Chapter 8: Linux Boot Process & systemd

1. Why the Linux Boot Process Matters (Interview Context)

When a Linux system:

- Starts
- Reboots
- Fails to boot

You are expected to understand:

- What happens step by step
- Where failures occur
- How to recover the system

Interview insight:

Boot process questions test real system-level understanding, not memorization.

---

2. High-Level Linux Boot Flow (Must Remember)

```
Power On
 → BIOS / UEFI
 → Bootloader (GRUB)
 → Kernel
 → init / systemd
 → Services
 → Login Prompt
```

This sequence is commonly asked directly in interviews.

---

3. BIOS vs UEFI

BIOS (Basic Input Output System)

- Legacy firmware
- Uses MBR partition table
- Slower boot
- Limited disk size support

UEFI (Unified Extensible Firmware Interface)

- Modern firmware
- Uses GPT partition table
- Faster boot
- Supports large disks
- Secure Boot support

Interview Comparison

Feature	BIOS	UEFI
Boot type	Legacy	Modern
Disk support	Limited	Very large
Speed	Slower	Faster
Security	Basic	Secure Boot

Interview line:

Modern Linux systems mostly use UEFI.

---

4. Bootloader (GRUB)

What Is GRUB?

GRUB (GRand Unified Bootloader) is responsible for:

- Loading the Linux kernel
- Passing kernel parameters
- Allowing OS selection

Why GRUB Is Important

- Without GRUB, kernel cannot load
- Used for recovery and troubleshooting
- Allows booting into rescue or single-user mode

Interview-ready line:

GRUB loads the kernel and hands control to it.

---

5. Linux Kernel During Boot

What Kernel Does at Boot Time

- Initializes hardware
- Loads device drivers
- Mounts root filesystem (read-only)
- Starts init/systemd process

Important fact:

Kernel runs in kernel space with full privileges.

---

6. init vs systemd

init (Legacy)

- Older initialization system
- Sequential startup
- Slower boot

---

systemd (Modern)

Definition

systemd is the **first userspace process (PID 1)** responsible for initializing the system and managing services.

Why systemd Replaced init

- Parallel service startup
- Faster boot times
- Better dependency management
- Centralized logging

Interview line:

systemd is PID 1 on modern Linux systems.

---

7. systemctl vs service

systemctl

- Used with systemd
- Modern and preferred

Examples:

```
systemctl start nginx
systemctl stop nginx
systemctl status nginx
systemctl enable nginx
```

---

service

- **Legacy wrapper**
- **Works on older systems**

Interview insight:

systemctl is preferred on modern distributions.

---

8. Runlevels vs Targets

Runlevels (Legacy init)

- Numeric levels (0–6)

Runlevel	Meaning
0	Shutdown
1	Single-user
3	Multi-user (CLI)
5	Multi-user (GUI)
6	Reboot

---

systemd Targets (Modern)

Target	Equivalent
poweroff.target	Runlevel 0
rescue.target	Runlevel 1
multi-user.target	Runlevel 3
graphical.target	Runlevel 5

Interview line:

systemd uses targets instead of runlevels.

---

9. What Happens During Linux Boot (Step-by-Step)

1. **System power on**
2. **BIOS/UEFI performs hardware checks**
3. **Bootloader loads kernel**
4. **Kernel initializes hardware**
5. **Root filesystem mounted**
6. **systemd starts as PID 1**
7. **systemd starts services**
8. **Login prompt appears**

---

10. How to Check Boot and Service Status

Check System Boot Logs

```
journalctl -b
```

Check Failed Services

```
systemctl --failed
```

Check Default Target

```
systemctl get-default
```

---

11. Recovering a Failed Boot (VERY IMPORTANT)

Common Boot Issues

- Corrupt filesystem
- Wrong fstab entry
- Failed service blocking boot
- Full disk

Recovery Methods

- Boot into rescue mode
- Edit GRUB parameters
- Fix filesystem using fsck
- Disable failing service

Interview insight:

Boot failures are usually configuration or disk related.

---

12. Real-Life Production Scenarios

Scenario 1: System Stuck at Boot

- Check GRUB logs
- Boot into rescue target
- Inspect failing services

Scenario 2: Service Preventing Boot

- Use emergency mode
- Disable service
- Fix configuration

Scenario 3: Server Rebooted Unexpectedly

- Check kernel logs
- Look for OOM or kernel panic

Chapter 8: Interview Takeaways

After this chapter, you should confidently explain:

- Linux boot sequence
- BIOS vs UEFI
- Role of GRUB
- Kernel responsibilities at boot
- systemd and PID 1
- Targets vs runlevels
- Boot failure recovery steps
