# Determine and configure hardware settings

An **operating system (OS)** is system software that manages computer hardware, and software resources, and provides common services for computer programs.

**Firmware** is the software on your hardware that runs it; Think of it as a built-in OS or driver for your hardware.

**BIOS** (Basic Input/Output System). Older, you can configure it from a text menu-based system and boot the computer from a bootloader first sector of the first partition of your hard disk (MBR).

**UEFI** (Unified Extensible Firmware Interface). Started as EFI in 1998 at Intel. Now the standard. Uses a specific disk partition for boot (EFI System Partition (ESP)) and uses FAT. On Linux it is on /boot/efi and files are .efi. You need to register each bootloader.

## Peripheral Devices

**PCI** (Peripheral Component Interconnect). Allows hardware boards to be added to the motherboard.

**USB** (Universal Serial Bus). Serial and need fewer connections.

**GPIO** (General Purpose Input Output) To control other devices.

## In the Linux folders

**SysFS** (/sys) is a pseudo file system provided by the Linux kernel that exports information about various kernel subsystems, hardware devices, and associated device drivers from the kernel's device model to user space through virtual files.

root@funlife:~# ls /sys

All block devices are at the block and bus directory has all the connected PCI, USB, serial, ... devices. Note that here in sys we have the devices based on their technology but /dev/ is abstracted.

**UDev** (/dev) is a device manager for the Linux kernel.

If a program wants to read from or write to a device, it will use the corresponding file in /dev to do so.

**DBus** (/bus) is a message bus system, a simple way for applications to talk to one another.

**Proc** (/proc) is where the Kernel keeps its settings and properties.

You can find things like:

* IRQs (interrupt requests)
* I/O ports (locations in memory where CPU can talk with devices)
* DMA (direct memory access, faster than I/O ports)
* Processes
* Network Setting

Ex.: Turn LED on and off by writing here:

root@funlife:/proc/acpi/ibm# echo on > light

root@funlife:/proc/acpi/ibm# echo off > light

Another very useful directory here, is /proc/sys/net/ipv4 which controls real-time networking configurations.

**#lsusb** Shows PCI devices that are connected to the computer.

**#lsusb** Shows all the USB devices connected to the system.

**#lshw** Shows hardware.

**#lsblk** Used for list devices that can read from or write to by blocks of data.

## Loadable Kernel Modules

Loadable kernel modules (.ko files) are object files that are used to extend the kernel of the Linux Distribution. They are used to provide drivers for new hardware like IoT expansion cards that have not been included in the Linux Distribution.

You can inspect the modules using the lsmod or manage them via modprobe commands.

**#lsmod** Shows kernel modules. They are located at /lib/modules.

These are the kernel modules that are loaded. Use modinfo to get more info about a module, If you want.

If you need to add a module to your kernel (say a new driver for hardware) or remove it (uninstall a driver) you can use rmmod and modprobe.

Resource: [LPIC 01 Prof. Jadi Mirmirani](https://linux1st.com/)

Thanks a lot [Jadi](https://jadi.net/)

[](https://www.buymeacoffee.com/yebneddin)