# Operating Systems

The layers in a computer.

- **Hardware** provides basic computing resources.
- **OS** controls and coordinate resources.
- **Applications** and **users**.

### Bus Hierarchy

- The **processor bus** is the widest and fastest for CPU to talk to cache.
- The **memory bus** talks to memory.
- The **PCI bus** communicate with devices.
- **Bridges** forwards a single from a bus to another.

### Booting

1. Bootstrap program runs when computer powers on:

   A small part of the CPU/board stores instructions to tell the CPU how to
   - Access the memory.
   - Initialise the bus and talk to devices.
2. Kernel
3. Normal operation of a computer: communicates with memory, IO devices.

**Interrupts** are how devices communicates to the CPU, when an interrupt occurs:

1. Store the program counter
2. Jumps to interrupt service routine, a **interrupt vector** contains address to all ISR.
3. The CPU resumes

An **interrup** usually happens at an **instruction boundary**.

> **Definition**
> An **exception** is a software interrupt.

### Storage

> **Definition**
> A **word** is the computer's smallest native unit of data.

Storage is organised in order of speed

1. Registers
2. Cache

   > **Note**
   > Cache is managed transparently by the computer.
3. Main memory
4. Storage / IO devices

   > **Note**
   > Each device needs an IO driver to provide a uniform interface between the controller and kernel.

**Definitions**

- **Jitter** is the variation in latency.
- **Impedence mismatch** happens when two computers operates at different speeds
- **Caching**: high performance storage to mask the performance cost of accessing slow stuff.
- **Buffering** is a memory between two components with small differences in bandwidth.
- A **bottleneck** is the most constrained resource in system.
- A **balanced system** is where all resources are bottlenecked.

**Resource Management**

| Resource | Description |
|---|---|
| CPU | • **Multiplexes** many running programs<br>• Taking turns until the timer hits zero, then interrups |
| Memory | Prevent programs from accessing memory outside its own chunk |
| IO | • Make IO instructions privilleged.<br>• For devices accessed via memory, use memory protection mechanisms. |

---