

# Booting Up and Shutting Down

tsaimh (2022-2025, CC BY-SA)

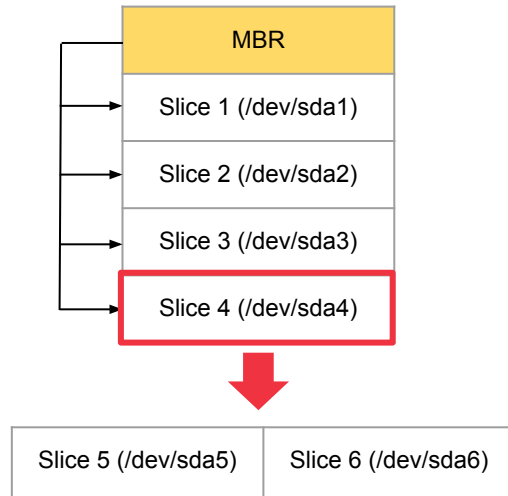
lctseng (2019-2021, CC BY-SA)

# Outline

- Partitioning - MBR / GPT
- Booting Phases
  - BIOS / UEFI
  - Boot Loader / Boot Manager
  - Kernel Initialization
  - Early Userspace
  - Late Userspace
- Systemd Target
- initramfs-tools
- GRUB (Grand Unified bootloader)
- Single User Mode
- Multibooting

# MBR - Master Boot Record

- First 512 bytes of disk, outside the regular filesystem
  - Last 2 Bytes are 0x55AA
  - First 446 bytes - code for booting
  - Following 64 bytes - partition table
  - Responsible to find the boot code on the boot sector of bootable slice

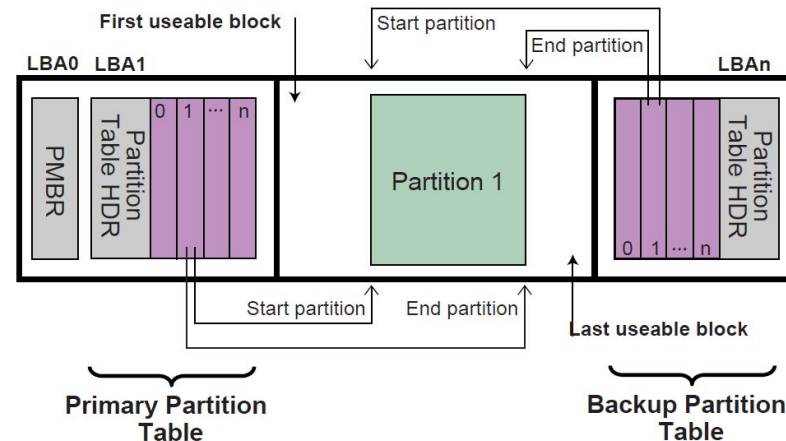


```
$ head -c 512 /dev/sda | xxd
00000000: eb63 9000 0000 0000 0000 0000 0000 0000  .c.....
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  .....
...
000001d0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000  .....
000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  .....U.
```

# GPT - GUID Partition Table

- The first logical block (LBA 0) contains a **protective MBR**
- Two GPT Header are stored on the device
  - Primary: located in LBA 1
  - Backup: located in the last LBA of the device
- The GPT Header defines the range of LBAs that are usable by GPT

## Partition Entries



# Booting Phases

## 1. UEFI / BIOS

- Setup some hardware and run program from boot sector

## 2. BootLoader

- Find your operating system and starts it
- Provide the necessary files for early userspace to start the late userspace

## 3. Early userspace

- Setup **root filesystem**, block devices, kernel modules

## 4. Late userspace

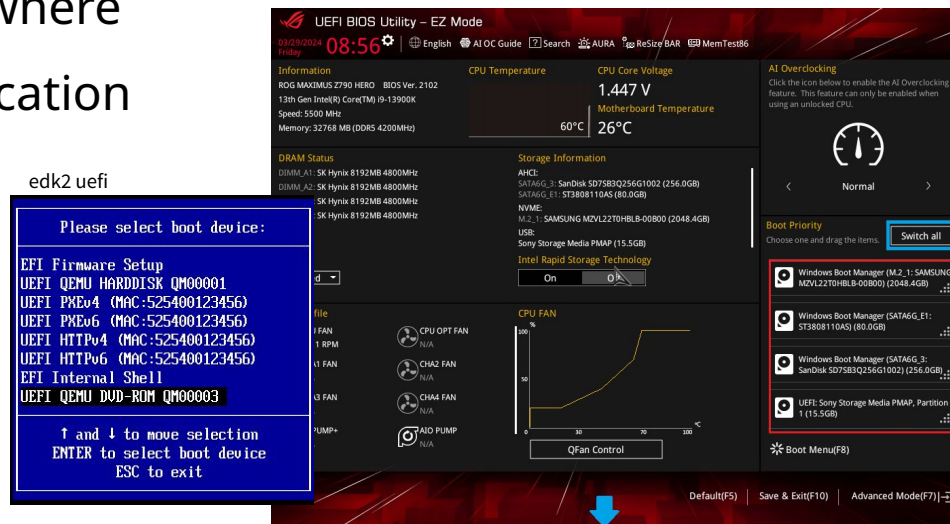
- Virtual terminal, login shell, graphical session

# Booting Phases - BIOS

- Initialize the hardware (disk, keyboard controller, etc.)
- BIOS launches the first 440 bytes of the first disk in the BIOS disk order
- The boot loader's first stage in the MBR boot code then launches its second stage code (if any) from either
  - Next disk sectors after the MBR, i.e. post-MBR gap (only on a MBR partition table)
  - In a BIOS boot partition (used in GPT)
- The actual boot loader is launched

# Booting Phases - UEFI

- Initialize the hardware (disk, keyboard controller, etc.)
- Firmware reads the boot entries in the NVRAM to determine which EFI application to launch and from where
- Firmware launches the EFI application
  - Boot loader
  - Kernel (using EFI boot stub)
  - UEFI shell
  - Boot manager (e.g. systemd-boot)



source: <https://www.asus.com/hk/support/faq/1053205/>

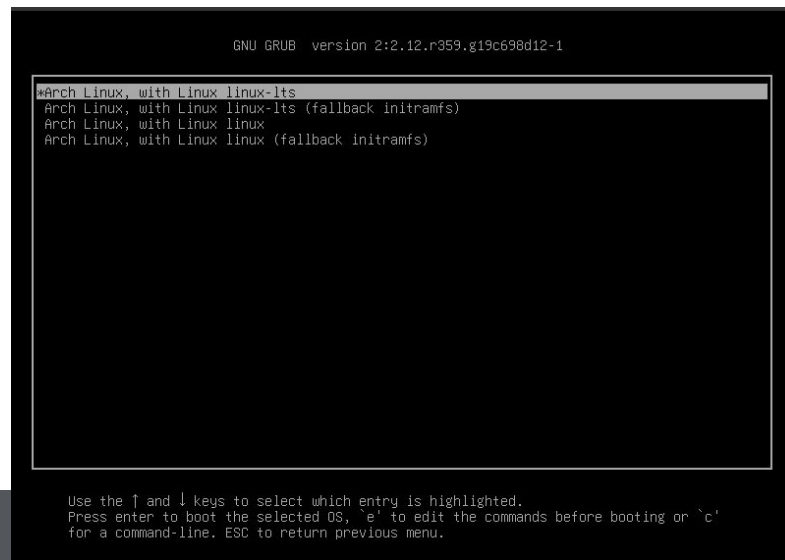
# Booting Phases - Boot Loader / Boot Manager

- Responsible for loading the kernel with kernel parameters and (if any) initramfs images
- Boot manager presents a menu of boot options or provides some other way to control the boot process.

systemd-boot (from ArchLinux installation ISO)



GRUB with multiple boot options (different kernel)



# Booting Phases - Kernel Initialization

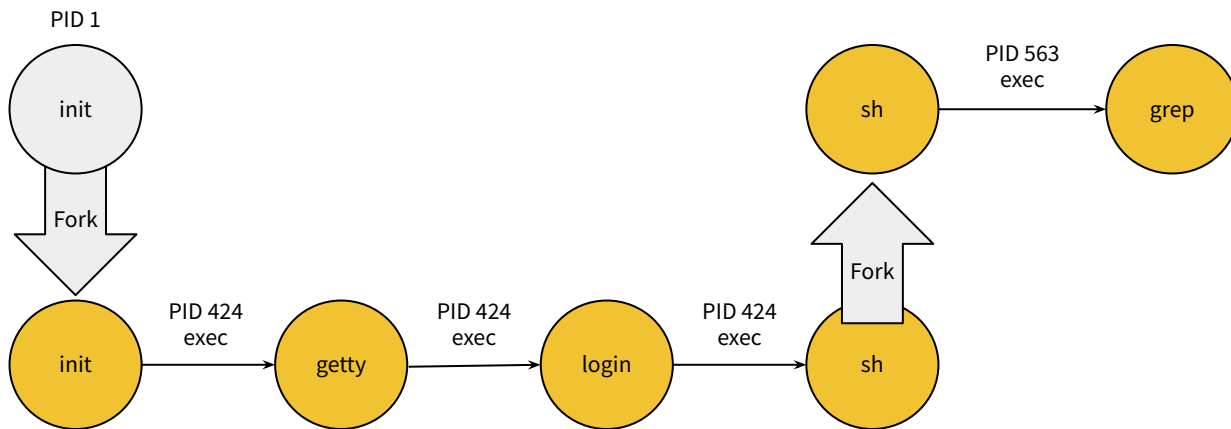
- Boot loader boots the vmlinux image containing the kernel
  - vmlinux image is a statically linked executable file that contains the Linux kernel
  - vmlinux image often placed in `/boot/vmlinuz-linux`
  - The letter z at the end denotes that it is compressed
- Performs hardware enumeration and initialization
- Unpacks its builtin initramfs into the **temporary root directory**

# Booting Phases - Early Userspace

- Starts by the kernel executing the /init binary in **the temporary rootfs** as PID 1
- Setup storage stack for the real rootfs (e.g. dm-crypt, mdadm, lvm)
- Resolve persistent block device names to real device through udev
- Handle decryption of the the real rootfs
- Load the DRM module
- Mount the real root and switch root

# Booting Process - Late Userspace

- Starts by executing the init program in **the real rootfs** as PID 1
- Debian use systemd by default
- Done booting after reach multi-user.target or graphical.target



# Systemd Target

- Systemd tasks are organized as **units** (e.g. ssh.service)
- **Targets** are groups of units
- Targets call units to put the system together
  - graphical.target calls all units that are necessary to start a workstation with graphical user interface
- Targets can build on top of another or depend on other targets
- At boot time, systemd activates the target **default.target**
  - Alias for another target, e.g. graphical.target
  - Try `systemctl get-default`

# Common systemd targets

| SysV Runlevel | systemd Target        | Notes                                                                                              |
|---------------|-----------------------|----------------------------------------------------------------------------------------------------|
| 0             | poweroff.target       | Halt the system                                                                                    |
| 1             | rescue.target         | Single user mode                                                                                   |
| 2, 3, 4       | multi-user.target     | Multi-user, non-graphical, users can usually login via multiple consoles or via the <b>network</b> |
| 5             | graphical.target      | Multi-user, graphical                                                                              |
| 6             | reboot.target         | Reboot                                                                                             |
| -             | emergency.target      | Emergency shell                                                                                    |
| -             | network-online.target | Network is online                                                                                  |

```

Welcome to Fedora 20 (Heisenbug)!

[ OK ] Reached target Remote File Systems.
[ OK ] Listening on Delayed Shutdown Socket.
[ OK ] Listening on /dev/initctl Compatibility Named Pipe.
[ OK ] Reached target Paths.
[ OK ] Reached target Encrypted Volumes.
[ OK ] Listening on Journal Socket.
      Mounting Huge Pages File System...
      Mounting POSIX Message Queue File System...
      Mounting Debug File System...
      Starting Journal Service...
[ OK ] Started Journal Service.
      Mounting Configuration File System...
      Mounting FUSE Control File System...
[ OK ] Created slice Root Slice.
[ OK ] Created slice User and Session Slice.
[ OK ] Created slice System Slice.
[ OK ] Reached target Slices.
[ OK ] Reached target Swap.
      Mounting Temporary Directory...
[ OK ] Reached target Local File Systems (Pre).
      Starting Load Random Seed...
      Starting Load/Save Random Seed...
[ OK ] Mounted Huge Pages File System.
[ OK ] Mounted POSIX Message Queue File System.
[ OK ] Mounted Debug File System.
[ OK ] Mounted Configuration File System.
[ OK ] Mounted FUSE Control File System.
[ OK ] Mounted Temporary Directory.
[ OK ] Started Load Random Seed.
[ OK ] Started Load/Save Random Seed.
[ OK ] Reached target Local File Systems.
      Starting Recreate Volatile Files and Directories...
      Starting Trigger Flushing of Journal to Persistent Storage...
[ OK ] Started Recreate Volatile Files and Directories.
      Starting Record System Reboot/Shutdown in UTMPT...
[ OK ] Started Trigger Flushing of Journal to Persistent Storage.
[ OK ] Started Record System Reboot/Shutdown in UTMPT.
[ OK ] Reached target System Initialization.
[ OK ] Reached target Timers.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Basic System.
      Starting Permit User Sessions...
      Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
      Starting Login Service...
      Starting Cleanup of Temporary Directories...
[ OK ] Started Permit User Sessions.
[ OK ] Started Cleanup of Temporary Directories.
      Starting Console Getty...
[ OK ] Started Console Getty.
[ OK ] Reached target Login Prompts.
[ OK ] Started Login Service.
[ OK ] Reached target Multi-User System.

Fedora release 20 (Heisenbug)
Kernel 3.9.2-200.fc18.x86_64 on an x86_64 (console)

fedora login:

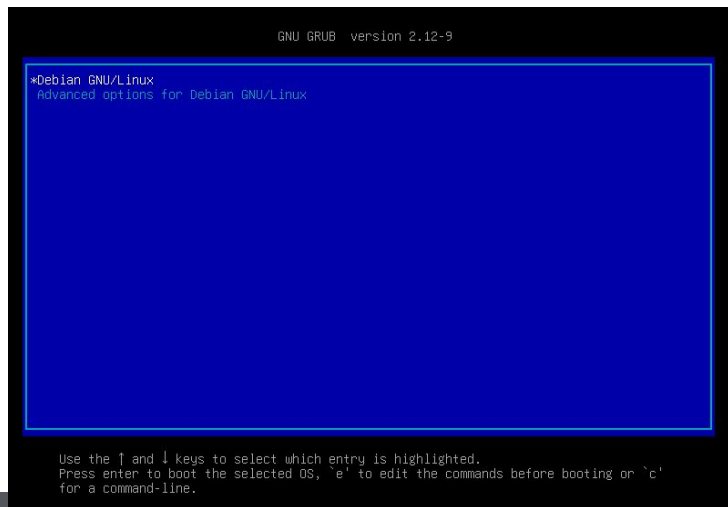
```

# initramfs-tools

- Debian use the initramfs managed by the initramfs-tools package
- Config with `/etc/initramfs-tools/initramfs.conf` or `/etc/initramfs-tools/conf.d/`
- Contains config such as **modules, devices, compress, fstype**
- Update initramfs using `update-initramfs -u -v`

# GRUB (GRand Unified bootloader)

- Use GRUB v2 by default
- Configurations are placed in `/etc/default/grub`
  - The most used config is `GRUB_CMDLINE_LINUX_DEFAULT`
  - e.g. remove the `quiet` parameter will cause systemd print more info during booting
- Update using `update-grub`



# Single User Mode

- Similar to Windows “Safe Mode”
- Repair system
  - Inconsistent file system
  - Error in a boot configuration
- Reset lost root password
  - Entering single user mode in some ways requires no password
- Full access to local file system and configuration (root permission)
- No network access

# Boot in single user mode

- Using GRUB for example
- Press “e” to edit boot parameters in boot menu
- Append “single” in “linux” line
- Ctrl+x to boot

```
GNU GRUB  version 2.12-9

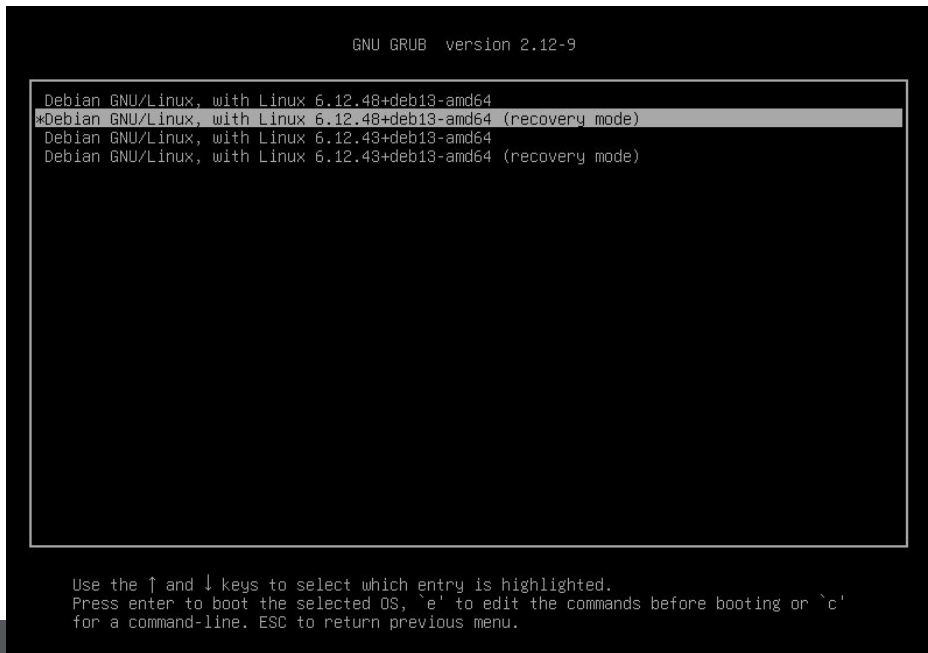
setparams 'Debian GNU/Linux'

load_video
insmod gzio
if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
insmod part_msdos
insmod ext2
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1
  \
si --hint-baremetal=ahci0,msdos1  de4a04ee-4de1-47dc-b302-0a64766139f1
else
  search --no-floppy --fs-uuid --set=root de4a04ee-4de1-47dc-b302-0a64766139f1
fi
echo      'Loading Linux 6.12.48+deb13-amd64 ...'
linux    /boot/vmlinuz-6.12.48+deb13-amd64 root=UUID=de4a04ee-4de1-47dc-b302-0a64766139f1 ro quiet single_
echo      'Loading initial ramdisk ...'
initrd   /boot/initrd.img-6.12.48+deb13-amd64

Minimum Emacs-like screen editing is supported. TAB lists completions. Press Ctrl-x
or F10 to boot, Ctrl-c or F2 for a command-line or ESC to discard edits and return
to the GRUB menu.
```

## Boot in single user mode (Cont.)

- **OR**, choose “Advanced options for Debian GNU/Linux
- Choose option with “(recovery mode)”



```
GNU GRUB version 2.12-9

Debian GNU/Linux, with Linux 6.12.48+deb13-amd64
*Debian GNU/Linux, with Linux 6.12.48+deb13-amd64 (recovery mode)
Debian GNU/Linux, with Linux 6.12.43+deb13-amd64
Debian GNU/Linux, with Linux 6.12.43+deb13-amd64 (recovery mode)

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. ESC to return previous menu.
```

# Using Single User Mode

- Single user mode requires password by default
- However, use `rw` and `init=/bin/bash` will start shell directly and spawn a shell for you to modify files on disks

```
/dev/sda1: clean, 39247/493856 files, 353136/1973504 blocks
You are in rescue mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, or "exit"
to continue bootup.rg) 00:03.0 C900 PCI2.10 PnP PMM+BEFD40B0+BEF340B0 C900
Enter root password for system maintenance
(or press Control-D to continue): _
```

# Multibooting

- Each OS can maintain its own files within the EFI partition
- UEFI can launch different EFI applications
- FYR
  - <https://wiki.debian.org/DualBoot/Windows>
  - [https://wiki.archlinux.org/title/Dual boot with Windows](https://wiki.archlinux.org/title/Dual_boot_with_Windows)

# Halt and Poweroff

- Halt
  - Terminate all process, write data back to disks
  - When everything is ready, tell user to turn off the power
  - In older systems, you need to manually do so
  - `systemctl halt` or `/usr/sbin/halt`
- Poweroff
  - Halt + Turn off the power
  - ACPI (Advanced Configuration and Power Interface) / APM (Advanced Power Management)
  - `systemctl poweroff` or `/usr/sbin/poweroff`

## Other system / service managers

- OpenRC
- FreeBSD RC
- System-V
- Runit
- Supervisord

# Reference

- BootProcess - Debian Wiki - <https://wiki.debian.org/BootProcess>
- Initramfs - Debian Wiki - <https://wiki.debian.org/initramfs>
- 5. GUID Partition Table (GPT) Disk Layout - UEFI Specification 2.10 - [https://uefi.org/specs/UEFI/2.10/05 GUID Partition Table Format.html#](https://uefi.org/specs/UEFI/2.10/05_GUID_Partition_Table_Format.html#)
- Arch boot process - ArchWiki - [https://wiki.archlinux.org/title/Arch\\_boot\\_process](https://wiki.archlinux.org/title/Arch_boot_process)
- systemd - ArchWiki - <https://wiki.archlinux.org/title/Systemd>
- systemd - Debian Wiki - <https://wiki.debian.org/systemd>
- System and Service Manager - <https://systemd.io/>