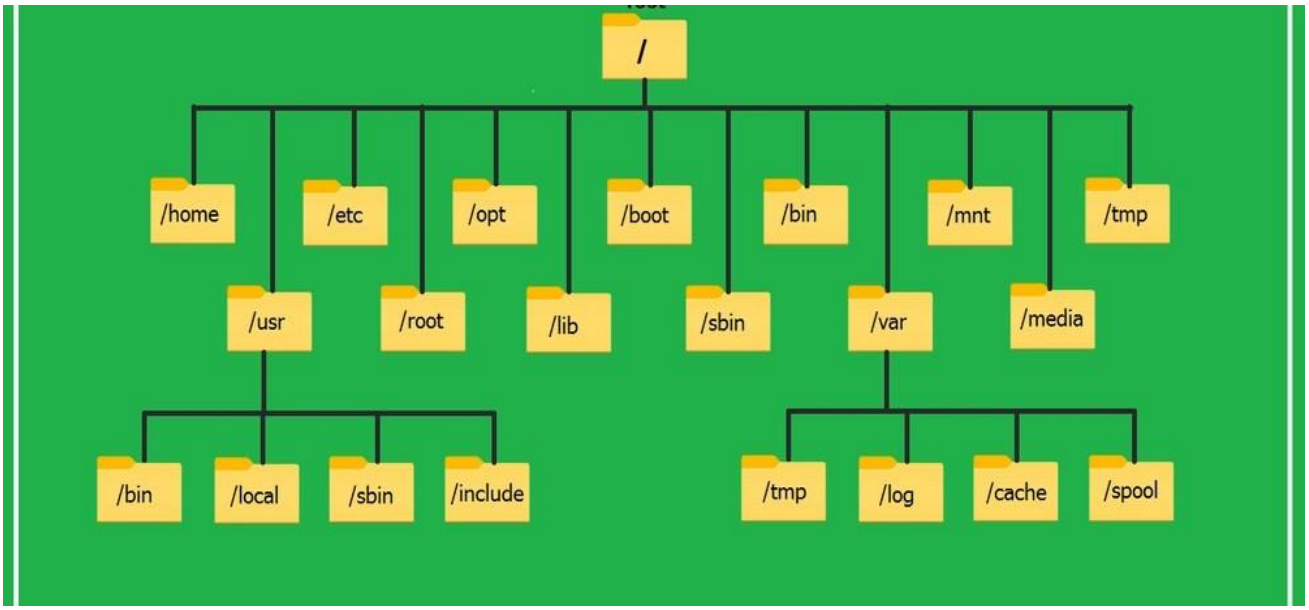
Day 07 – Linux File System Hierarchy & Scenario-Based Practice

➢ File System Hierarchy:

1. In Linux Everything is a file and directory including a hardware program. Files are stored in directory.
2. Directory contains a file with tree structure. That's called File System Hierarchy.



Core Directories:

a. root directory / :
- All Linux systems have a directory structure that starts at the root directory.
- Everything that exists on your Linux system can be found below this root directory. All files and directories come under this root directory.
- It is like the top-level parent folder of the entire system.

```
ubuntu@ip-172-31-33-36:~$ ls /
bin             boot  etc   lib             lib64       media  opt   root  sbin             snap  sys  usr
bin.usr-is-merged  dev   home  lib.usr-is-merged  lost+found  mnt    proc  run   sbin.usr-is-merged  srv   tmp  var
ubuntu@ip-172-31-33-36:~$
```

b. /home :
- This directory contains home folders of normal users.
- Each user gets their own folder here to store personal files, documents, scripts, etc.

```
ubuntu@ip-172-31-33-36:~$ ls /home
Tokyo  berlin  nairobi  professor  ubuntu
ubuntu@ip-172-31-33-36:~$
```

c. /root :
- It is the home directory for the root user (superuser).
- It is different from /home and only accessible by the root user. Root user stores admin-related files here.

d. /etc :

- This directory contains system configuration files.
- Important configurations like network settings, user accounts, services, and software configurations are stored here.

```
bash_completion         gnutls              logrotate.conf      pollinate           sudoers.d
bash_completion.d       groff               logrotate.d         ppp                 supercat
bindresvport.blacklist  group               lsb-release         profile             sysctl.conf
binfmt.d                group-              lvm                 profile.d           sysctl.d
byobu                   grub.d              machine-id          protocols           sysstat
ca-certificates         gshadow             magic               python3             systemd
ca-certificates.conf    gshadow-            magic.mime          python3.12          terminfo
chrony                  gss                 manpath.config      rc0.d               timezone
cloud                   hdparm.conf         mdadm               rc1.d               tmpfiles.d
cni                     hibagent-config.cfg mime.types          rc2.d               ubuntu-advantage
console-setup           hibinit-config.cfg  mke2fs.conf         rc3.d               ucf.conf
credstore               host.conf           modprobe.d          rc4.d               udev
credstore.encrypted     hostname            modules             rc5.d               udisks2
cron.d                  hosts               modules-load.d      rc6.d               ufw
cron.daily              hosts.allow         mtab                rcS.d               update-manager
cron.hourly             hosts.deny          multipath           resolv.conf         update-motd.d
cron.monthly            init                multipath.conf      rmt                 update-notifier
cron.weekly             init.d              nanorc              rpc                 usb_modeswitch.conf
cron.yearly             initramfs-tools     needrestart         rsyslog.conf        usb_modeswitch.d
crontab                 inputrc             netconfig           rsyslog.d           vconsole.conf
cryptsetup-initramfs    iproute2            netplan             screenrc            vim
crypttab                iscsi               network             security            vmware-tools
dbus-1                  issue               networkd-dispatcher selinux             vtrgb
debconf.conf            issue.net           networks            sensors.d           wgetrc
debian_version          kernel              newt                sensors3.conf       xattr.conf
default                 landscape           nftables.conf       services            xdg
deluser.conf            ld.so.cache         nginx               sgml                xml
depmod.d                ld.so.conf          nsswitch.conf       shadow              zsh_command_not_found
dhcp                    ld.so.conf.d        opt                 shadow-
ubuntu@ip-172-31-33-36:~$
```

e. /var/log :

- This directory stores system and service log files.
- Logs help in troubleshooting errors and monitoring services like nginx, ssh, etc.

```
ubuntu@ip-172-31-33-36:~$ ls /var/log
README            auth.log              cloud-init.log    dmesg.2.gz  kern.log.1     syslog
alternatives.log  auth.log.1            cloud-init.log.1  dmesg.3.gz  kern.log.2.gz  syslog.1
amazon            auth.log.2.gz         dist-upgrade      dmesg.4.gz  landscape      syslog.2.gz
apport.log        btmp                  dmesg             dpkg.log    lastlog        sysstat
apt               chrony                dmesg.0           journal     nginx          unattended-upgrades
atop              cloud-init-output.log dmesg.1.gz        kern.log    private        wtmp
ubuntu@ip-172-31-33-36:~$
```

f. /tmp :

- This directory stores temporary files created by the system and applications. It is used for temporary data storage.
- Files here are usually **deleted automatically after reboot**.

```
ubuntu@ip-172-31-33-36:~$ ls /tmp
runbook-demo
snap-private-tmp
systemd-private-af03dc30f5a94fa19d7cb9a5f65388ef-ModemManager.service-xA8Yck
systemd-private-af03dc30f5a94fa19d7cb9a5f65388ef-chrony.service-Vxq3J4
systemd-private-af03dc30f5a94fa19d7cb9a5f65388ef-polkit.service-zYkC9Q
systemd-private-af03dc30f5a94fa19d7cb9a5f65388ef-systemd-logind.service-rKuEDc
systemd-private-af03dc30f5a94fa19d7cb9a5f65388ef-systemd-resolved.service-VNCNGS
ubuntu@ip-172-31-33-36:~$
```

Additional Directories:

a. /bin :

- This directory contains essential system commands required for basic operation. Commands like ls, cp, mv, cat, and rm are stored here.
- These commands are needed even when no other filesystem is mounted.

```
gtbl                    sadf                    xzcat
gunzip                  sar                     xzcmp
gzexe                   sar.sysstat             xzdiff
gzip                    savelog                 xzegrep
h2ph                    sbattach                xzfgrep
h2xs                    sbkeysync               xzgrep
hardlink                sbsiglist               xzless
hd                      sbsign                  xzmore
head                    sbvarsign               yes
helpztags               sbverify                ypdomainname
hexdump                 scalar                  zcat
hibagent                scandeps                zcmp
hibinit-agent           scp                     zdiff
host                    screen                  zdump
hostid                  screendump              zegrep
hostname                script                  zfgrep
hostnamectl             scriptlive              zforce
htop                    scriptreplay            zgrep
hwe-support-status      scsi_logging_level      zipdetails
i386                    scsi_mandat             zless
iconv                   scsi_readcap            zmore
id                      scsi_ready              znew
inetutils-telnet        scsi_satl               zstd
info                    scsi_start              zstdcat
infobrowser             scsi_stop               zstdgrep
infocmp                 scsi_temperature        zstdless
infotocap               sdiff                   zstdmt
install                 sed
install-info            select-editor
ubuntu@ip-172-31-33-36:~$
```

b. /usr/bin :

- This directory contains most of the user-level commands and software binaries. It includes programs like nano, vim, git, python, etc. These are used by normal users for daily tasks.

```
efibootmgr                  mtr-packet              sg_read_buffer          write
egrep                       mv                      sg_read_long            x86_64
eject                       namei                   sg_readcap              x86_energy_perf_policy
enable-ec2-spot-hibernation nano                    sg_reassign             xargs
enc2xs                      nawk                    sg_referrals            xauth
encguess                    nc                      sg_rep_pip              xdg-user-dir
env                         nc.openbsd              sg_rep_zones            xdg-user-dirs-update
envsubst                    neqn                    sg_requests             xsubpp
eqn                         netaddr                 sg_reset                xxd
ex                          netcat                  sg_reset_wp             xz
expand                      networkctl              sg_rmsn                 xzcat
expiry                      networkd-dispatcher     sg_rtpg                 xzcmp
expr                        newgrp                  sg_safte                xzdiff
factor                      ngettext                sg_sanitize             xzegrep
faillog                     nice                    sg_sat_identify         xzfgrep
fallocate                   nisdomainname           sg_sat_phy_event        xzgrep
false                       nl                      sg_sat_read_gplog       xzless
fgconsole                   nohup                   sg_sat_set_features     xzmore
fgrep                       nproc                   sg_scan                 yes
file                        nroff                   sg_seek                 ypdomainname
finalrd                     nsenter                 sg_senddiag             zcat
find                        nslookup                sg_ses                  zcmp
findmnt                     nstat                   sg_ses_microcode        zdiff
flock                       nsupdate                sg_start                zdump
fmt                         ntfs-3g                 sg_stpg                 zegrep
fold                        ntfs-3g.probe           sg_stream_ctl           zfgrep
free                        ntfscat                 sg_sync                 zforce
fs-idmap                    ntfscluster             sg_test_rwbuf           zgrep
ftp                         ntfscmp                 sg_timestamp            zipdetails
fuser                       ntfsdecrypt             sg_turs                 zless
fusermount                  ntfsfallocate           sg_unmap                zmore
fusermount3                 ntfsfix                 sg_verify               znew
fwupdmgr                    ntfsinfo                sg_vpd                  zstd
fwupdtool                   ntfsls                  sg_wr_mode              zstdcat
gapplication                ntfsmove                sg_write_buffer         zstdgrep
gawk                        ntfsrecover             sg_write_long           zstdless
gawkbug                     ntfssecaudit            sg_write_same           zstdmt
gdbus                       ntfstruncate            sg_write_verify
geqn                        ntfsusermap             sg_write_x
getconf                     ntfswipe                sg_xcopy
```

c. /opt :

- This directory is used to install optional and third-party applications. Software like Docker, custom applications, or vendor tools are often installed here.

- It helps keep third-party software separate from system files.

```
ubuntu@ip-172-31-33-36:~$ ls /opt
containerd  dev-project  team-workspace
```

Hands on Task :

1. du -sh /var/log/* 2>/dev/null | sort -h | tail -5 :
   - This command checks the size of all log files in /var/log directory.
   - It then sorts them and shows the 5 largest log files. This helps to find which logs are using more disk space for troubleshooting and cleanup.
   - 2>/dev/null: its throw error in null space this space doesn't exist in system. (Throw in trash or black hole)

```
ubuntu@ip-172-31-33-36:~$ du -sh /var/log/* 2>/dev/null | sort -h | tail -5
664K    /var/log/cloud-init.log
720K    /var/log/syslog.1
1.2M    /var/log/cloud-init.log.1
1.3M    /var/log/syslog
141M    /var/log/journal
ubuntu@ip-172-31-33-36:~$
```

2. cat /etc/hostname :
   - This command is used to display hostname of the system.
   - The hostname is the name given to the server or computer on a network.

```
ubuntu@ip-172-31-33-36:~$ cat /etc/hostname
ip-172-31-33-36
ubuntu@ip-172-31-33-36:~$
```

3. Ls -la ~ :
   - list all files and directories in the current user's home directory with detailed information.
   - It shows hidden files, permissions, owner, size, and modification date.

```
-rw-r--r--  1 ubuntu     ubuntu       3771 Mar 31  2024 .bashrc
drwx------  2 ubuntu     ubuntu       4096 Feb  3 17:07 .cache
drwx------  4 ubuntu     ubuntu       4096 Feb 19 16:53 .config
drwxrwxr-x  8 ubuntu     ubuntu       4096 Feb 14 06:06 .git
-rw-rw-r--  1 ubuntu     ubuntu         65 Feb 14 05:24 .gitconfig
-rw-------  1 ubuntu     ubuntu         20 Feb 20 12:08 .lesshsQ
-rw-------  1 ubuntu     ubuntu         20 Feb 20 08:41 .lesshst
drwxrwxr-x  3 ubuntu     ubuntu       4096 Feb 14 05:18 .local
-rw-r--r--  1 ubuntu     ubuntu        807 Mar 31  2024 .profile
drwx------  2 ubuntu     ubuntu       4096 Feb  3 17:05 .ssh
-rw-r--r--  1 ubuntu     ubuntu          0 Feb  3 17:08 .sudo_as_admin_successful
-rw-------  1 ubuntu     ubuntu       7543 Feb 14 05:30 .viminfo
drwxrwxr-x  2 ubuntu     ubuntu       4096 Feb 14 05:30 Git
drwxrwxr-x  3 ubuntu     ubuntu       4096 Feb 14 05:17 Git_Command
drwxrwxr-x  2 berlin     heist-team   4096 Feb 15 12:02 app-logs
drwxrwxr-x  2 ubuntu     ubuntu       4096 Feb 15 12:27 bank-heist
-rw-rw-r--  1 berlin     ubuntu          0 Feb 15 11:30 devops-file.txt
-r--r--r--  1 ubuntu     ubuntu         19 Feb 12 11:56 devops.txt
-rw-rw-r--  1 ubuntu     ubuntu          0 Feb 14 05:25 gitcommand.txt
drwxrwxr-x  4 professor  planners     4096 Feb 15 12:13 heist-project
-rw-rw-r--  1 ubuntu     ubuntu         48 Feb  6 16:09 myfile.txt
-rw-r-----  1 ubuntu     ubuntu         51 Feb  6 16:02 newfile.txt
-rwxrwxr-x  1 ubuntu     ubuntu         77 Feb 12 11:57 newscript.sh
drwxrwxr-x  2 ubuntu     ubuntu       4096 Feb  3 19:04 nginx_logs
-rw-rw-r--  1 ubuntu     ubuntu         51 Feb 20 11:30 notes.txt
drwxr-xr-x  2 ubuntu     ubuntu       4096 Feb 12 11:48 project
-rw-rw-r--  1 professor  heist-team      0 Feb 15 11:58 project-config.yaml
-rwxrwxrwx  1 ubuntu     ubuntu         21 Feb  6 16:18 script.sh
-rw-rw-r--  1 ubuntu     heist-team      0 Feb 15 11:37 teams-notes.txt
```

Part 2: Scenario-Based Practice

Scenario 1: Service Not Starting

Step 1 : Check status of “myapp” service using command “sudo systemctl status myapp”.

Reason: This command checks whether the myapp service is running, stopped or failed, and shows basic error information.

Step 2: Check log of myapp service using command “sudo journalctl -u myapp -n 50”.

Reason: This command shows the last 50 log entries of my app service that helps to find exact error causing the failure.

Step 3: Check service is enabled or not using command “systemctl is-enabled myapp”

Reason: This command checks whether the service is enabled to start automatically after system reboot.

Step 4: check myapp service exists in the system using command

“systemctl list-units –type=service | grep myapp”.

Reason: This command verifies whether the myapp service exists in the system.

Outcome: First check the service status, then check the logs for errors, and service is enabled and properly configured.

Scenario 2: High CPU Usage

Step1: Check CPU usage and running process using “top” command.

Reason: shows CPU usage.

Step 2: sort processes by highest CPU usage. Ps aux –sort=-%cpu | head -19

Reason: list the top 10 processes that use highest CPU along with their PID.

Step 3: use “htop” command it makes easier to identify high CPU consuming processes.

Reason 4: user-friendly and colourful view of CPU usage, making it easier to identify high CPU consuming processes.

Outcome:
First monitor live CPU usage, then identify the top CPU consuming process and its ID.

Scenario 3: Finding Service Logs (docker)

Step 1: Check Docker service status. “sudo systemctl status docker”

Reason: Checks docker service is running and shows basic log and errors.

Step 2: check last 50 lines of docker service logs. “journalctl -u docker -n 50”

Reason: Check recent activity and error.

Step 3: check real time docker log. “journalctl -u docker -f”

Reason: helping to monitor live service activity.

Scenario 4: File Permissions Issue

Step 1: Check current permission of backup.sh “ls -l /backup.sh”

```
ubuntu@ip-172-31-33-36:~$ ls -l backup.sh
-rw-rw-r-- 1 ubuntu ubuntu 0 Feb 20 14:56 backup.sh
ubuntu@ip-172-31-33-36:~$
```

Reason: This command shows the current file permissions and confirms that the file does not have execute (x) permission.

Step 2: Add execute permission. “chmod +x backup.sh”

```
ubuntu@ip-172-31-33-36:~$ chmod +x backup.sh
ubuntu@ip-172-31-33-36:~$ ls -l backup.sh
-rwxrwxr-x 1 ubuntu ubuntu 0 Feb 20 14:56 backup.sh
ubuntu@ip-172-31-33-36:~$
```

Reason: This command adds execute permission so the script can be run.

Step 3: Verify execute permission added

```
ubuntu@ip-172-31-33-36:~$ ls -l backup.sh
-rwxrwxr-x 1 ubuntu ubuntu 0 Feb 20 14:56 backup.sh
ubuntu@ip-172-31-33-36:~$
```

Reason: confirms that execute (x) permission is successfully added.

Step 4: Run the script

```
ubuntu@ip-172-31-33-36:~$ ./backup.sh
WE Made Script Executable !!!!
ubuntu@ip-172-31-33-36:~$
```