(7) /home

- /home is the directory that stores personal data of normal user.
- It contains user files, setting, downloads, document picture, videos, config

Example:
/home
/home/username

Run:
ls /home
To see your usernome and all the files of your computer.

- Purpose of /home
- keeps your data seperate from system file
- Makes linux multi-user friendly
- Allows system upgr udgrade without deleating file

System file → /bin, /etc, /usr

User file → /home

So, what's inside a /home/username directory?

⇒ Typical contents.

Desktop/
Document/
Downloads/
Music/
Picture/
Videos/

Also hidden files (start with .):

To see hidden files
Run
ls -a

- -a → shows all files including hidden ones
- Hidden files start with . (dot)

Example:
- bashc → shell setting
- profile
- cache
- config → application config

They are hidden cause:
- They store user configuration
- Prevent accidental deletion
- Mostly not meant to be deleted-edited
- Shell setting

That's why setting remain unchanged after reboot.

- Permission and security
  - Each user can access their own directory
  - Other user cannot access your file
  - Root (/root) is seperate from /home

/root → root user
/home → normal user

Important Note

- /home is not required for system boot
- If /home is deleted → system will run but user might loose data
- Often placed on a seperate ~~dis~~ partition for saftey.

/aiem080/

(8) /var

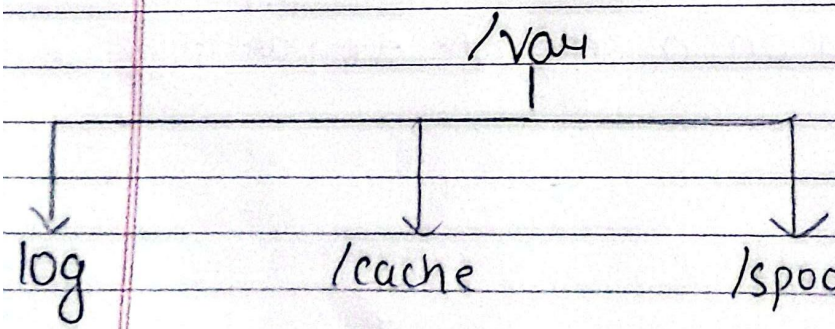
/var ~~state~~ stands for variable data

It stores files that changes frequently while the system is running

→ Unlike /bin, /sbin or /usr ; /var is not static

Why VAR exits?

- To keep growing and changing file seperates
- Prevent system directories from filling up
- Helps in logging, caching and spooling



Important Directories inside /var

/var/log

→ Stores system log

Example:

- syslog
- auth.log
- kern.log

⇒ /var/log/auth.log

- It tracks login attempt (SSH, sudo, GUI) password changes.
- Hackers look for "failed password" to see if someone is brute-forcing the machine.

⇒ /var/log/syslog

- It tracks ~~system~~ General system message; the catch all log
- If system crashes a wierd ~~messo~~ script runs, the error message ends up here.

⇒ /var/log/kern.log

- It tracks Kernel message (hardware, driver, firewall

⇒ Hackers checks if a buffer overflow exploit crashed the system memory.

⇒ /var/log/apache2
- Web server access and errors
- Find out hidden URLs an attacker tried to visit

/var/cache

- Stores cached data from application
Example:
- Package manager cache (apt)
- apt/archieves/ ; when you install software, the .deb file are kept here so you don't have to download them again if you reinstall
- man/ : Pre-formated manual page for quick reading
- fontconfig/ - Pre calculated lists of available fonts.

Note: Cache is temporary. It is safe to delete it as system can always recreate or redownload it (though the system might run bit slower for a moment while it re-learn that data

/var/spool

- Spool stands for Simultaneous Peripheral Operation On-line. It is a holding area of task that are waiting to be processed by a program or piece of hardware

How spool works?

- Think of a printer. if you send a 50 page document

to print, your computer doesn't sit there frozen until the printer finishes. Instead it spools data into "/var/spool". The data sits there comfortably while the printer slowly works through pages. Once the job is finished, the data/file is deleted.

Run `ls /var/spool`

- /var/spool/mail : This is where incoming mail sit for user until they open their mail to read them

- /var/spool/cron : When you are editing by running crontab -e the file you are editing is actually stored here.

- /var/spool/rsyslog :

rsyslog (reliable System log) is the service responsible for collecting all the message from Kernel and applications and writing them to /var/log.

rsylog writes message directly to disk. However, rsylog is often used to send log to another computer over the network (a central log server)

If your Internet goes down or central log server crashes, rsylog doesn't want to loose those important log message. So it spools them to /var/spool/rsylog/.

- The log sit in the spool as temporary file
- When rsylog keeps checking the connection
- As soon as connection is back, it "flushes" the spool, sending all backed-log to their destination

Note: Your rsylog may be empty as it is normal for home user or single-server setup.
This is because

- logs are being written locally at /var/log
- There is no network congestion causing a backup

Run: sudo ls /var/spool/rsyslog
If no file that means you system is healthy
Your log is being written to /var/log without any "traffic jams".

/var/crash

→ When an program doesn't just quit, but crashes (segmentation fault, etc), the system tries to capture a snapshot of exactly what was happening in the computer's memory at the exact microsecond.

- What's inside: Core dumps or crash report:
On system like Ubuntu, you you'll see files ending in .crash.

/ alimo80 /

- Why it's there: Developer or you can figure out why the program ~~Failed~~. failed

Run.
ls -lh /var/crash

- If you see empty, then app has been behaving perfectly.

/var/metrics

This is relatively newer folder in the linux world
It is where the system stores data about how it's performing

- What's inside: Small data files tracking things like how long it took to boot, how much data was sent over the network, or how much battery was used.

Run:
ls -l /var/metrics

If you see 0 then it's fine

var/tmp

⇒ Here files are saved even after reboot
⇒ large files that need to survive reboot or ~~crash~~ restart
⇒ ~~Hives~~ lives in a hard disk.

(For example)

If you are downloading 50 GB files and the installer need to put the pieces somewhere while it works, it uses /var/tmp. If your power goes out, the data is still there when you turn the computer back on.

Difference between Cache and Spool

- Cache (var/cache): The data is kept so it can be reused later to save time

- Spool (var/spool): The data is kept because it is waiting to be finished. Once processed data is gone (for e.g printing).

(9) /tmp

/tmp stands for temporary file

It stores the file which are ~~running~~ needed only for a short time while the system or application are running.

- /tmp is usually cleared automatically on reboot
That's the key difference from /var/tmp.

	cleared on reboot
/tmp	Yes
/var/tmp	NO

What live inside /tmp.

If you Run: ls -la /tmp
You will see bunch of stronge looking file.
These are usually "temporary sockets" or "lock files".

Permission

If you look at the permission of /tmp (ls -ld /tmp) you will see

drwxrwxrt

rwx → anyone can read write
t → sticky bit

alisn080

The t at the end is virtual for security.
/tmp is a "world-writable" folder. Any user or program can put a file there.
Without sticky bit, a mean user can accidentally delete another user's temporary file.
The t ensures that only the person who created a file can delete it.

NOTE: Never save your file at /tmp.

Fun command

Run:

```
df -h /tmp
```

- If you see tmpfs under the filesystem it's in you RAM (super fast)
- If you see /dev/sda1 or /dev/nvme it's on your disk.

10) /boot

⇒ /boot contains file required to start (boot) the Linux System.

Without /boot, Linux cannot load or boot (start)

What happen during Boot?

1) System pre powers on
2) Firmware (BIOS/UEFI) runs
3) Bootloader starts (loads) file from /boot
4) Linux Kernel ~~sys~~ starts
5) System loads and login appear

/boot involves in step 3 and 4

Files Inside /boot

Run As /boot

- vmlinux-6.x.x

→ Compress linux kernel
→ Main core of the operating system.

- Initramfs / Initrd

→ Temporary rootfilesystem
⇒ Helps kernel load drivers and mount real root /

- Bootloader files

GRUB:

→ Configuration for boot Menu

→ Kernel selection

→ Boot parameters

Most Common bootloader

→ GRUB (modern linux)
→ LILO (older, rarely used)

Run:

~~ls~~ uname -r → shows running Kernel version

Matches with.

ls /boot/ vmlinuz-*