

Computer Boot Process: Essential Guide

Understanding How Your Computer Starts Up

HRM

August 21, 2025

Table of contents

| | |
|--|----------|
| Computer Boot Process: Essential Guide | 1 |
| Table of Contents | 1 |
| Overview | 2 |
| Boot Process Steps | 2 |
| Step 1: Firmware Initialization | 2 |
| Step 2: POST and Boot Device Detection | 2 |
| Step 3: Bootloader Execution | 2 |
| Step 4: Operating System Loading | 2 |
| Partition Styles vs File Systems | 2 |
| Partition Styles | 2 |
| File Systems | 4 |
| Boot Process Summary | 5 |
| Frequently Asked Questions | 5 |
| Q1: What's the difference between BIOS and UEFI? | 5 |
| Q2: What's the difference between partition style and file system? | 5 |
| Q3: Can I dual boot multiple operating systems? | 5 |
| Resources | 5 |
| Video Tutorials | 5 |
| Remove Previous Installed Distributions | 6 |
| Update WSL -> Ubuntu Installation | 6 |
| Install JavaScript Tools | 6 |
| Install Python Tools | 6 |
| Install C/C++ Tools | 7 |
| Install Java Tools | 7 |
| Use NPX | 7 |

Boot Process Detailed Guide

Computer Boot Process: Essential Guide

Table of Contents

- [1. Overview](#)
- [2. Boot Process Steps](#)
- [3. Partition Styles vs File Systems](#)
- [4. Boot Process Summary](#)
- [5. Frequently Asked Questions](#)
- [6. Resources](#)

Overview

The computer boot process transforms your computer from powered-off state to a fully operational system. This guide covers the essential steps and concepts needed to understand how computers start.

Boot Process Steps

Step 1: Firmware Initialization

When you press the power button:

- **CPU** executes the first program BIOS/UEFI
 - **BIOS** (Basic Input/Output System) - Legacy firmware (used in old computers)
 - **UEFI** (Unified Extensible Firmware Interface) - Modern firmware

Step 2: POST and Boot Device Detection

POST (Power-On Self Test):

- Tests CPU, RAM, and storage devices
- Validates hardware components

Boot Device Selection:

- Reads **Boot Order** from firmware settings
- **GPT drives:** Looks for EFI System Partition
- **MBR drives:** Checks Master Boot Record in first sector

Step 3: Bootloader Execution

Common Bootloaders:

Linux or Window bootloader, both can scan and start any OS windows or linux.

- **Linux:** GRUB2, LILO, systemd-boot - **Windows:** Windows Boot Manager

Bootloader Tasks:

- Scans partitions for installed operating systems
- Presents boot menu (if multiple OS found)
- Loads selected OS kernel into memory

Step 4: Operating System Loading

Linux OS Boot:

1. Kernel loads and initializes hardware
2. **systemd** starts (modern init system)
3. System services launch
4. User login interface appears

Windows OS Boot:

1. **NT Kernel** (`ntoskrnl.exe`) loads
2. **Hardware Abstraction Layer** initializes
3. **Registry** and system drivers load
4. **Session Manager** starts Windows subsystems
5. **Windows Logon** presents login interface

Partition Styles vs File Systems

Partition Styles

Partition styles define how a drive is divided into sections:

| Feature | MBR | GPT |
|-----------------------|--|--------------|
| Max Partitions | 4 primary OR 3 primary + 1 extended | 128 primary |
| Max Storage | 2 TB | 18+ exabytes |
| Boot Support | BIOS only | BIOS + UEFI |

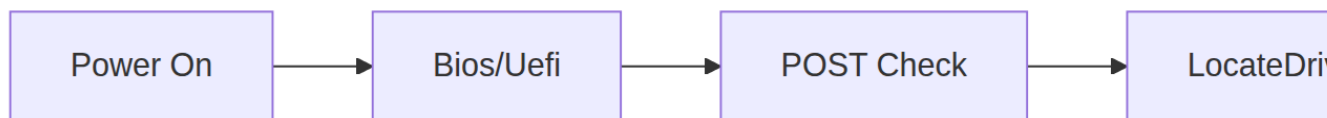
File Systems

File systems determine how data is stored within partitions:

| File System | OS | Use Case |
|--------------|----------------|----------------------------|
| NTFS | Windows | System drives, large files |
| FAT32 | Cross-platform | USB drives, compatibility |
| ext4 | Linux | Linux system drives |
| APFS | macOS | macOS system drives |

Boot Process Summary

| Phase | Component | Purpose |
|-------|----------------------|--------------------------|
| 1 | Firmware (BIOS/UEFI) | Hardware initialization |
| 2 | POST | Hardware verification |
| 3 | Bootloader | OS selection and loading |
| 4 | OS Kernel | System initialization |



Frequently Asked Questions

Q1: What's the difference between BIOS and UEFI?

| Feature | BIOS | UEFI |
|------------------------|-----------|--------------------|
| Interface | Text-only | Graphical possible |
| Storage Support | 2 TB max | No practical limit |
| Security | Basic | Secure Boot |
| Speed | Slower | Faster |

Q2: What's the difference between partition style and file system?

Partition Style: Defines how the drive is divided (MBR vs GPT) **File System:** Defines how files are stored within each partition (NTFS, ext4, etc.)

Q3: Can I dual boot multiple operating systems?

Yes, by:

- Installing each OS on separate partitions
- Using a bootloader that detects all systems
- Selecting which OS to boot at startup

Resources

Video Tutorials

- [Boot Process \(English\)](#)
- [Boot Process \(Hindi\)](#)
- [Windows Partitions \(Hindi\)](#)

WSL Setup

Remove Previous Installed Distributions

```
wsl --list --verbose
```

For Each Listed distribution - `wsl --unregister <DistributionName>` - Open Settings → Apps → Installed apps, Find each Linux distribution, click the three-dot menu, and select Uninstall

Update WSL → Ubuntu Installation

```
wsl --update
```

```
wsl --list --online
```

```
wsl --install Ubuntu-24.04 -> Install Ubuntu
```

```
sudo apt update && sudo apt upgrade -y -> Update Ubuntu
```

```
PS C:\Users\hrith> wsl --install Ubuntu-24.04
Downloading: Ubuntu 24.04 LTS
Installing: Ubuntu 24.04 LTS
Distribution successfully installed. It can be launched via 'wsl.exe -d Ubuntu-24.04'
Launching Ubuntu-24.04...
Provisioning the new WSL instance Ubuntu-24.04
This might take a while...
Create a default Unix user account: hrm
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

hrm@bitnd:/mnt/c/Users/hrith$ sudo apt update && sudo apt upgrade -y
[sudo] password for hrm:
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
```

Install JavaScript Tools

- nvm install visit and run bash script <https://github.com/nvm-sh/nvm>
- node install <https://nodejs.org/en/download>
 - `nvm install 22`
 - `nvm list nvm use 22 nvm current`
 - `corepack enable yarn`
 - `corepack enable pnpm` > Check
- `nvm -v`
- `node -v`
- `npm -v npx -v`
- `pnpm -v`
- `yarn -v`

Install Python Tools

1) Python VENV

- `python3 -m venv .venv -> Copy error code and run sudo apt install python3.12-venv`

- 2) PIPX
 - `sudo apt install pipx`
- 3) UV - Rust-based Python package installer
 - `pipx install uv` It will maintain isolation
- 4) LLM
 - `pipx install llm -> pipx ensurepath`
 - Configure it
 - `llm install llm-gemini` or `llm install llm-ollama`
 - `llm keys set gemini`
 - `llm -m gemini-2.0-flash 'Tell me fun facts about Mountain View'`
- 5) MiniConda
 - Download `.sh` <https://www.anaconda.com/download/success>
 - `bash <pathto .sh file>`
 - `conda config --set auto_activate_base false`

Install C/C++ Tools

`sudo apt install build-essential - gcc -> The C compiler - g++ -> The C++ compiler`

Check - `gcc -v` - `gcc --version` - `g++ -v` - `g++ --version`

Install Java Tools

`sudo apt install default-jdk`

This command installs:

Java Development Kit (JDK) - Compiler, debugger, and development tools

Java Runtime Environment (JRE) - Required to run Java applications

Java Virtual Machine (JVM) - Core execution environment

- Configure JAVA_HOME Environment Variable
 - `echo 'export JAVA_HOME="/usr/lib/jvm/default-java"' >> ~/.bashrc`
 - ## confirm the path first using below update-alternative... command
 - restart shell
- Install other versions of java
 - `sudo apt install openjdk-17-jdk`
- Set Default Java/Javac installed version
 - `sudo update-alternatives --config java`
 - `sudo update-alternatives --config javac`

Check

- `java --version` - `javac --version` - `echo $JAVA_HOME`

Use NPX

- `npm install -g promptfoo` then `npx promptfoo view`