on the same physical drive.
- If your drive fails or is destroyed, you'll lose everything on it.

4. Unnecessary for the Average user
- Typical computer users don't typically have enough files that they need a disk partition to manage them. And they don't often install other operating systems

5. SSDs Negate many past benefits
- Many reasons for why partitioning was beneficials ~~are~~ don't matter much now due to wide spread inclusion of SSDs (solid state Drives) in modern computers.

6. Slower Data Moves
- Moving data from one partition to another takes awhile unlike in the same partition

7. Reduced Space
- When you have two partitions, some space is lost.

Disk partitioning
- AKA disk slicing is the creation of one or more partitions on secondary storage, so each region can be managed separately.
- These regions are called partitions.
- First step of preparing a newly installed disk, before file system is created.

Page No.
Date: / /

Fdisk commands & its role in disk partitioning.

1. View all disk partitions.
- To list all the partitions on your system & see their /dev names.
For ex: - /dev/sda, /dev/sdb, /dev/sdc

Syntax :- $ sudo fdisk -l

2. View partition on a specific disk
- To view all disk partitions on device /dev/sda

3. View all fdisk commands
- To see all the commands which are available under fdisk command, you can use /dev/sda partition with fdisk command
$ sudo fdisk /dev/sda

4. Create a Hard disk partition
- First go inside the hard drive partition that is the /dev/sda partition, & use
$ sudo fdisk /dev/sda

Now type 'n' to create new partition
Then type 'p' for making primary partition
or 'e' for making extended partition

Run w command to write the changes & reboot your system

5. Delete a Hard disk partition
- $ sudo fdisk /dev/sda
and type 'd' to go to delete partition menu.
It'll prompt the partition number you want to delete (type the number)

Run 'w' command to write the changes & reboot the system

6. View the size of your partition
$ sudo fdisk -s /dev/sda

Notes:
$ man fdisk ⇒ to check for the manual page of fdisk command

$ sudo fdisk -h ⇒ to see the help msg & listing of all options.

⇒ Hard drive naming convention in Linux
- There's no C or D drive in Linux. So you'll typically see /dev/sda, /dev/sdb, /dev/sdc, /dev/nvme0n1, /dev/nvme0n2 nvme1on1, etc.
- 'dev' is short for device and
sd is short for SCSI mass-storage drive wher SCSI (Small Computer System Interface)
- If your computer has NVMe SSD, a newer SSD disk that plugs directly into motherboard's PCIe slot, you'll see nvme0n1.

When Windows sees Disk 0 and Disk 1, Linux ~~takes~~ detects the first hard drive carrying sda label (numerically as hard drive 0) and the second drive is sdb, the third is sdc and so on

⇒ Partition numbering

- To install an operating on a hard drive, it must be subdivided into distinct storage units called partition.
- Under the MBR partitioning scheme, there are different types of partitions:-
  - i- Primary
  - ii- Extended and Logical partitions

i) With MBR, any partition that is not explicitly created as an extended or logical partition, is a primary partition.
- There can be no more than four primary partitions. (sda1, sda2, sda3, and sda4)
- Unlike hard drives, partition numbers start from 1, not 0.

- Any space that is not allocated to the primary partitions is listed as Free or free space.
- The free space, however, is unusable because ~~that~~ as far as system is concerned the free space does not exist.

MBR - Master boot record

So to get around the four primary partition limitation of MBR, extended partition was introduced. The extended partition makes it possible to create many more partitions under it. AKA logical partitions.

Being able to create logical partitions comes in handy when you have to dual-boot Linux with other Operating systems

Types of partitions:-

1. Partition primary
- Contains one file system & typically stores the boot files for the primary OS can be set active & has a max of 4 primary partitions on MB2 disks

2. Extended Partition
A defined area where logical drives are stored Max of one partition per drive.

3. logical Partition (Unlimited number of)
- It can be used to store data, but cannot boot an operating system. Often used as an organizing tool limit of logical partitions only restricted by disk storage of disk.

ssh

Page No.
Date: / /

ctrl Alt T pwd
cd.. cd .. home
ls igic
man sudo // in detail
ls ls -a //shows hidden • local means private → hidden
no hidden list
touch filename.txt create new file
ls
vi. file.txt
cat mv.. file.txt documents
$ top for task manager – Activity Monitor but we can't kill process so to end task sudo kill 011(pid)

SSH – Secure Shell

– For remote system login using commands in Linux
– SSH is often used for logging into remote servers as root. However, the default configuration in OpenSSH prevents root login using password. To enable root login, change the value of PermitRootLogin configuration option in /ssh/sshd_config (d for daemon any sys that is running silently in the background)

Kudzu ✓ S.N How to configure

1. It checks the hardware connected to your computer (wake signals from H/W
2. It compares the h/w it finds to the database of h/w info stored in (/etc/sysconfig/hwconfig file) config file
3. Prompts you to change system configuration.

$ -> # when using su ~~su admin~~ super user

Page No.
Date : / /

* listing loaded module
- RAM, - Ex of module
Command # lsmod
To see which modules are currently loaded into the running kernel on your computer, you can use the lsmod command. Basically "lsmod" is a command that is used to display the o/p from "/proc/modules" file. Below is ex of the "lsmod" command in action.

lsmod

* loading modules
Command :- ~~# modt~~ # modprobe parport
// probe - request parport -> parallel
You can load any module that has been compiled and installed (to the /lib/modules directory) into your running kernel using the modprobe command.

* Removing modules
# command :- # rmmod parport_pc

=> Assignment 8 mrks

Q. Su vs Sudo Q. Sure Qsn
=> points to include
- vim stat :> sudoers lists
- configuring sudo command

Page No.
Date: / /

Q. Monitoring system performance

- We have several commands that provides us with relevant information about our Linux system. For instance, memory space used, unused, number of threads, CPU resource and so on.
Here are some of the commands we can use to monitor our system's performance.

1. top - process activity monitoring command. It displays linux processes in real time and updates the list every five seconds. List of hot keys (keys that can be pressed when top is active)

	Hot keys	Description
i	t	Displays summary information
ii	m	Displays memory information
iii	A	Sorts the display by top consumers of various system resources
iv	f	Enters an interactive configuration screen for top
v.	k	Issues kill command

2. htop - Improved version of top.
It uses colors and visual information about processor, swap, and memory status. It allows you to scroll horizontally and vertically.

3. vmstat - virtual memory statistics
This command reports information about processes, memory, paging, block I/O, traps and CPU activity

4. d Options:-

Option	Description
-m	Displays memory utilization slabinfo. (sys resource memory block info)
-a	Get information about active/~~and~~ inactive memory pages

4. lsof list of open files. ~~his~~ used to display all the open files & processes. The files include disk files, Network sockets, pipes, devices and processes

5. uptime - how long ~~ha~~ has the Linux system been running

6. tcpdump - It is a network packet analyser or packet sniffer that is used to capture or filter tcp/ip packets.
We can store the result in the file.

7 netstat - It is command line tool for monitoring incoming and outgoing network packets (tcp, udp, smtp, etc)

Page No.
Date: / /

⇒ Su vs Sudo

- The full form of Su is 'substitute user'. (switch to root) The job of the su command is to execute commands and provide the privilege of another account of the user. Upon execution, the su command invokes a shell and doesnot change the present working directory or the environment & the user.

- The full form of Sudo is 'substitute user do'. The job of Sudo command is to allow users to run programs & provide security privileges for another user.

The differences between Su and Sudo are:-

	Su	Sudo
1	Su command users need to create separate root and user account passwords during installation.	1 Sudo command users only need to provide a single password.
2	Su invokes the root shell and keeps it open during normal functioning.	2 Sudo runs fewer commands in roots and increases security.
3.	The users can assume another identity of another user without changing the login credentials.	3. The user does not need to change identity.
4.	It can expose the entire system information and poses the risk of modification.	4. Sudo grants privileged permission only for specific command demanded & hence is safer to use.

Parameter	Su	Sudo
Type	It is a command	It is privilege authorization
Operating System	Unix and Unix-like	Unix like
Function	To seek root permission through switching to superuser or root user	To seek root permission through a single command
Command	It does not commonly include another command.	It commonly includes another command.
	It doesnot have definable constraint.	It has definable constraints
	It doesnot keep a log of all commands.	It keeps a log of all commands.
	It doesnot elevate the privileges of the user.	It elevates the privileges of the user.

Configuring sudo

- A properly configured sudo is very flexible and has number of commands that need to be run may be precisely configured.

The syntax of configured sudo line is

User_name Machine_name = (Effective_user) command

The command can be divided into four parts:-

su will ask for the pswd for the user whom you are switch to

to become root

sudo will ask your pswd
sudo apt-get update

take permission from root
switch to root user only for that command

RHEL-

Chapter 5

User Management

a) Root ~~conte~~ a/c - Everything except Kernel
b) System a/c
c) User a/c -i) Group

Types of User

a) Root account
⇒ It is also called superuser account and would have complete unrestricted control of the system. A super user can run any command without any restriction. One should assume as a system administration account.

b) System account
⇒ System accounts are those accounts which are needed for the operation of system specific components for ex:- mail accounts, ~~SSHD~~ Sshd (Secure Shell daemon), bind9 account.

c) User account
⇒ User accounts provide interactive access to the system for users and the groups of users. These accounts are usually assigned to have limited access to the critical system files and directories.
- Group A/c 'A' (Privilege ...)

↳ Group A/c
A number of user accounts can be grouped together to create a group account. A Linux / UNIX group plays an important role in handling file permissions and process management.

Sys 32 - most impt

Imp Past Qsn

* Mention user management commands

= There are four important directories for managing users & groups.

i) /etc/passwd
- Keeps the user account and password information. This file holds the majority of information about the accounts on the UNIX/Linux system

ii) /etc/shadow
- Holds the encrypted password of the corresponding account.

iii) /etc/groups
- This file contains the group information for each account.

iv) /etc/gshadow
- This file contains secure group account information.

Note:- We cat command to view the file system

① Creating an account

#useradd
- It is used to create a new user account.

#useradd <-option> <fieldname> <account name>

Options

i) -d <directory_name>
- Specifies the home directory for the account.

Chapter 5

RHEL-

Page No.
Date : / /

User Management

a) Root ~~conte~~ a/c - Everything except Kernel
b) System a/c
c) User a/c -> Group

Types of User

a) Root account

⇒ It is also called superuser account and would have complete unrestricted control of the system. A super user can run any command without any restriction. One should assume as a system administration account.

b) System account

⇒ System accounts are those accounts which are needed for the operation of system specific components for ex:- mail accounts, ~~SSHD~~ sshd (Secure Shell daemon), bind9 account.

c) User account

⇒ User accounts provide interactive access to the system for users and the groups of users. These accounts are usually assigned to have limited access to the critical system files and directories.

- Group A/c 'A' (Privilege ...)

↳ Group A/c

A number of user accounts can be grouped together to create a group account. A Linux / UNIX group plays an important role in handling file permissions and process management.

GURUKUL

Page No.
Date: / /

Imp Past Qsn

* Mention user management commands

= There are four important directories for managing users & groups.

i) /etc/passwd
- Keeps the user account and password information. This file holds the majority of information about the accounts on the UNIX/Linux system

ii) /etc/shadow
- Holds the encrypted password of the corresponding account.

iii) /etc/groups
- This file contains the group information for each account.

iv) /etc/gshadow
- This file contains secure group account information.

Note:- We cat command to view the file system

(1) Creating an account
#useradd
- It is used to create a new user account.

#useradd <-option> <fieldname> <account name>

options
i) -d <directory_name>
- Specifies the home directory for the account.

default shell - bash bourne again shell

iii) -g <group-name> → group name must exist
- Specifies a group account for this account.

iiii) -s <shellname> : Ex -s //shellname -s /bin/bash
- Specifies the default shell name for this account.

ivj -u <user_id>
- You can specify a userId for this account.

v- <accountname>
- Actual accountname to be created.

vij -m :- Creates a new directory if it doesnot exist

Combined example:-

#useradd -d /home/devteam -g developers -m -s /bin/csh -u 101 heroalom

viij -e
Set the default expiration date after which the user account is disabled at midnight.
Date format : MM-DD-YY

(2) User Management (use #) / Modifying an account
#usermod.
- The user mode command enables you to make changes to an existing user account using the command line. It uses the same arguments as in the user-add command with the addition of -l argument/option, which allows you to change the username
Src :- tutorialspoint

GURUKUL

#usermod -d /home/user -s /bin/tcsh -e 10/12/2023 -g users -u 102 -l heroalom heroalom vaxo

(3) Deleting an account

#userdel

- We can use this command to delete a specific user account. We should be very cautious while using this command and it has limited options available.

options :-

a) -r

We can use this argument to remove the user account's home directory and mail file

#userdel -r heroalomvaxo

#passwd (affects /etc/passwd)

= Once account is created, you can set its password using the #passwd command. You must be logged in as a superuser.

Files owned by the deleted user but not located in the user's home directory will not be deleted. The system Admin (s.a.) must search for and delete those files manually. We can use $find command to do this.

Command :-

i) $find / -user heroalom vaxo

chown - change ownership
root.

Page No.
Date : / /

Search the entire file hierarchy starting at root (/) for all files and directories owned by the user and print the file names to the screen.

ii) $find /home -user hemalamvaxo -exec rm -id {} \;

Search for all files and subdirectories under /home owned by the user. The rm command to interactively delete each file owned by the user.

iii $find /home -users hemalamvaxo -exec chown vaxoNites {} \;

We can use the -exec to the parameter to run a command against each ~~file~~ matching file or directory.
The {} curly bracket characters designate where the matching files should be filled in.
The \ forward slash at the end simply tells Linux where the command ends.

Note:-
Remember any user added later on with the same UID/GID as the previous owner will now have access to this folder if you have not taken the necessary precautions. ~~To~~
(User ID / Group ID)
You may want to change these UID / GID values to something more appropriate, such as the root account, and perhaps even relocate the

GURUKUL

1st Assessment
groups

folder to avoid future conflicts.

⇒ Command to temporarily log or unlog an user account.

```
# passwd -l username      // log
# passwd -u username      // unlog
# passwd -l -e 10/12/2023 username
```

⇒ Simply disabling a user account will not prevent a user from logging in into your server, without removing any remotely. They'll still be able to gain shell access to the server without the need for any password.

⇒ Remember to check the user's home directory for files that will allow for this type of authenticated SSH access.

We should remove the given directory to prevent a user from logging in into your server.
/home/username/.ssh/authenticated_keys

⇒ Creating group account

We need to create groups before creating any account and all the groups are listed in /etc/groups. All the default groups are system account specific groups and it is not recommended to use them for ordinary accounts.

Syntax:-

```
# groupadd <-option> <groupaccount>
```

GURUKUL

Page No.
Date : / /

Options	Description
i) -g	Provide numerical value of the group's ID
ii) -o	This option permits to add groups with non-unique GID
iii) -r	This flag instructs group add to add a system account
iv -f	This option causes to just exit with success status

// the prev commands donot show results like user created, group created ... etc. So the -f option can be used for the success status

Not all options are important.

⇒ Modifying a group account
#groupadd -r

#groupmod
- We use groupmod command to change or modify a groups credentials.

New option	Description
-n	To change the name of group account

* Other ~~commands~~ options are same as in create user

Change Id and name :-
#groupadd -r -g 103 developers
⇒ #groupmod -g 104 -n devops developers

GURUKUL

4th term

⇒ Deleting a group
groupdel groupname
– Deletes a group of the groupname specified

⇒ Checking disk quotas
dudf

Limited disk space can be another source of user support calls. Red Hat Linux offers RHE(Enterprise) Linux & the quotas software package for limiting & displaying the amount of disk space that a user can consume. Generally, we use the du command to see how much disk space has been used in a particular directory and related sub directory

/etc/fstab

In ~~a case of~~ A careless or greedy user can use all the space on our hard disk and possibly bring our computer to a halt. By using disk quota, we can limit the amount of disk resources a user or a group can use up.
The quota package contain a set of tools that lets us limit the amount of disk space based on disk blocks and files based on inodes that a user can consume (Eg a user can create 10K file

//verbose user group vug
detail

Page No.
Date: / /

The general steps for setting disk quotas are:-

a) Creating quota files
b) Editing the /etc/fstab file
c Creating quota rules
d) Checking quotas
e

We set quotas on disk partitions listed in our /etc/fstab file. For computers that are shared by many users, there may be separate /home/var partition - where users are expected to put all their data. That kind of partition boundary can prevent an entire disk from being consumed by one user.

a. Creating ~~Edit~~ quota files
⇒ We need to have aquota.user and the / or aquota.group files in the root directory of the partition on which we want to establish disk quotas. To add quotas based on individuals or users, we need an aquota.user file. While aquota.group is needed to set quotas based on groups. One way to create these files is with the quota check command:-

```
#aquota -vug
```

A /home/user aquota.user file is created from the previous command. (To create an initial aquota.group file, type:-

#touch /home/aquota.group)

GURUKUL

hda21 – hard disk with 21 partition

1st Assessment

The quotacheck command in this example looks at the file system partition mounted on /home and builds a table of disk usage. The -v option produces verbose output from the command, the -u option causes user quotas to be examined and the -g cause the group quotas to be examined.

Note:- Permission on the two files are set so that only root can access them. ∴ # is used, superuser

b) Editing the /etc/fstab file

- We need to add quota support to the file system. To do that we need to edit the /etc/fstab file and add the usrquota option to field number 4 of the partition for which we want to set quotas in the hda21 first line

/dev/hda21 /home ext3 defaults, usrquota, grpquota 1 2

Here the /home file system is used to allow disk quotas for all users' home directories under the /home directory.

Note:- Before the usrquota option can take its effect, the file system must be remounted. (Reboot instead of Restart)

This happens automatically when we reboot or we might also use mount and unmount commands to cause userquota option to take/show effects.

c) Creating quotas rules :-
- If the quotas file doesn't include a startup script (current RHEL (Red Hat Enterprise Linux) does not included) we can create our own. We want this script to check quotas (#quotacheck) command start the quota service (#quotastart) command.

We can use the #addquota command to create quota rules for a particular user or group. The #addquota command uses the vi text editor to edit our quotas file.

d) Checking quotas
- To report on how much disk space and how many inodes each user in our computer has consumed, ~~we~~ use the #repquota command.

Note:- We can use du and df to see how much a user has consumed, ~~the use the #~~ the user has consumed the allocated quota or disk space.

Sending mail to all users:
Some times you need to send messages to all users in your system. For example warning user of planned downtime of hardware, rebooting the system or tackling a software vulnerability. It would be tedious to send individually the message to all the user and keeping the mailing list of all the users and keeping the mailing list of all the users on your system can

be problematic as the mailing list can change frequently.

foreach user (`awk -f: '{print $name}' /etc/passwd | tail +17`)
→ 17 line
echo mailing to user
mail → "The message's subject" $name < $filename → message.txt
(OR)
mailfile "The mail" message.txt
Sleep 2

The script accept two parameters the first is the email message which includes the quotes the second is the name of the file containing the text message to send.

Chapter 6

Security and System handling

Page No.
Date : / /

#Automating System task

It is extremely tedious to perform mundane tasks in Linux only using commands, rather it is suggested that tasks or commands that we regularly perform be grouped together and via means of shell programming are subjected to automatic execution by running shell script, Furthermore, it is more efficient to use shell scripts to do all these tasks.

A shell script is a group of commands, functions, variables or other objects that can be used within a shell. Shell scripts are used by the Linux System to ~~initiate~~ initialise system. Shell scripts follow the syntax based on the shell interpreter, that is BASH Shell's scripting syntax is different to that of C Shell (csh). (running multiple at same time)

Shell scripts are the equivalent of batch files in MS-DOS and can contain a long list of commands, complex flow control, complex arithmetic evaluations, user-defined variables, user defined functions, and sophisticated condition testing.

V. Imp in board

#Execution of shell scripts

Shell scripts can be opened and edited using any text editor (Vi editor or Pico)

However, one of the caveats (warning) of running shell scripts is that is slower to execute than compiled programs. So in order to run a shell script in RHEL, we have the following methods:-

1. The filename is used as an argument to the shell. For eg. (bash myscript.sh). Here, the shell script file with file extension (.sh) contains just a list of shell commands. The shell specified on the command line is used to interpret the commands in the script file.

2. The second methods is to have the name of the interpreter placed in the first line of script preceded by (#!/bin/bash). So after we have placed the name of interpreter, we need to set the execute bit of the script file to be set, to exemplify, we can use very helpful and important command (chmod +x Scriptname.sh) +x to execute

Best Practices while writing Shell scripts

It is important that certain scripts can be reused by other users too, therefore it is important to ensure that other users also understand the purpose of the script & its function

- Place an echo statement at the beginning of the lines within the body of a loop. That way rather than executing the code, you can see

⇒ variable to allocate space in memory.

Page No.
Date : / /

what will be executed without making any permanent changes.

- To achieve the same goal, you could place dummy echo statements throughout the code. If these lines get printed, you know the correct branch is being taken.

- You could use set -x near the beginning of the script to display each command that is executed or launch your scripts using bash -x myscript.

- Use proper use of comments using (#) to write single line comment in the script since the script can be long and complicated over time.

Understanding the shell variables

To reuse certain information, during the processing the shell script, the name or number representing this information may change. To store information used by shell script in such a way that is can be easily reused, one can set variable

Syntax:-

NAME = value # it follows general variable naming conventions only ("_", "0-9", "a to z or A to Z") can be used to enter value.

GURUKUL

Page No.
Date : / /

Example:-

CITY = "Spring Field"

PI = 3.1415

A variable can contain the output of a command or command sequence. One can accomplish this by preceding the command with a dollar sign and open parenthesis

MYDATE = $(date) #here today's date is transferred to MYDATE variable.

MYDATE = $`date` #here back tick can also have the same effects

A variable can contain the value of other variables.

curBalance = 1023120

BALANCE = "$curBalance" #1023120 has been transferred to BALANCE variable

We can make a variable read-only and we can unset a variable the following ways:

readonly curbalance #this variable is readonly, we cannot perform any arithmetic operation

unset curbalance # the value stored has been cleared.

GURUKUL

Page No.
Date : / /

Performing Arithmetic Operation

Bash uses uses untyped variables meaning it normally treats variables as strings or text. Integer arithmetic can be performed using the built-in let command or via the extended expression or bc commands

```
BIGNUM = 1024
Let RESULT = $ BIGNUM / 16
RESULT = `EXPR $BIGNUM / 16`
```

Conditional Statements
"If ... then" Statements

SYNTAX:-

```
Variable = 1 ;
if [ $variable - eq 1 ]; then
  echo "The variable is 1"
elif [ $variable -eq 1 ]
  echo "The variable is not 1"
else
  echo "It is something else"
fi endif used only in if else end (fi)
```

Note:- that [[is actually a command/program that returns either 0 (true) or 1 (FALSE). Any program that obeys the same logic (like ball base utils, such as grep (1) or ping (1)) can be used as condition.

GURUKUL

[[-z STRING]] Empty String
[[-n STRING]] Not empty String
[[STRING == STRING]] Equal
[[STRING != STRING]] Not equal
[[NUM -eq NUM]] Equal
[[NUM -ne NUM]] Not Equal
[[NUM -lt NUM]] less than
[[NUM -le NUM]] less than or equal
[[NUM -gt NUM]] Greater than
[[NUM -ge NUM]] Greater than or equal
[[STRING =~ STRING]] Regexp

Importance
Imp in board

Execution of Shell Scripts
- Shell scripts can be opened and edited using any text editor (vi editor or Pico).
- However, one of the

daemon :- background processes

Page No.
Date : / /

File Condition

((NUM < NUM4))	Numeric conditions
[[-e FILE]]	Exists
[[-f FILE]]	Readable
[[-h FILE]]	Symlink
[[-d FILE]]	Directory
[[-w FILE]]	Writable
[[-s FILE]]	Size is >0 bytes
[[-f FILE]]	File
[[-x FILE]]	Executable
[[FILE -nt FILE2]]	1 is more recent than 2
[[FILE -ot FILE2]]	2 is more recent than 1
[[FILE -ef FILE2]]	Same files

Defining Arrays

- It is a container object that holds a fixed number of values of a single type.

Fruits = ("Apple" 'Banana' 'Orange')

Fruits[0] = "Apple"
Fruits[1] = "Banana"
Fruits[2] = "Orange"

Working with arrays

```
echo ${Fruits[0]}      # Element #0
echo ${Fruits[-1]}     # Last element
echo ${Fruits[@]}      # All elements, space-separated
echo ${#Fruits[@]}     # Number of elements
echo ${#Fruits}        # String length of the 1st element
```

GURUKUL

Page No.
Date : / /

```
echo ${#Fruits[3]}  # String length of the Nth element
echo ${Fruits[@]:3:2}  # Range (from position 3, length 2)
echo ${!Fruits[@]}  # keys of all elements, space-separated
```

Looping Through An Arrays

```
for i in "${arrayName[@]}"; do
  echo $i
  done
```

Loops for, while and until

- In this section you'll find for, while and until loops.
- The for loop is a little bit different from other programming language. Basically, it let's you iterate over a series of "words" within a string.
- The while executes a piece of code if the control expression is true and only stops when it is false (or a explicit break is found within the executed code.
- The until loop is almost equal to the while loop, except that the code is executed while the control expression evaluates to false.

For sample :-

```
#!/bin/bash
  for i in $(ls); do
      echo item: $i
  done
```

GURUKUL

Page No.
Date : / /

While loop example

```
#!/bin/bash
COUNTER=0
while [ $COUNTER -lt 10 ]; do
    echo The counter is $COUNTER
    let COUNTER=COUNTER+1
done
```

On the second line, we declare i to be the variable that will take the different contained in $(ls).

8. Functions
- Declaring a function is just a matter of writing function my-func { my-code }

Functions sample

```
#!/bin/bash
function quit {
    exit
}
function hello {
    echo Hello!
}
hello
quit
echo foo
```

Function with parameters sample

```
#!/bin/bash
function quit {
    exit
}
function e {
    echo $1
}
e Hello
e World
quit
```

GURUKUL

Page No.
Date : / /

Run levels in Linux

A run level is a state of init and the whole system that defines what system services are operating.

Whatever Whenever a Linux system boots, finally the init process is started which is actually responsible for running other start scripts which mainly involves initialization of your hardware, bringing up the network, starting the graphical interface.

A runlevel in other words can be defined as a preset single digit integer for defining the operating state of your LINUX or UNIX-based operating system. Each runlevel designates a different system configuration and allows access to different combination of processes.

The standard LINUX kernel supports these seven different runlevels:-

0 - System halt i.e the system can be ~~easily~~ safely powered off with no activity
1 - Single user mode
2 - Multiple user mode with no NFS (Network File System)
3 - Multiple user mode under the command line interface
4 - User definable
5 - Multiple user mode under GUI (graphical user interface) and this is the standard runlev for most the LINUX based systems

GURUKUL

Page No.
Date : /

6 - Reboot which is used to restart the system.

By defaut, most of the LINUX based system boots to runlevel 3 or runlevel 5.

Runlevels 2 and 4 are used for user defined runlevels and runlevel 0 and 6 are used for halting & rebooting the system.

These start scripts corresponding to each run level can be found in special files present at under rc subdirectory

At rc.d directory there will be ~~directory~~ corresponding subdirectories named as such rc.0, rc.1, rc.2, rc.3, rc.4, rc.5, rc.6 or rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d, rc6.d.

For example, runscript or start script for run level will be under /etc/rc.d/rc.1 file

System startups & shutdown

During system startup, a series of scripts are run to start the services, namely Kudzu, system monitoring scripts, network interface scripts and so on. Mostly these run scripts are run from sub directories under /etc/rc.d

Starting run level scripts

The /etc/rc.d/rc script is integral to the

GURUKUL

concept of run level. It performs the following tasks:-

i) Checks that run level scripts are correct.
ii. Determines current and previous run levels; to discern which run level scripts to stop (previous) and start (current level)
iii) Decides whether to enter ~~into~~ interactive startup

If the confirm option is passed to the bootloader at boot time, all server processes must be confirmed ~~and~~ at the system console before starting.

iv) Kills and starts run level scripts
-Stops runlevel scripts from the previous level then starts run level scripts from the current level

⇒ Using THE cron Facility

- Cron is a scheduling daemon that executes tasks at specified intervals. These tasks are called cron jobs and are mostly used to automate system maintenance or administration tasks. Eg

- The cron jobs can be scheduled to run by the minute, hour, day of month, month, day of the week or any combination of these.

- The file /etc/cron.allow and /etc/cron.deny, the former contains the list of the users allowed to scheduled tasks and latter the list of user denied. If the /.deny directory is empty all users can gain access to cron facility.

At ; cron
crond :- cron daemon runs in bckgrnd always running

Page No.
Date : / /

These are a few of the advantages of using cron jobs:-

i) You have much more control over when your job runs i.e. you can control the minute, the hour, the day, etc. when it will execute.

ii) It eliminates the need to write the code for the looping and logic of the task and you can shut it off when you no longer need to execute the job.

iii) Jobs do not occupy your memory when not executing so you are able to save the memory allocation.

iv) If a job fails to execute and exits for some reason it will run again when the proper time comes.

⇒ There are four places where a job can be submitted for execution by the cron daemon crond :-

i) The /var/spod/cron/username file.
- This method, where each individual user (indicated by username) controls his or her own separate file is the method used on UNIX System V systems.

(ii) The /etc/crontab file. This is referred to as the system crontab file and was the original crontab file from BSD UNIX and its derivatives. Only root has permission to modify this file.

(iii) The /etc/cron.d directory. Files placed in this directory have the same format as the /etc/crontab file. Only root is permitted to create or modify files in this directory.

~~The /etc/cron.d directory. Files.~~

(iv) The /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly, and /etc/cron.monthly directories. Each file in these directories is a shell script that runs at the times specified in the /etc/crontab file (by default, at one minute after the hour; at 4:02 am everyday, Sunday at 4:22 am; and 4:22 a.m. on the first day of the month, respectively. Only root is allowed to create or modify files in ~~their~~ these directories.

Score Fast 8 mks

what is sys scheduling. Complete the following cron tab

Cron tab

C:At < cron < cron tab (most advance)

= So, the parts of a cron tab command are:-

i) The first five fields a b c d e specify the time/date and recurrence of the job.

ii) In the second section, the /directory/ command specifies the location and script

you want to run.

3. The final segment output is optional. It defines how the system notifies the user of the job completion.

Field Number	Field	Acceptable Values
1(a)	minute	Any integer between 0 and 59
2(b)	hour	Any integer between 0 and 23
3(c)	day of the month	Any integer between 0 and 31
4(d)	month	Any integer between 0 and 12 or an abbreviation for the name of the month (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec)
5(e)	day of the week	Any integer between 0 and 7 (where both 0 and 7 can represent Sunday) or abbreviation for the day (Sun, Mon, Tue, Wed, Thu, Fri, Sat)

An asterisk (*) in any field indicates all possible values for that field. For example, an asterisk in the second column is equivalent to 0, 1, 2 ... 22, 23, and an asterisk in the fourth column means Feb, Mar, ... Nov, Dec.

1st = 0 - 59 min ***** - Every min

4th = 0 - 11 mnth

crontab guru 28/2 - Every 2hr

Page No.
Date : / /

* An asterisk (*) stands for all values. Use this operator to keep tasks running during all months, or all days of the week.
* A comma (,) specifies separate individual values.
* A dash (-) ~~specifies sep~~ indicates a range of values.
* A forward-slash (/) is used to divide a value into steps (*/2 would be every other value, */3 would be every third, */10 would be every tenth, etc.).

Cron Syntax:-

- To be able to set up a cron job, we need to understand the basic elements that make up this syntax. The standard form for a crontab line is as follows:

a b c d e /directory/command output

Cron Job	Command
Run Cron Job Every Minute	* * * * * /root/backup.sh
Run Cron Job Every 30 Min	30 * * * * /root/backup.sh
Run Cron Job Every Hour	0 * * * * /root/backup.sh
Run Cron Every Day at Midnight	0 0 * * * /root/backup.sh
Run Cron Job at 2am Every Day	0 2 * * * /root/backup.sh
Run Cron Job at Every Day 1st of the Month	0 0 1 * * /root/backup.sh
Run Cron Job Every 15th of the Month	0 0 15 * * /root/backup.sh
Run Cron Job On December 1st	0 0 1 12 * /root/backup.sh

3 characters 3 3 3

r w x -
4 2 1 0

Page No.
Date : / /

Midnight

Run Cron Job on Saturday at Midnight — 0 0 * * 6 /root/backup.sh

At every 15th minute — */15 * * * * /root/backup.sh

At 4:00 on every day of week from Mon through Fri — 0 4 * * 1-5 /root/backup.sh

At minute 37 past every 2nd hour from 1 through 23 — 37 1-23/2 * * * /root/backup.sh

S.No ⇒ chMod Command

- Control who can access files, search directories & run scripts using the Linux's chmod command.

i — One set for owner of the file
ii — members of the group
iii — others

- r(i) w(i) x(i) r(ii) - x(ii) r(iii) - -
  7 @ 5 4

$ chmod 755 testfile
rwx r-x r-x
0 4 5

chown — change owner (chown) user1 (to) user2

chmod o+wx testfile # 777

#chown – change ownership
D– – – – directory
– 000 – file

chMod Command

The permissions control the actions that can be performed on the file or directory. They either permit, or prevent, a file from being read, modified or if it is a script or program, executed.

On each line, the first character identifies the type of entry that is being listed. If it is a dash (-) it is a file. If it is the letter 'd' it is a directory.

The next nine characters represents the settings for ~~the user who owns the file (user permissions).~~ the three sets of permissions.

- The first three characters shows the permissions for the user who owns the file (user permissions).

1. ls -l to show detailed list which also includes RWX permissions of files in that directory

2. chmod syntax
chmod (Hex value) filename

3. Illustrates of hex values

4 2 1	4 2 1	4 2 1
R W X	R W X	R W X
Owners Value	Group's Value	Others

- The middle three characters show the permissions for members of the file's (group group permissions)

- The last three characters show the permissions for anyone not in the first two categories (other permissions).

⇒ There are three characters in each set of permissions. The characters are indicators for the presence or absence of one of the permissions. They are either a dash (-) or a letter. If the character is a dash, it means that permission is not granted. If the character is an r, w or an x, that permission has been granted.

Exam

Backup

Types of Backup :- Full Backup - Space, Time, more space

Incremental »

How to transfer files from local to remote system or vice versa or local to local

Rsync - Remote Synchronization // command for backup

⇒ Backing up using rsync

rsync or remote synchronization is a software utility for ~~Unity~~ Unix-like systems that efficiently sync files and directories between two hosts or machines. One of them being the source or the local-host from which the files will be synced, the other one being the remote-host, on which synchronization will take place. There are basically two ways in which rsync can copy/sync data:-

GURUKUL

apt - advance package Tool
a repository with slw packages.

Page No.
Date: / /

- Copying / syncing to/from another host over any remote shell like ssh, rsh.
- Copying / Syncing through rsync daemon using TCP.

rsync Syntax:-

Local to Local: rsync [OPTION] [SRC] DEST
Local to Remote: rsync [OPTION] [SRC] [USER@] HOST:DEST
Remote to Local: rsync [OPTION] [USER@] HOST:SRC [DEST]

OPTION - The rsync options.

SRC - Source directory
DEST - Destination directory
USER - Remote username
~~BE~~HOST - Remote hostname or IP Address

- v: verbose
- -r: copies data recursively (but don't preserve timestamps and permissions while transferring data. compress
- -a: archive mode, which allows copying files recursively and it also preserves symbolic links, file permissions, user & group ownerships, and timestamps
- -z: compress file data
- -h: human-readable, output numbers in a human-readable format.

GURUKUL

Rsync, sudo are prgrm accessed in command

Page No.
Date: / /

Installing Rsync
- The rsync utility is pre-installed on most Linux distributions and macOS. If you don't have rsync installed on your system, you can easily install it using your distribution's package manager

Install Rsync on Ubuntu and Debian
sudo apt install rsync

Install Rsync on CentOS and Fedora
sudo yum install rsync

The first example shows a simple backup of a user's personal files. Here, the command

cont..

⇒ Example of local to local sync:-

```
# rsync -av /home/chris /mnt/backup/homes
      option   src        destination
```

continue....
is copying the /home/chris directory (including all its files and subdirectories) to another directory on the local computer. That directory (/mnt/backup/homes) could be on a separate partition for creating separate partition hard disk or a remote NFS file system.

rsync will copy files from that directory to a target directory named chris (/mnt/backup/home/chris). With a trailing slash all files from /home/christ are copied directly to the homes directory (/mnt/backup/homes/).

GURUKUL

Page No.
Date : / /

BACKING UP FILES REMOTELY

- While the previous example was a quick, informal backup method, with more critical data, you want to make sure that the data are being backed upto another computer and that the backup is done at regular intervals. This can be accomplished by using rsync in combination with ssh and cron.
- Having ssh as the transport mechanism ensures that data will be encrypted when it is transferred also, because the ssh service (sshd) is enabled by default on many Fedora and Red Hat Linux systems, you need only a username and password to the remote system to do the backup. (As long as you can use ssh to connect to the remote machine and rsync is installed remotely, you can use the rsync command to transfer files there). Here's an example.

```
# rsync -a2v -e ssh /home/chris duck
:/mnt/backup/homes root@duck's password:
********** building file list ... done/
```

Here, I identify the remote computer (named duck in this case) by putting it before the remote directory name, separated with a colon. I use some different options as well. To the archive (-a) and verbose (-v) options, I add -z option to compress the data (making it more efficient to transfer). I also use the -e ssh option to have rsync use a ssh remote shell to transfer data. The password prompt you see is the ssh login prompt.

File pe
Page No.
Date : / /
*Chapt-6 complete

You can repeat this command each time you want to backup your files. However, the more efficient way to do this is to set up this command to run as a cron job so that the backups happen automatically at set intervals.

To have rsync run automatically, you can't have it prompt you for a password. To have the rsync command run without prompting for a password, follow this procedure:

1. Set up ssh to do no-password login for the user who is going to perform the backup (see the section "Using ssh, scp and sftp without passwords").
2. Decide how often you want the backup to run. For ex, if you want to run the rsync command once each day, as root user you could create a file called /etc/cron.daily/mybackup
3. Set permissions to be executable

```
# chmod 755 /etc/cron.daily/mybackup
```

4. Add the command line to the mybackup file that you want to use

```
rsyn -a2v -e ssh /home/chris chris@duck:/mnt/backup/homes/
```

Chapter 7

SETTING UP A WEB SERVER

Page No.
Date: / /

Assignment

i) What are HTTP REQUEST and HTTP Response

- An HTTP request is messages sent by the client to initiate an action on the server.
- The aim of the request is to access a resource on the server.

HTTP request methods

- GET, HEAD, POST, PUT (like update), DELETE, CONNECT, TRACE, OPTIONS, etc

An HTTP response is message sent by the server to the client.

- The aim of the response is to provide client with the resource it requested or inform the client that the action it requested has been carried out. or else to inform the client that an error occurred in processing its request.

- It has/contains
  i. Status line
  ii) Response Header Fields / series of HTTP header
  iii) Message Body

i) Status line has
  a) HTTP Version Number
  b) Status Code
  c) Reason Phrase

1XX : Information (Processing request
2XX : Success
3XX : Redirection
4XX : Client Error
5XX : Server Error

GURUKUL

Impt:- What is Web Server? How to configure it?

Page No.
Date: / /

⇒ Installation of Apache Server

⇒ Introduction to web server.

- A web server is a system that delivers content or services and users over the internet. The development of world wide web server dates to the time when Tim Berners Lee began a project at CERN. Then, the concept of using one client (Web Browser) to access info / data (Multimedia) from several servers (FTP, HTTP, SMTP
- A web server is a computer or combination of computer, which is connected through internet or intranet to serve the clients' requests coming from their web browser. Every web server has a unique IP address & domain name which identifies that matching on the N/w.

- The collection of all our web pages is a website.
- To let others view our webpages, we must publish our website.
- To publish our work, we must copy our site to a web server.
- Our own PC can act as a web server if it is connected to a network.
- Most common is to use an Internet Service Provider (ISP).

The web server usually has a simple job: to accept Hypertext Transfer Protocol (HTTP) requests and send a response to the client. However this job can get much complex (as the server

GURUKUL

Page No.
Date : / /

can also, executing functions such as:-

- Performs access control based on file permissions, username/password pairs and host name IP address restriction
- Parsing a document (substituting appropriate values for any conditional fields within the document) before sending it to the client.
- Spawning a Common Gateway Interface (CGI) script or custom Application Programming Interface (API) program to evaluate the contents of a submitted form, presenting a dynamically created document, or according a database.
- Logging any ~~database~~ successful access, failures and errors.

⇒ APACHE Installation

There are several ways to install RHEL/CENTOS Machine, they are:-

1) USING RPM

- httpd - 2.4.4 - 1.x86_64.rpm
- mod_ssl - 2.4.4 - 1.x86_64.rpm

httpd - 2.4.4 - 1.x86-64.rpm is a base package, which will install Apache input 2.4.4.
mod_ssl - 2.4.4 - 1.x86_64.rpm is mod_ssl.so file, which will be installed under the module directory.

ivh - verbose
ssl - secure socket layer
apachectl - control

Page No.
Date : / /

Install Apache 2.4.4

rpm -ivh httpd-2.4.4_1.x86-64.rpm

If you require SSL enabled on your web server you must install mod_ssl.

rpm -ivh mod_ssl-2.4.4-1.x86_64.rpm

⇒ What and where it will install?
- Impt directories in APACHE
httpd.conf - /etc/httpd/conf/httpd.conf
apachectl - /usr/sbin/apachectl Chas log

Continue 1st para:-
SMTP, Gopher, NNTP & so on. A web server, server operating system (OS) and software used to facilitate HTTP communication.

As always, make a backup copy of the main configuration files before editing it:

cp /etc/httpd.conf /etc/httpd/conf/httpd.conf $(date +%Y%m%d)
Then open it with your preferred text editor and look for the following variables: -

- ServerRoot : the directory where the server's configuration, error and log files are kept.

GURUKUL

Page No.
Date : / /

- Listen: instructs Apache to listen on specific IP addres and/or ports.
- Include: allows the inclusion of other configuration files, which must exist. Otherwise, the server will fail, as opposed to the Include Optional directive, which is silently ignored if the specified configuration files do not exist
- User and Group: the name of the user/group to run the httpd service as.
- DocumentRoot :- The directory out of which Apache will serve your documents. By default, all requests are taken from taken from this directory, but symbolic links and aliases may be used to point to other locations
- ServerName: this directive sets the hostname (or IP address) and port that the server uses to identify itself.
- The first security measure will consist of creating a dedicated user and ~~for~~ group (i.e shiva/shivad) to run the web server as & changing the default port to a higher one (9000 & in this case).

ServerRoot "/etc/httpd"
Listen 192.168.0.1.8:9000 (default=80)
User ~~Shiva~~ user1
Group Admins
DocumentRoot "var/www/html"
ServerName 192.168.0.18:9000

GURUKUL

You can test the configuration file with

#apachectl configtest

and if everything is OK, then restart the web server.

#systemctl restart httpd

and don't forget to enable the new port (and disable the old one) in the firewall:

- The primary Apache configuration file in /etc/httpd/conf/httpd.conf. It contains a lot of configuration statements that don't need to be changed for a basic installation. In fact, only a few changes must be made to this file to get a basic website up and running.
- First, browse through the httpd.conf file to familiarize yourself with it. Red Hat versions of most configuration files is the number of comments that describes the various sections & configuration directives in the files. The httpd.conf file is no exception, as it is quite well commented. Use these comments to understand what the file is configuring.
- The first item to change is the Listen statement which defines the IP address and port on which Apache is to listen for page requests.

Listen 127.0.0.1:80

In linux, html instead of htdocs folder.

Page No.
Date : / /

With this directive set to the IP address of the localhost, Apache will listen only for connections from the local host. If you want the web server to listen for connection from remote hosts, you would use the host's external IP address.
- The DocumentRoot directive specifies the location of the HTML files that make up the pages of the website. That line does not need to be changed because it already points to the standard location. The line should look like this:-

DocumentRoot "/var/www/html"

The Apache installation RPM creates the /var/www directory tree. If you wanted to change the location where the website files are stored, this configuration item is used to do that. For example, you might want to use a different name for the www subdirectory to make the identification of the website more explicit. That might look like this:-

DocumentRoot "/var/mywebsite/html"

These are the only Apache configuration changes needed to create a simple website. For this little excercise, only one changes was made to the httpd.conf file — the Listen directive. Everything else is already configured to produce a working web server.

gpt, mbr

Page No.
Date : / /

One other change is needed, however, opening port 80 in our firewall. Use iptables as firewall, so I change /etc/sysconfig/iptables to add a statement that allows HTTP protocol. The entire file looks like this:-

~~# sample configuration for ipta~~
// to add or change
-A INPUT -p tcp -m state -state NEW -m tcp -dport 9000 -j ACCEPT

Chapt 8

DHCP and NIS

Page No.
Date : / /

DHCP

- DHCP based on TCP due to double Handshaking.

Setting up a Dynamic Host Configuration Protocol (DHCP) server enables you to centrally manage the address and other network client computers on your private network. With DHCP configured on your network, a client computer can simply indicate it wants to use DHCP & the DHCP server can provide its IP address, network mask, DNS server, NetBIOS server, router (gateway) and other information needed to get up & running on the network.

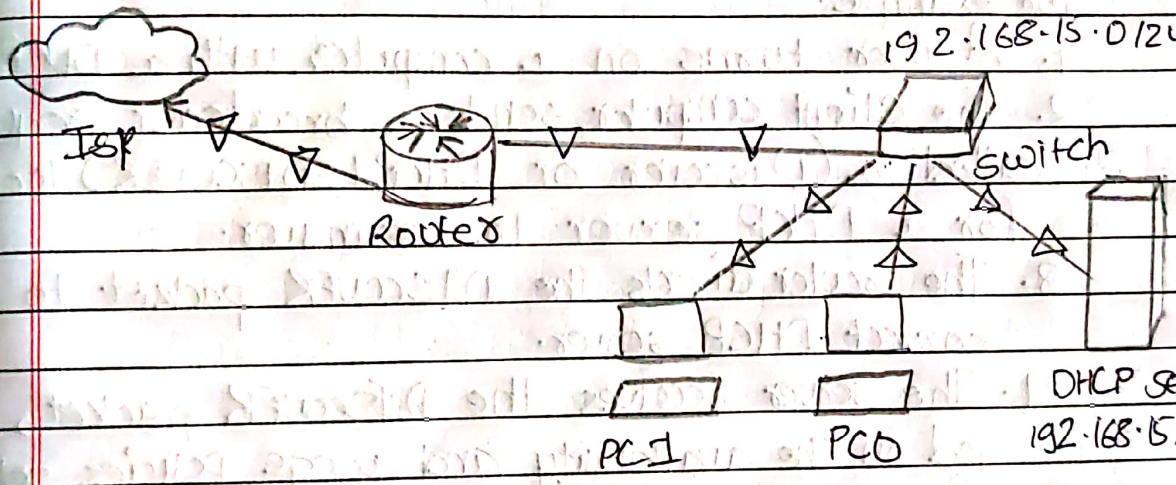
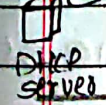


Fig :-

⇒ DORA (Discover, Offer, Request and Acknowlege) process



Client : Hi will you please give me IP addr

Discover → DHCP server

Server :- Sure I have xyzw IP for you. Will you accept it

offer ← DHCP server

Client : Yes, I accept Kindly issue x.y.z.w. IP to me & send me remaining config

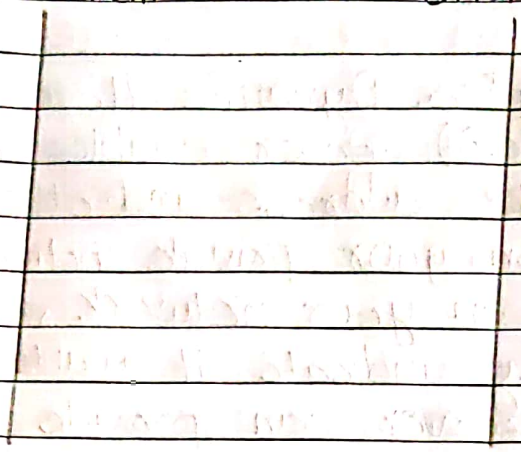
Request → DHCP server

Server : OK xyzw IP is assigned to you.

DHCP ack ← DHCP server

Client

Page No.
Date : / /



DHCP assigns and IP address when a system is started for example.

1. A user turns on a computer with a DHCP client.
2. The client computer sends a broadcast request called a (Discover or DHCPISDISCOVER) looking for a DHCP server to ~~se~~answer.
3. The router directs the DISCOVER packet to the correct DHCP server.
4. The server receives the DISCOVER packet, based on the availability and usage policies set on the server such that the server determines & appropriate address (if any) to give to the client. The server then temporarily reserves that address for the client and sends back to the client an OFFER (or DHCPOFFER) packet, with that address information. The server also configures the client's DNS servers WINS server, NTP servers and sometimes other services as well.

5. The client sends a REQUEST (or DHCP RE-QUEST) packet, letting the server know that it intends to use the address.
6. The server sends an ACK (or DHCPACK) packet, confirming that the client has been given a lease on the address for a server-specified period.