

Technical Writeup: Linux Boot Process — From BIOS to Shell

The Linux boot process is a sequence of steps that your system follows to transition from powered-off hardware to a fully functional Linux environment, complete with a shell. Understanding each stage is crucial for system administrators, developers, and security professionals.

2. Bootloader: GRUB or systemd-boot

The bootloader is responsible for loading the Linux kernel into memory.

GRUB (GRand Unified Bootloader)

- Reads its configuration file: `/boot/grub/grub.cfg`
- Presents a menu to choose the kernel (if multiple available).
- Loads the selected kernel image (`linux-<version>`) and **initramfs** (initial RAM filesystem) into memory.
- Passes control to the kernel with selected parameters (e.g., root filesystem).

systemd-boot (alternative to GRUB on UEFI systems)

- Simpler than GRUB, reads `/boot/loader/loader.conf` and entries in `/boot/loader/entries/`.

4. init System: PID 1

The init system is the first userspace process (PID 1) that manages user-space services.

Options:

- **Systemd** (default in most modern distros): Reads configuration from `/etc/systemd/`
- **SysVinit** (older): Executes scripts in `/etc/init.d/`
- **Upstart** (legacy Ubuntu): Uses `.conf` files in `/etc/init/`

What init does:

- Mounts remaining filesystems.
- Starts essential services (udev, networking, cron, syslog).
- Spawns **getty** processes to manage TTYs (terminals).
- Launches the **default target** (multi-user, graphical, etc.).



Summary Diagram

```
[Power On]
  ↓
[BIOS / UEFI]
  ↓
[Bootloader (GRUB / systemd-boot)]
  ↓
[Kernel + initramfs]
  ↓
[Init system (systemd, SysV)]
  ↓
[Login Prompt (TTY or GUI)]
  ↓
[Shell or Desktop Environment]
```



Final Notes

- **Fast Boot** options in UEFI/BIOS can skip POST checks.
- **Secure Boot** restricts unsigned kernel/modules — must be configured correctly for custom kernels.
- **initrd vs initramfs**: `initrd` is a block device image, `initramfs` is a cpio archive in RAM.