System's view of OS
- resource allocator
- control program

~~OS~~ User's view of OS
- ease of use
- resource utilization

} OS

OS = Kernal
- ↗ system programs (not necessarily part of the kernel)
- ↘ application programs (all programs not associated with the operation of the system)

After giving the power to pc;

1. Power Supply Unit (PSU) checks and sends a Power Good Signal to the motherboard.

2. CPU starts execution from predefined memory location 0xFFFF0 ~~(magic address)~~, which hardcoded. to
   - * JMP instruction
   - * jumps to the entry point of BIOS/UEFI (Modern)

   The BIOS is mapped to the TOP of the 1MB address space. (0xF0000 - 0xFFFF)

3. BIOS
   - 16-bit Real Mode.
   1. Run POST (Power-On Self-Test)
      checks CPU, RAM, GPU, Storage & peripherals.
   2. Initializes IVT at 0x0000 (RAM) (Interrupt Vector Table)

   Table containing pointers to Interrupt Service Routines (ISRs).
   Size = 1KB each entry is 4 bytes.
   (IVT allows software to interact with hardware via interrupts during the POST)

   3. Hardware detection (HDD, SSD, USB, CD-ROM)
      for each device, issues INT 0x13 calls. if the device responds correctly, it is marked as bootable.
   4. Boot device selection.
      follows a boot order ~~and~~ to scan devices for a valid MBR. (Master Boot Record)

If an MBR is found;

First 512 bytes of the boot device
- 446 bytes - Bootloader code
- 64 bytes - partition table (16 x 4) 4 = no of partitions
- 2 bytes - Magic number 0x55AA (boot signature)

This indicates that this is the boot sector. ↑ little endian format.

(4.) Kernel Loading & Initialization.

4.1 Kernel Loading

4.2 CPU Mode switching.

Real Mode (16bit) ⟶ Protected Mode (32-bit) ⟶ Long Mode (64-bit)

↑ used only during boot

Ring levels activated.

kernel runs here ⟶ Ring 0 - direct hardware access
- page tables, segmentations has done here.

users are commonly use this. ~~used~~ - Ring 3 - to access kernel functions must use system calls (Syscall / int 0x80)

4.3. Kernel initialization steps.

* Hardware Abstraction Layer initializes CPU, memory & devices.
* Mounts root file system ⟶ Linux → /
  Windows → C:\
* Starts the first user-space process. ⟶ windows - Smss.exe
  ⟶ Linux - Init {systemd (Modern)
  ↑ PID 1

(5.) User space Initialization

In Linux Kernel directly executes init

Init → mounts file system (/proc, /sys, /dev)
Starts critical services like networking, logging.
Launches a login prompt.