OS

L I N U X

COMMAND LINE INTERFACE

unix

COMMANDS

Command line interface

01. BASIC COMMANDS

02. MANAGEMENT

03. PERMISSIONS

04. NETWORKING

05. VOLUME

06.SHELL , BASH SCRIPT

One

1 BASIC COMMANDS

01. Basic Command

1. pwd	-- Print Working Directory.
2. mkdir dir-name	-- create directories(folders).
3. mkdir dir1 dir2	-- create multiple dirtory
4. mkdir my-dir{1..5}	--create the range of dir’s
5. mkdir -p demo/de2/de3/de4	-- to create dir within their parent dir
6. mkdir .my-dir	-- to create the hidden dirtory
7. mkdir .dir1 .dir2	-- to craete multiple hidden dir
8. mkdir .my-dir{1..5}	-- create the range of hidden dir’s
9. rmdir dir-name	-- delete the particular dir
10. rm -rf dir-name	-- force fully delete dir with files

01. Basic Command

1. cd dir-name -- to change the dir to particular dir with dir-name
2. cd .. --one step back dirtory
3. cd ../.. -- two step back dirtory
4. cd --change current user home dirtory
5. cd ~ -- change current user home dirtory
6. cd / --- change the "/ " dirtory
7. cp -R <source path> <destination path> --- copy the dir one place to another place
8. cp -R /home/ubuntu/dir /home/sunil/ -- copy the dirtory one place to another
9. touch file.txt -- craete empty file
10. touch .file.txt -- create hidden empty files

01. Basic Command

1. ls	-- to list all normal files and dir
2. ls -a	-- to list all the hidden and normal files
3. ls -l	-- to list all normal files with their permission
4. ls -al	-- to list the permissions of all hidden and normal file & dir's
5. ls -l file-name	-- to list the permissions of particular file
6. ls -ld dir-name	--to list the permissions of particular dir
7. ls -i	-- to list the inode number of a file and dir's
8. ls -i file-name	-- to check particular file inode number
9. ls -id	-- to check particular dir's inode number
10. ls -lrt	-- to list all the files and dir's based on their date & time

02. Basic Commands

1. timedatectl	-- to check the date and time with time zone
2. sudo timedatectl list-timezones s	-- to list the all time zones
3. sudo timedatectl set-timezone Asia/Kolkata	-- to set the timezone IST
4. sudo timedatectl set-time 2020-05-25	-- to change the date [yyyy-mm-dd]
5. sudo timedatectl set-ntp false	-- disable the automatic time and date
6. sudo timedatectl set-ntp true	-- enable the autmatic time and date
7. sudo timedatectl set-time 10:42:43	-- to change the time [hh:mm:ss]

03. vi / vim [visual editor & visual editor monitor]

1. vi
2. nano
3. vim
4. cat

vi	-- editor
1. i	-- insert mode(to edit mode the file)
2. Esc	-- used for exit from editor mode (chage to execute mode)
3. :wq	-- to save and quit the file
4. :q!	-- to quit the file without saving forcefully
5. :Set number	-- to check the number of lines in file

04. Linux editors

1.	dd	-- delete the line
2.	yy	-- copy the line
3.	p	-- past the line
4.	u	-- undo
5.	:set hlsearch	-- Highlight the searching word
6.	:s/word ,?word	-- search the word
7.	:%s/arun/magi	-- search and repace the word
8.	rm filename	-- delete the file
9.	cp source dest	-- copy the file(/home/ubuntu/file.txt /home/ubuntu/demo/test/)
10.	mv filename new-name	-- change the name of file and move one to another place

TWO

2 MANAGEMENT

01. user and group management

Types of users:-

1) super user (root)

2) normal user

3) sudo user

1. sudo adduser username -- create the user with home dir and primary group
2. sudo useradd username -- create user with home dir
3. sudo userdel username -- delete the user
4. sudo cat /etc/passwd -- check the users
5. sudo cat /etc/shadow -- to check encripted password
6. sudo passwd username -- modify or set-up password for user
7. usermod -l login-name old-name -- change the username
8. users have UID (user ID) for the kernel referance & username is for our referance

Types of groups:

1. primary group --> the default group created while a user with the name as its username
2. secondary group --> except primary group if a user is present any other group is called as secondary group

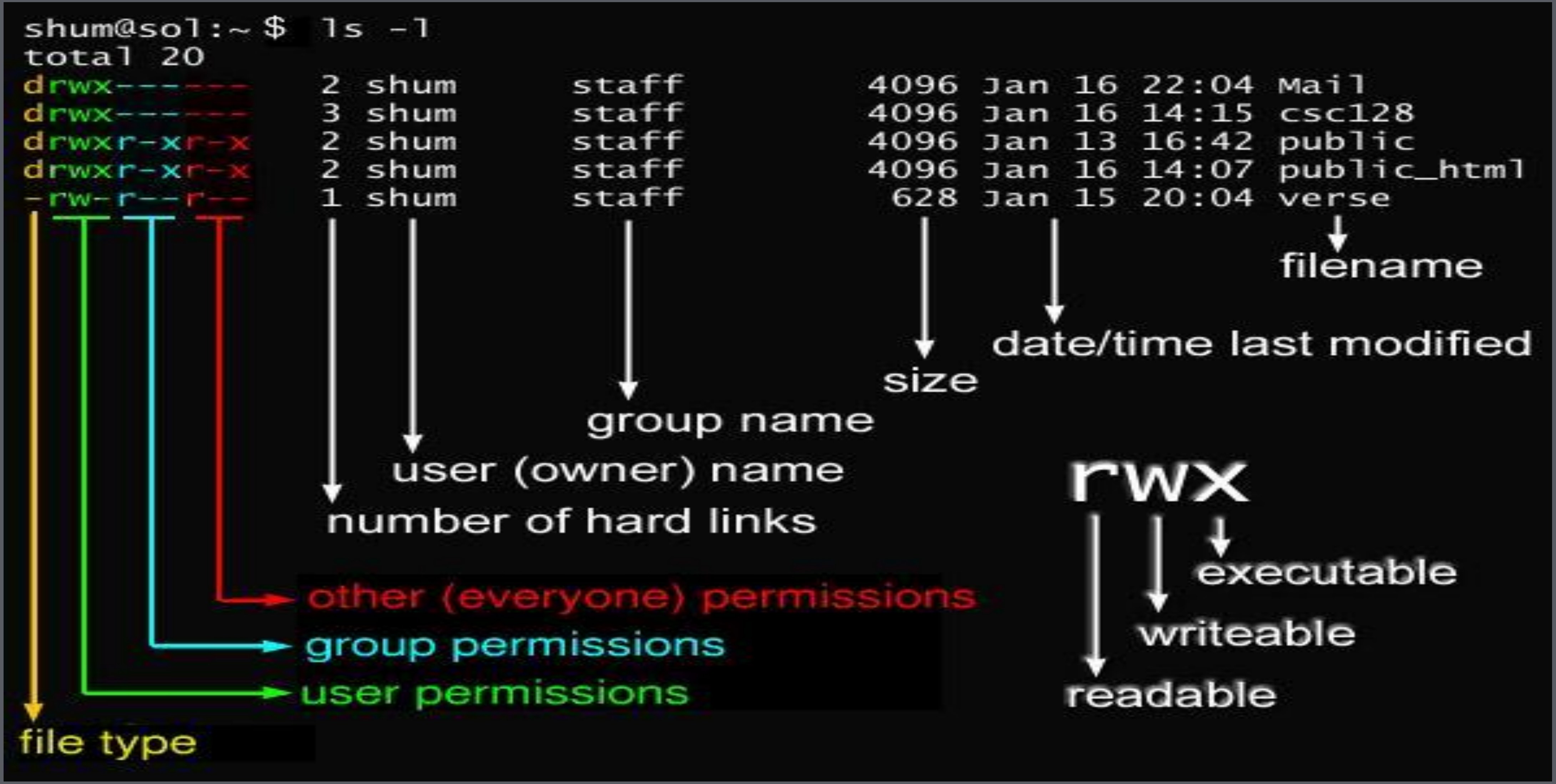
1. `sudo groupadd groupname`	-- create the group
2. `sudo groupdel groupname`	-- delete the group
3. `groupmod -n <new groupname> <old groupname>`	-- change group name
4. `sudo cat /etc/group`	-- check the group and group users
5. `gpasswd <group name>`	-- set-up password for group
6. `vim /etc/gshadow`	--to verify the password in encrypted format

1. gpasswd -a <secoundry group> <username> --> user add to secoundry group
2. usermod -G <secoundry group> <username> --> user add to secoundry group
3. usermod -a -G <secoundry group> <username> --> one user add to multiple group
4. gpasswd -d <username> <group name> --> deleting user from group
5. gpasswd -M <username1>,<username2> <group name> --> to add multiple users into the group and it also removes the previously present users from group

THREE

3 PERMISSIONS

03. USER AND GROUP PERMISSION FOR FILES



```
shum@sol:~ $  ls -l
total  20
drwx------   2 shum     staff       4096 Jan 16 22:04 Mail
drwx------   3 shum     staff       4096 Jan 16 14:15 csc128
drwxr-xr-x   2 shum     staff       4096 Jan 13 16:42 public
drwxr-xr-x   2 shum     staff       4096 Jan 16 14:07 public_html
-rw-r--r--   1 shum     staff        628 Jan 15 20:04 verse
```

03. user , group management and permission

	Read	Write	Execute
	4	2	1
0	-	-	-
1	-	-	X
2	-	W	-
3	-	W	X
4	R	-	-
5	R	-	X
6	R	W	-
7	R	W	X

1. sudo chmod 764 filename -- file permission for user and group
2. sudo chown user:group filename -- user permission for user and group

FILE

- default permission ===> root user ==> 644 (rw-r--r--)
- default permission ===> normal user ==> 664 (rw-rw-r--)
- max permission of file is ==> 666 (rw-rw-rw-)
- max permission of executable file ===> 777 (rwxrwxrwx)

DIRECTORY

- default permission ==> root user ==> 755 (rwxr-xr-x)
- default permission ==> normal user ==> 775 (rwxrwxr-x)
- max permission of dir is ==> 777 (rwxrwxrwx)

02. file management

1. sudo find / -type f -name file-name	-- to find the file path
2. sudo find / -type d -name dir-name	-- to find the dirtory path
3. sudo find / -type f -name ‘test* ‘	-- find the test starting all files
4. sudo find / - type f -neme ‘*.txt’	-- find the files ending with .txt all files
5. touch file-name	-- create the empty files
6. echo “hello world ”	-- print the word
7. free	-- to check the memory (ram)
8. free -m	-- to check the memory usage
9. nproc	-- to check vcpu on server
10. du -h file-name	-- memory utilaztion for file

02. file management

1. df	--display the available file system info in blocks and disk's
2. df -h	-- display all the FS info in human readable format
3. df -k	-- display FS info in KB's
4. df -m	-- display FS info in MB's
5. df -T	-- display FS info with its types
6. df -Th	--display FS info in human readable format with filesystem type

03. file management

1. top -- to check the real-time running process and cpu and memory usage.

- PID -- Shows task’s unique process id.
- PR -- The process’s priority. The lower the number, the higher the priority.
- VIRT -- Total virtual memory used by the task.
- USER -- User name of owner of task.
- %CPU -- Represents the CPU usage.
- %MEM --Shows the Memory usage of task.
- COMMAND -- The name of the command that started the process. \

1. top -n 10 -- top 10 cpu and utlization process
2. top -u root -- to check the spcific user process

Four

4

GENREL COMMAND

Click here to add content of the text.

01. Regular Linux Commands

1. tar -cvf file.tar file1.txt file2.txt -- to compress two into one file
2. tar -cvf file.tar “file*” -- create the tar (zip) file or compressed file
3. tar -cvzf file.tar “file*” -- create extra commpressed file [10 gb -->2gb]
4. tar -xvf file.tar -- extract the tar file
5. tar -xvf file.tar -C dir-name -- extract the file to spcific folder
6. zip test.zip file1.txt file2.txt -- zip the files
7. zip test.zip file* -- zip the all files starting with files
8. unzip test.zip -- extrac the file in current dirtory
9. unzip test.zip -d /home/ubuntu -- diexterc the file in another dirtory
10. head -n 4 > output.txt -- output copy to new file

01. Regular Linux Commands

1. head file-name	-- display the first 10 lines of file
2. head -n 30 file-name	-- display the first 30 lines of file
3. head -c 200 file-name	-- display the first 200 characters
4. head -v file-name	-- display first 10 lines and output with verbose
5. head -n 4 > output.txt	-- output copy to new file
6. tail file-name	-- display the last 10 lines of file
7. tail -n 30 file-name	-- display the last 30 lines of file
8. tail -c 200 file-name	-- display the last 200 characters
9. tail -v file-name	-- display the last 10 line with output with verbose
10. tail -n 4 > output.txt	-- display the last 4 and output copy to new file

02. Regular Linux Commands

1. cat file-name	-- display the content of file
2. cat -n file-name	-- display the content with number lines in file
3. cat > file-name	-- create the new file[ctrl+d -save the content]
4. cat >> file-name	-- add the new line into bottom of already excited file
5. cat file1 file2 file3 > file4.txt	-- copy the content of 3 files to file4.txt
6. > file.txt	-- to dalete content

03. Regular Linux Commands

1. tac file-name	-- display the content reversely
2. more file-name	-- more command displays text, one screen at a time
3. more -s file-name	-- to delete the gap between two lines.
4. more -5 file-name	-- view first 5 lines
5. more +5 file-name	-- view after 5 th line
6. less file-name	-- less command displays text, one screen at a time
7. less -X file-name	-- Keep content on screen after quitting
8. less -5 file-name	-- forward and backward not suppoted
9. grep word file-name	-- Search the word from file with full line
10. grep -o word file-name	-- Search only work without full line

05. Reguler Linux Command

1. grep -v word file-name	-- Search the without word all other lines.
2. grep -r word * .	-- Search word from all files in current dirtory
3. grep -i word file-name	-- Search the work without case sensitivity
4. grep -c word file-name	-- count the word no.of time present in line
5. grep -n word file-name	--Search the word and line number that containe
6. grep -w word file-name	-- Search for exact matching word using the -w option
7. grep --color word file-name	-- Search word with color
8. grep -A2 word file-name	-- display the line after the result.
9. grep -B2 word file-name	-- display the line before the result.
10. grep -C2 word file-name	-- display the line after and line before the result.

04. Reguler Linux Command

1. cut -b 1,2,3 file-name	-- Search the first three bytes in all rows.
2. cut -b 1-3,6-8 file-name	-- Search the first three and next 5 to 7 bytes all rows.
3. cut -b 2- file-name	-- Search the after 2 bytes in all rows.
4. cut -b -3 file-name	-- Search the first three bytes in all rows.
5. cut -f 2 file-name	-- Search the 2 field in table.[if sprated with tab]
6. cut -f 1,3 file-name	-- Search the 1 and 3 field in table.[if sprated with tab]
7. cut -d “ ” -f 2,3 file-name	-- Search the 2 and 3 field.[if sprated with -,:,;]
8. cut --complement -f 2 file-name	-- Search the without 2 field.
9. cut --output-delimiter="_" -f 1,2 file-name	-- Search the output with underscore.
10. cut -c 1,2,3 file-name	-- Search the first three characters lines.

04. Reguler Linux Command

1. sort file-name	-- sort used for arranging the records in a particular order.
2. sort filename > new-file-name	-- arraged file copy to another file.
3. sort -r file-name	-- Reverse-oder sort the file.
4. sort -n file-name	-- Numer sorting .
5. sort -nr file-name	-- Reverse number sorting.
6. sort -k 2 file-name	--sort the secound column if that numbers.
7. sort -c file-name	-- check the file already sorted or not .
8. sort -u file-name	-- sort and remove the duplicates.
9. sort -M file-name	-- sort by month.
10. ls -l | sort -nk 5	-- sort the files and dirtorys in date and time.

04. Reguler Linux Command

1. comm file1 file2	-- compare the two sorted files.
2. comm -1 file1 file2	-- compare two files and suppers lines unique to file1.
3. comm -2 file1 file2	-- compare two files and suppers lines unique to file2.
4. comm -3 file1 file2	--compare two files and suppers lines both file1 file2.
5. comm --check-order file1 file2	-- compare the file with order.
6. comm --nocheck-order file1 file2	--compare the files without order.
7. uniq file-name	-- to check file without duplicates .
8. uniq -c file-name	-- to count the duplicates .
9. uniq -d file-name	-- to check only duplicated words.
10. uniq -u file-name	--to check only uniq words.

04. Reguler Linux Command

1. sed ‘s/sad/happy/’ file-name -- find and replace new word in every line first occre.
2. sed ‘s/sad/hay/g’ file-name -- find and replace every where on file.
3. sed ‘s/sad/hay/3’ file-name -- replacing the nth occurrence of in same line a file.
4. sed ‘ 4 s/sad/hay/’ file-name -- replacing the word in only fourth line of file.
5. sed -n ‘ 4 s/sad/hay/p’ file-name -- print only replaced line of file.
6. sed ‘ y/sd/no/’ file-name -- find and replace the character.
7. sed ‘ y/sa/no/’ file-name > new-file -- find and replace the character and save to new.
8. sed ‘3d’ file-name -- delete the 3rd line on file
9. sed ‘1,3d’ file-name --delete the line range
10. sed ‘/unix/d’ file-name -- delete the line where ever unix is there.

04. Reguler Linux Command

1. cal | tee file-name	-- save the calender in file
2. date | tee -a file-name	--date also save exsiting file .(append the line in file)
3. cal | tee file1 file2 file3	-- calender save to multiple files.
4. time linux-command	--to check the time take process to complete.
5. ps -aux	-- list the all process running in server.
6. ps -aux | grep sleep	-- to process id of sleep process
7. kill pid	-- kill the process
8. kill -9 pid	-- fource fully delete the process
9. apt install ncal	-- to install calender in sever

04. Reguler Linux Command

1. cat filename | tr ‘a-z’ ‘A-Z’	-- to change all lower case into upper case.
2. cat filename | tr [:lower:] [:upper:]	-- to change all lower case into upper case.
3. cat filename | tr -d character	-- to delete the word in file.
4. wc file-name	-- to count the no.of line,no.of words and no.of charater in file.
5. wc -l file-name	-- count only no.of lines in file.
6. wc -c file-name	-- count only no.of charters in file.
7. wc -w file-name	-- count only no.of words in file.
8. sleep 10s	--server will sleep 10s
9. cal;date	-- display the cal and date at a same time
10. # commad	-- # unused command

04. Reguler Linux Command

1. cat filename1 && cat filename2 -- if first command success after that only next will run.
2. cat filename1 || cat filename2 -- if first command will fail but secound will run.
3. history -- to check the history of excuted commands.
4. history -c -- to clear the history
5. history 10 -- to check the history of excuted last 10 commands.
6. ln sorce-name destiname -- copy the file and live connection between files.
7. ln -s source-name dest-name -- make the soft link for file path
8. apt install -- Advanced Packaging Tool (ubntu,debian)
9. yum install -- Red Hat Package Manager (RPM)[Fedora, CentOS, RHEL, etc.]

FIVE

5 NETWORKING

05. Networking Commands

1. ifconfig (ip a) -- to check the privte ip address
2. curl ifconfig.me -- check the public ip address
3. nslookup google.com -- to check the ip for domin name
4. traceroute amazon.com -- to check the gateways to reach the website
5. telnet ip port -- service is connected or not
6. netstat -tulpn -- to check local connect(Active Internet connections (only servers)
7. ping google.com -- to check the internet on server
8. ping -c 10 google.com -- ping the 10 times
9. curl url -- search the link
10. wget url -- to download the link file

05. Networking Commands

1. apt install firewalld
2. firewall-cmd --zone=public --add-port=80/tcp --permanent
3. firewall-cmd --reload
4. firewall-cmd --list-ports
5. systemctl status firewalld
6. firewall-cmd --zone=public --remove-port=80/tcp --permanent

05. shortcut command

1. [root@172.22.32.34]#	----> root
2. [dev-user@172.22.32.34]$	----> normal user
3. [dev-user@172.22.33.22 ~]$	--> perticuler user home dirtory
4. hostname	--> to check hostname
5. hostnamectl set-hostname	--> to set-up new hostname
6. reboot	--> restart the server
7. poweroff	--> stop the server
8. touch file{1..5}.txt	---> create the range of files
9. vi .text.txt	---> create the hidden file

06. Reguler Linux Command

Hard Link :

- A hard link acts as a copy (mirrored) of the selected file. It accesses the data available in the original file.
- If the earlier selected file is deleted, the hard link to the file will still contain the data of that file.

Soft Link :

- A soft link (also known as Symbolic link) acts as a pointer or a reference to the file name. It does not access the data available in the original file.
- If the earlier file is deleted, the soft link will be pointing to a file that does not exist anymore.

07. Reguler Linux Command

Absolute Path

- An Absolute Path is a full path specifying the location of a file or directory from the root directory or start of the actual filesystem.
- Example: /home/javatpoint/Desktop/CollegeStudent

Relative Path

- The relative path of a file is its location relative to the current working directory. It never starts with a slash (/). It begins with the ongoing work directory.
- Single Dot (.) resolves to the current directory.
- Double Dot (..) resolves to the parent directory of the present work directory.
- Tilde (~) represents the home directory of logged in user.

Disk Partitioning

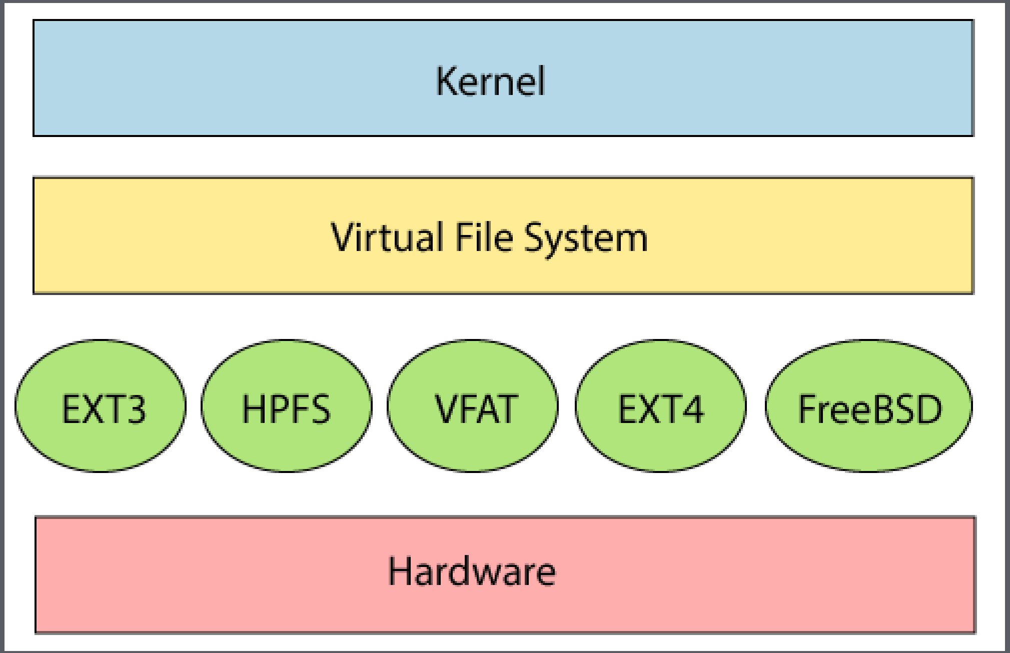
- Disk Partitioning is the process of dividing a disk into one or more logical areas, often known as partitions, on which the user can work separately.
- It is one step of disk formatting.
- If a partition is created, the disk will store the information about the location and size of partitions in the partition table.

Why we need it?

- To upgrade Hard Disk (to incorporate new Hard Disk to the system)
- Dual Booting (Multiple Operating Systems on the same system)
- Efficient disk management
- Ensure backup and security
- Work with different File Systems using the same system

Linux File System

- A Linux file system is a structured collection of files on a disk drive or a partition. A partition is a segment of memory and contains some specific data.
- In our machine, there can be various partitions of the memory.
- Generally, every partition contains a file system.
- The general-purpose computer system needs to store data systematically so that we can easily access the files in less time.



09. volume mounting

1. sudo lsblk -- to check file partion.
2. sudo fdisk /dev/xvdf -- to create the partion
3. sudo file -s /dev/xvdf -- to check file system
4. sudo mkfs -t ext4 /dev/xvdf -- to create the file system for disk
5. sudo mount /dev/xvdf /home/ubuntu/dir-name -- mount the volume
6. sudo umount /home/ubuntu/dir -- unmount the disk
7. sudo df -h -- to check mounted disks

src-server dest-server

src-server

```
apt update
	apt install nfs-kernel-server
sysemctl restart nfs-kernel-server
vi /etc/exports
/mnt/nfs_share 10.0.2.15(rw,sync,no_subtree_check)
systemctl restart nfs-kernel-server
```

dest-server

```
sudo apt update
sudo apt install nfs-common
mkdir test
sudo mount ip:/home/ubuntu  test
```

allow the nfs port in src and dest secuty group

11. umask (user mask)

1. root user's dufalt umask value 022
2. normal user's dufalt umask value 002

for dirctory	777	for file	666
permission	- 655	pemission	- 655
umask value=122		umask value= 011	

1. umask -p -- to check current umask value of user
2. umask -S --to check permission
3. umask 102 -- to set up umask value for user

11. umask (user mask)

1. vi /etc/sudoers -- change normal user into sudo user
2. user-name ALL=(ALL:ALL) ALL
3. vi /etc/ssh/sshd_config -- set custom port for linux and user login enable
4. Port 21
5. PermitRootLogin yes
6. PubkeyAuthentication yes
7. PasswordAuthentication yes
8. sudo systemctl restart sshd -- restart the sshd
9. hostnamectl -- to check os and version
10. cat /etc/os-release -- to check os and version

12. key and sevice

1. ssh-keygen	-- to genrate public and private key
2. ssh-copy-id ubuntu@172.22.333.4	-- copy the key another server
3. scp -r file.txt ubuntu@172.22.333.4	-- copy the file one sever to another
4. cd /home/ubuntu/.ssh/	-- check public key and private key of ubuntu user
5. cd /root/.ssh	-- check root account public and private key

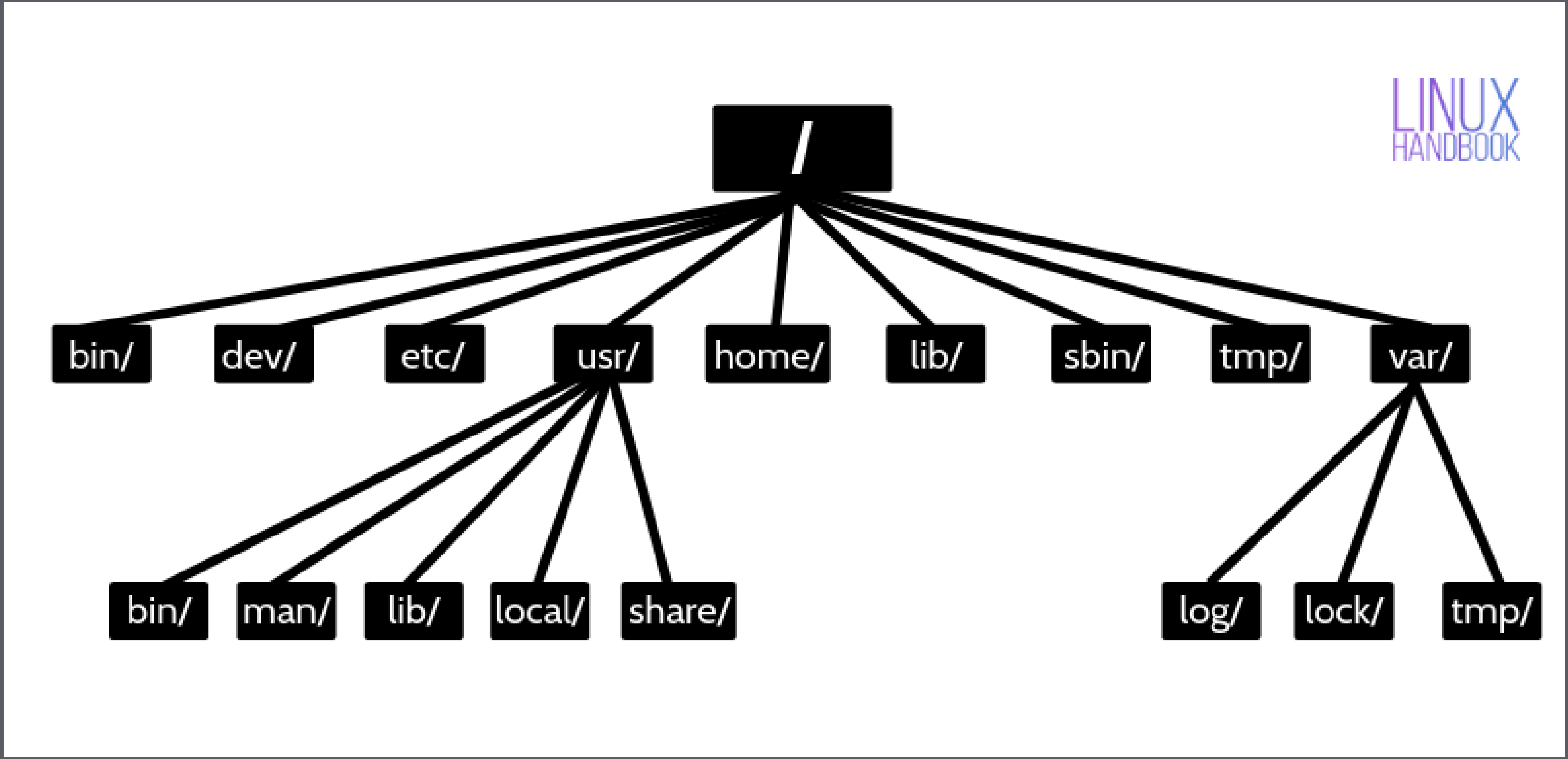
12. key and sevice

1.	sudo apt update	-- to check the version
2.	sudo apt upgrade	-- update the application to new version
3.	sudo apt install apache2	-- install the apache2 application in ubuntu
4.	sudo yum update	-- update the centos os
5.	sudo yun install httpd	-- install httpd in centos
6.	sudo systemctl enable apache2	-- enable the apache2 application
7.	sudo systemctl start apache2	-- start the apalication
8.	sudo systemctl stop apache2	-- stop the application
9.	sudo systemctl status apache2	-- to check the status of application
10.	sudo systemctl restart apache2	-- restart application

12. key and sevice

1. sudo apt-get remove apache2 --
2. apt-get purge apache2 -- remove application

Linux Directory Structure



1. /

This is the root directory which should contain only the directories needed at the top level of the file structure

2 /bin

This is where the executable files are located. These files are available to all users

3 /dev

These are device drivers

4 /etc

Supervisor directory commands, configuration files, disk configuration files, valid user lists, groups, ethernet, hosts, where to send critical messages

5 /lib

Contains shared library files and sometimes other kernel-related files

6 /boot

Contains files for booting the system

7 /home

Contains the home directory for users and other accounts

8 /mnt

Used to mount other temporary file systems, such as cdrom and floppy for the CD-ROM drive and floppy diskette drive, respectively

9 /proc

Contains all processes marked as a file by process number or other information that is dynamic to the system

10 /tmp

Holds temporary files used between system boots

11 /usr

Used for miscellaneous purposes, and can be used by many users. Includes administrative commands, shared files, library files, and others

12 /var

Typically contains variable-length files such as log and print files and any other type of file that may contain a variable amount of data

13 /sbin

Contains binary (executable) files, usually for system administration. For example, fdisk and ifconfig utlities

14 /kernel

Contains kernel files

What is Shell?

- Shell is a UNIX term for an interface between a user and an operating system service.
- Shell provides users with an interface and accepts **human-readable commands** into the system and executes those commands which can run **automatically and give the program's output** in a shell script.

Types of shell

- Bash (**B**ourne **A**gain **SH**ell)
- c shell
- ksh shell (**k**orn **sh**ell)
- zsh shell

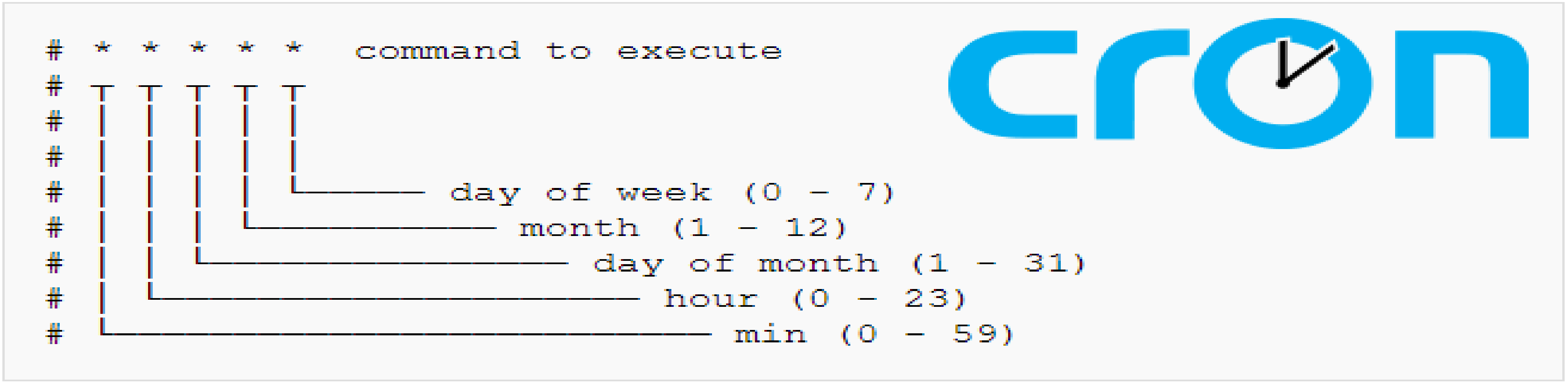
Shell Scripting

- sh is also called Bourne Shell . sh is implemented by programs like **dash, kash**, and original Bourne Shell.
- sh is **not a programming language** itself. It is just a specification.
- Users can provide human-readable commands to a shell and then shell convert them into kernel understandable form.
- For avoiding manual and **repeted work**.
- Shell scripting is used by system admins for many routine backups.

crontab -e -- set up cronjob

crontab -l -- check the cron jobs

cd /root -- cron executed files if root is set-up cronjob



```
# * * * * *  command to execute
# ┬ ┬ ┬ ┬ ┬
# │ │ │ │ │
# │ │ │ │ │
# │ │ │ │ └───── day of week (0 - 7)
# │ │ │ └────────── month (1 - 12)
# │ │ └─────────────── day of month (1 - 31)
# │ └──────────────────── hour (0 - 23)
# └───────────────────────── min (0 - 59)
```

sample script

```
#!/bin/sh
pwd
mkdir demo
cd demo
pwd
touch file.txt
echo “this new dirtory” > file.txt
cp file.txt /home/ubuntu
ls -l
chmod 655 file.txt
```

install apache2

```
#!/bin/sh
apt update -y
apt install apache2 -y
sudo systemctl restart apache2
sudo systemctl enable apache2
sleep 10s
cd  /var/www/html/
rm -rf index.html
echo “this is my script” > index.html
systemctl restart apache2
```

script for send log file to aws s3

```
#!/bin/sh
pwd
cd /var/log/apache2/
#copy the access.log file to s3 bucket
aws s3 cp  -r * s3://test234567875/
sleep 20s
# delete  the file content without deleteing file
>  access.log
```

```
#!/bin/sh

apt update -y

apt install unzip -y

sleep 20s

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

10s

unzip awscliv2.zip

3m

sudo ./aws/install

1m
```

BASH SCRIPT

- Bash Scripting is a powerful part of system administration and development used at an extreme level.
- It is used by the System Administrators, Network Engineers, Developers, Scientists, and everyone who use Linux/Unix operating system.
- They use Bash for system administration, data crunching, web application deployment, automated backups, creating custom scripts for various pages, etc.
- A Bash script is a computer program written in the Bash programming language.
- It is used to automate repetitive tasks on Linux filesystem

print statment

```
#! /bin/bash

# This is the basic bash script

echo " Hello World! "
```

Bash variables

```
#! /bin/bash
# User-Defined Variables
name=Peter
ROLL_NO=5245325
echo "The student name is $name and his Roll
number is $ROLL_NO."
```

Add the command in bash script

```
#!/bin/bash

#This is a single-line comment in Bash Script.
echo "Enter your name:"
read name
echo
#echo output, its also a single line comment
echo "The current user name is $name"
#This is another single line comment
```

Bash Arithmetic Operators

```
Num1=10

Num2=3

A=$((Num1+Num2))

echo "Sum = $A"
```

```
Num1=10

Num2=3

A=$((Num1*Num2))

echo "Sum = $A"
```

```
Num1=10

Num2=3

A=     (Num1-Num2))

echo "Sum = $A"
```

```
Num1=10

Num2=3

A=$((Num1/Num2))

echo "Sum = $A"
```

if statements

```
#!/bin/bash

#if condition (greater than) is true
if [ 10 -gt 3 ];
then
echo "10 is greater than 3."
fi
```

```
#!/bin/bash

#if condition (equal to) is true
if [ 10 -eq 10 ];
then
echo "10 is equal to 10."
fi
```

```
#!/bin/bash

#if condition (lesser than) is true
if [ 3 -lt 10 ];
then
echo "3 is less than 10."
fi
```

```
#!/bin/bash

# TRUE && TRUE
if [ 8 -gt 6 ] && [ 10 -eq 10 ];
then
echo "Conditions are true"
fi
```

Bash If Else

```
#!/bin/bash

#when the condition is true
if [ 10 -gt 3 ];
then
  echo "10 is greater than 3."
else
  echo "10 is not greater than 3."
fi
```

```
#!/bin/bash

#when the condition is false
if [ 3 -gt 10 ];
then
  echo "3 is greater than 10."
else
  echo "3 is not greater than 10."
fi
```

Bash For Loop

```
#!/bin/bash

for num in {1..10}
do
echo $num
done

echo "Series of numbers from 1 to 10."
```

```
#!/bin/bash

#For Loop to Read Three-expression

for ((i=1; i<=10; i++))
do
echo "$i"
done
```

while loop

```
#!/bin/bash
#Table of 2

for table in {2..100..2}
do
echo $table
if [ $table == 20 ]; then
break
fi
done
```

```
#!/bin/bash
#An infinite while loop

while true
do
echo "Welcome to linux class.";
done
```

bash	sh
Bourne Again SHell	SHell
bash is the default SHELL	sh is the not default SHELL
#!/bin/bash	#!/bin/sh
It has more Functionality with up-gradation.	It has less functionality.
Easy to use	not as easy as bash
less portable than sh.	more portable than bash.
Bash scripting is scripting specifically for Bash	Shell scripting is scripting in any shell
supports command history.	does not supports command histoy.
Developed by Brain Fox	Developed by Stephen R. Bourne

See you tomorrow

THANK YOU

Click here to add content of the text.