Booting process in linux

1. Linux Booting Process

The Linux booting process is the series of steps the system follows from the moment it is powered on until the user is presented with a login prompt. Each stage is responsible for initializing hardware, loading the operating system, and starting system services.

1.1 BIOS / UEFI – Firmware Initialization

When the system is powered on, the BIOS (Basic Input/Output System) or UEFI (Unified Extensible Firmware Interface) firmware starts first.

Its main functions are:

* Perform POST (Power-On Self Test) to verify hardware such as CPU, memory, keyboard, and storage devices.

* Initialize basic hardware components.

* Identify the configured boot device (hard disk, SSD, USB, DVD, or network).

* Load the bootloader into system memory.

Purpose

To verify system hardware and locate the bootable device.

1.2 Bootloader (GRUB)

After BIOS/UEFI finishes, control is passed to the bootloader. The most common Linux bootloader is GRUB2.

The bootloader:

* Displays the boot menu.

* Allows selection of operating systems or kernel versions.

* Loads the Linux kernel and the initramfs image into memory.

Important boot files are located in:

```
/boot/
```

Common files include:

* vmlinuz (kernel)
* initramfs (initial RAM filesystem)
* grub.cfg (boot configuration)

Purpose

To load the kernel and initial RAM filesystem into memory.

1.3 Kernel Initialization

Once loaded, the kernel is decompressed and starts running.

The kernel performs the following tasks:

* Initializes CPU, memory, and system scheduling.
* Detects and configures connected hardware devices.
* Loads required drivers.
* Mounts the root filesystem in read-only mode.
* Starts the first user-space process (PID 1).

Kernel activity can be viewed using:

```
dmesg
```

Purpose

To control hardware, manage system resources, and prepare the operating system environment.

1.4 initramfs (Initial RAM Filesystem)

initramfs is a temporary filesystem stored in memory.

Its responsibilities include:

* Loading storage and filesystem drivers.

* Supporting RAID, LVM, and encrypted volumes.

* Locating and mounting the real root filesystem.

* Handing over control to the main operating system.

Once the real root filesystem is mounted, initramfs is removed from memory.

Purpose

To prepare the system to access the actual root filesystem.

1.5 systemd – System Initialization (PID 1)

After the root filesystem is mounted, the kernel launches systemd, which becomes the first process (PID 1).

systemd is responsible for:

* Mounting additional filesystems.

* Starting and managing system services.

* Handling device management.

* Managing system states called “targets”.

To verify:

```

ps -p 1

```

Purpose

To manage system services and system startup operations.

1.6 Targets and Service Startup

systemd loads a target that defines the system mode.

Common targets include:

* multi-user.target (server / text mode)

* graphical.target (desktop environment)

* rescue.target (maintenance mode)

During this phase, essential services such as networking, logging, cron jobs, and SSH are started.

Active services can be viewed with:

```
systemctl list-units --type=service
```

Purpose

To start all necessary services required for system operation.

1.7 Login Prompt

After all required services are started, the system displays a login interface. Users can now access the system locally or remotely.

Purpose

To allow user interaction with the system.

Boot Sequence Summary

BIOS/UEFI → Bootloader (GRUB) → Kernel → initramfs → systemd → Services → Login

2. Important Port Numbers

Port numbers are used to identify specific services running on a system and allow network communication.

Service	Port Number	Description
FTP	21	File transfer service
SSH	22	Secure remote login
Telnet	23	Remote login (unencrypted)
SMTP	25	Sending email
DNS	53	Domain name resolution
HTTP	80	Web service
POP3	110	Receiving email
IMAP	143	Email access
HTTPS	443	Secure web service
MySQL	3306	Database service

3. Advanced Server Administration Commands

These commands are used for system monitoring, troubleshooting, and performance analysis.

3.1 System and Process Monitoring

Command	Description
top	Real-time CPU and process monitoring
htop	Interactive process viewer
ps aux	Displays all running processes

uptime	Shows system running time and load
free -h	Displays memory usage
vmstat	CPU and memory statistics

3.2 Disk and Storage Monitoring

Command	Description
df -h	Displays disk usage
du -sh	Displays directory sizes
iostat	Disk I/O performance statistics

3.3 Logs and Service Management

Command	Description
systemctl status	Checks service status
journalctl -xe	Displays system logs
dmesg	Displays kernel messages

3.4 Network Monitoring and Troubleshooting

Command	Description
ss -tulnp	Lists listening ports
ping	Tests network connectivity
traceroute	Displays network path

| tcpdump | Captures network packets |

| lsof -i | Displays open network files |

3.5 Advanced Troubleshooting Tools

Command	Description
sar	Displays historical performance data
watch	Repeats a command at intervals
strace	Traces system calls of a process

4. Key Administrative Notes

* systemd is always the first process (PID 1).
* journalctl is used for centralized logging.
* ss is used to check open ports.
* df and free are essential for resource monitoring.
* strace is used for deep-level troubleshooting.