Day 2 -> Zero to Hero Batch 10

Owner of This Document-> Nikhil Trivedi

Task1-> The core components of Linux (kernel, user space, init/systemd).

Kernel : The **kernel** is the heart of Linux. It runs in **kernel space** and has full access to hardware.

- **Manages:**
    1. CPU scheduling
    2. Memory
    3. Processes
    4. Devices & drivers
- Exposes **system calls** (fork, exec, read, write)
- Direct hardware access (user programs cannot do this)

User Space : Where applications and services run

- **Example:**

1. Shells (bash)
2. Utilities (ls, ps)
3. Services (nginx, docker)

- Communicates with kernel via system calls
- Isolated for **stability and security**

Init/system:

- First user-space process (PID 1)
- Starts and manages all other services
- **Handles:**
    1. Service startup/shutdown
    2. Restarts on failure
    3. Logging (journald)
    4. Boot targets

Task2-> How processes are created and managed

Process Creation

- **fork()** → creates a new process

- **exec()** → loads a new program into that process
- **Every process has:**
    1. PID (process ID)
    2. PPID (parent process ID)

Process States (Very Important)

- R (Running)
    - Actively using CPU or ready to run
- S (Sleeping)
    - Waiting for an event (normal state for most processes)
- D (Uninterruptible Sleep)
    - Waiting for I/O (disk, network)
    - If many here → storage issue
- Z (Zombie)
    - Process finished but parent didn't clean it up
    - Indicates bad process management
- T (Stopped)
    - Paused manually (signal, debugger)

Task3-> What systemd does and why it matters

- Starts services in parallel
- Manages dependencies (DB before app)
- Auto-restarts crashed services
- Central logging via **journalctl**
- Replaces old **init** and **cron** (via timers)
- **Everything is a unit: service, timer, socket, mount**

Task4-> List 5 commands you would use daily

- **ps aux**
  → Check running processes & states

- **top / htop**
  → Live CPU, memory, load analysis
- **systemctl status <service>**
  → Service health & logs
- **journalctl -u <service>**
  → Debug service failures
- **df -h**
  → Disk usage (common outage cause)