

4. How a computer starts up (Raspberry Pi)

Boot Process Overview

The boot process of a Raspberry Pi can be divided into several stages:

1. Power On
2. GPU Firmware Execution
3. Boot Loader Execution
4. Kernel Loading
5. Init System Execution

Let's dive into each stage in more detail.

1. Power On

When you plug in the power supply or connect a charged battery to your Raspberry Pi, it begins receiving power. The Raspberry Pi's System on a Chip (SoC) detects the stable power supply and starts the boot process.

2. GPU Firmware Execution

The SoC's GPU (Graphics Processing Unit) is the first component to start running. It reads the first stage bootloader, known as `bootcode.bin`, from the SD card. This bootloader is responsible for initializing the GPU and loading the next stage bootloader.

3. Boot Loader Execution

The next stage bootloader, typically named `start.elf`, is loaded by the GPU. This bootloader initializes the RAM, sets up the CPU, and reads the configuration file `config.txt`. It then loads the kernel image, usually named `kernel.img`, into memory.

```
Ubuntu 8.04, kernel 2.6.24-16-generic
Ubuntu 8.04, kernel 2.6.24-16-generic (recovery mode)
Ubuntu 8.04, memtest86+
```

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.

4. Kernel Loading

Once the kernel image is loaded into memory, the GPU passes control to the CPU. The CPU then decompresses the kernel image and starts executing it. The kernel initializes the hardware, sets up memory management, and mounts the root filesystem.

```
x64 irq 1,12
[ 1.085227] serio: i8042 KBD port at 0x60,0x64 irq 1
[ 1.086072] serio: i8042 AUX port at 0x60,0x64 irq 12
[ 1.087317] mousedev: PS/2 mouse device common for all mice
[ 1.089087] input: AT Translated Set 2 keyboard as /devices/platform/i8042/serio0/input/input0
[ 1.091075] rtc_cmos rtc_cmos: registered as rtc0
[ 1.091929] rtc_cmos rtc_cmos: alarms up to one day, 114 bytes nvram
[ 1.094206] device-mapper: uevent: version 1.0.3
[ 1.096061] device-mapper: ioctl: 4.39.0-iocli (2018-04-03) initialised: dm-devel@redhat.com
[ 1.099224] Initializing XFRM netlink socket
[ 1.100206] NET: Registered protocol family 17
[ 1.101236] Key type dns_resolver registered
[ 1.103166] AUX version of gcm_enc/dec engaged.
[ 1.104095] AES CTR mode by8 optimization enabled
[ 1.132208] sched_clock: Marking stable (1132194021, 0)->(1696167804, -563973783)
[ 1.134349] registered taskstats version 1
[ 1.135110] Loading compiled-in X.509 certificates
[ 1.143821] Key type encrypted registered
[ 1.887377] tsc: Refined TSC clocksource calibration: 2493.720 MHz
[ 1.888969] clocksource: tsc: mask: 0xffffffffffffffff max_cycles: 0x23f20d369de, max_idle_ns: 440795277732 ns
```

5. Init System Execution

After the kernel has finished its initialization process, it starts the init system, which is responsible for starting and managing system services and user processes. On most Raspberry Pi systems running Linux, the default init system is **systemd**.

```
[FAILED] Failed to start Load Kernel Modules.
See 'systemctl status systemd-modules-load.service' for details.
Starting Apply Kernel Variables...
Mounting FUSE Control File System...
[ OK ] Mounted FUSE Control File System.
[ OK ] Started Apply Kernel Variables.
[ OK ] Started Create Static Device Nodes in /dev.
Starting udev Kernel Device Manager...
[ OK ] Started File System Check on Root Device.
Starting Remount Root and Kernel File Systems...
[ OK ] Started Remount Root and Kernel File Systems.
[ OK ] Reached target Local File Systems (Pre).
Starting Flush Journal to Persistent Storage...
Starting Load/Save Random Seed...
[ OK ] Started Flush Journal to Persistent Storage.
[ OK ] Started Load/Save Random Seed.
[ OK ] Started udev Kernel Device Manager.
Starting Show Plymouth Boot Screen...
```

Raspberry Pi File System on SD Card

The Raspberry Pi uses an SD card as its primary storage device. The SD card contains the necessary files for the boot process and the operating system. A typical Raspberry Pi SD card has the following file system structure:

- ◆ **/boot** : This partition contains the files required for booting, such as **bootcode.bin**, **start.elf**, **config.txt**, and **kernel.img**.

```
jush@ranedeer:/boot $ ls
bcm2708-rpi-b.dtb      bcm2710-rpi-2-b.dtb      bcm2711-rpi-cm4.dtb      fixup4.dat      kernel7.img      start4.elf
bcm2708-rpi-b-plus.dtb  bcm2710-rpi-3-b.dtb      bcm2711-rpi-cm4-io.dtb   fixup4db.dat    kernel7l.img     start4x.elf
bcm2708-rpi-b-rev1.dtb  bcm2710-rpi-3-b-plus.dtb  bcm2711-rpi-cm4s.dtb    fixup4x.dat     kernel8.img      start_cd.elf
bcm2708-rpi-cm.dtb      bcm2710-rpi-cm3.dtb      bootcode.bin            fixup_cd.dat    kernel.img        start_db.elf
bcm2708-rpi-zero.dtb    bcm2710-rpi-zero-2.dtb    cmdline.txt             fixup.dat       LICENCE.broadcom  start.elf
bcm2708-rpi-zero-w.dtb  bcm2710-rpi-zero-2-w.dtb  config.txt              fixup_db.dat    overlays          start_x.elf
bcm2709-rpi-2-b.dtb     bcm2711-rpi-400.dtb      COPYING.linux           fixup_x.dat     start4cd.elf      'System Volume Information'
bcm2709-rpi-cm2.dtb     bcm2711-rpi-4-b.dtb      fixup4cd.dat           issue.txt       start4db.elf
```

- ◆ **/** : The root partition contains the Linux operating system files and directories, such as **/bin**, **/etc**, **/home**, and **/usr**.

```
jush@ranedeer:/ $ ls
bin  boot  dev  etc  home  lib  lost+found  media  mnt  opt  proc  root  run /sbin  srv  sys  tmp  usr  var
jush@ranedeer:/ $
```

Raspberry Pi Booting

To boot a Raspberry Pi, follow these steps:

1. Write the operating system image to an SD card using a tool like Etcher or Raspberry Pi Imager.
2. Insert the SD card into the Raspberry Pi's SD card slot.
3. Connect the necessary peripherals, such as a keyboard, mouse, and monitor.
4. Plug in the power supply or connect a charged battery to power on the Raspberry Pi.

The Raspberry Pi will then go through the boot process as described earlier and load the operating system from the SD card.

See Also [5. Evolution of Operating Systems](#)