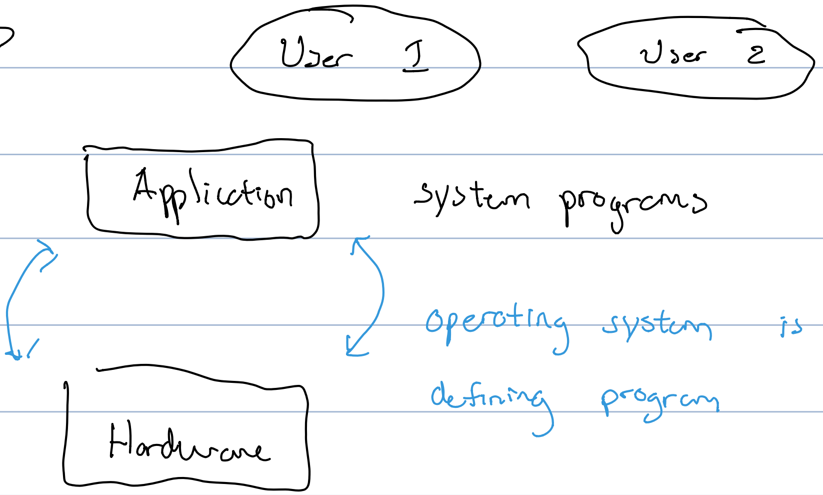
What is an OS?

Program that manages the hardware for aplication and system programs



operating system is the defining program

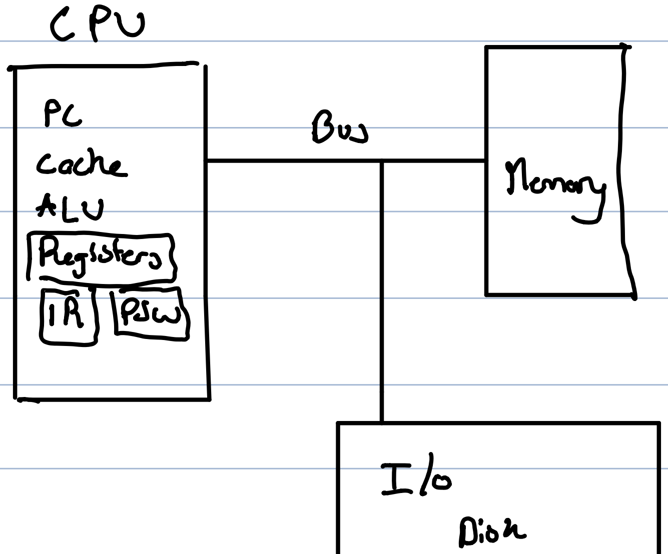
- Goal is to have ease of use

Two opposing views

- Resources vs Users
    1. Users
        -> Ease of use
    2. Systems
        → Resource Utilization

Computer System Organization

(PSW) Program Status Word



Operation

- When booted a program in firmware executes
- Initializes aspects of the system
- Loads the OS into memory
- OS executes it's first process and waits for events to happen

↖ while loop
(embeded system

  - init is the first process that executes

Events are usually signaled by an interrupt

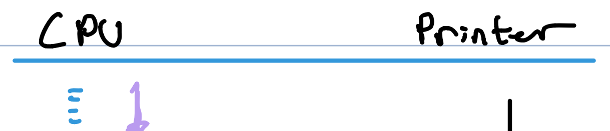
Interrupt : Mechanism in which uses software and hardware may interrupt the CPU

- Why?

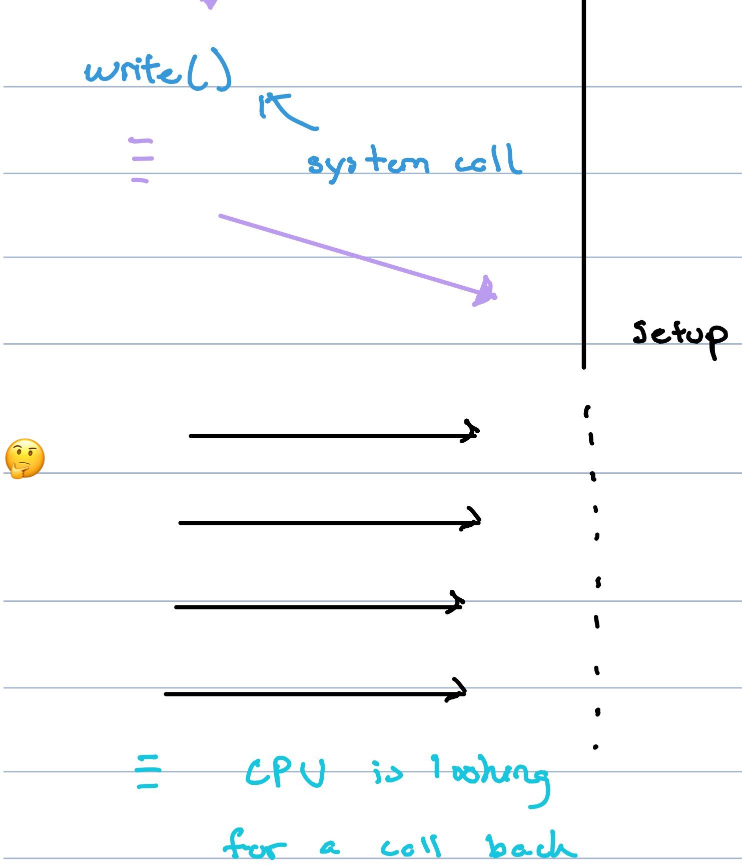
  Keeping CPU as busy as possible

  - Improve CPU utilization

  - I/O devices are much slower than CPU



the CPU is checking



to see if the device is printing or not

We don't want to use polling (Busy wait) but interrupts instead

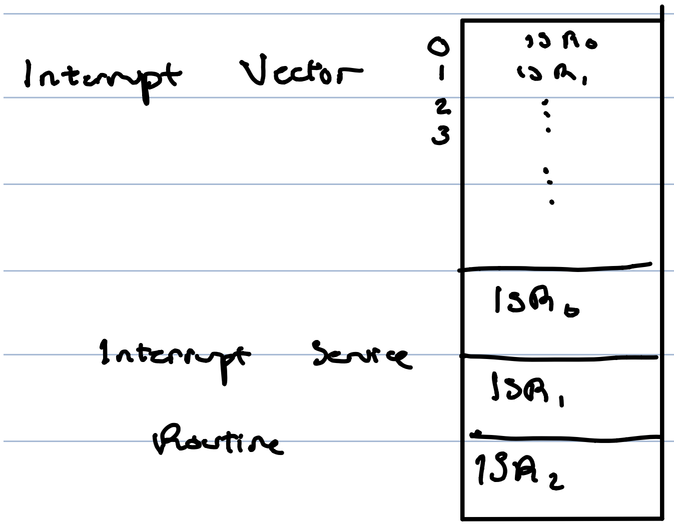
Interrupt Processing

1. An interrupt happens.
2. CPU finishes executing the current instruction
3. CPU acknowledges the interrupt
4. Store PC + PSW on the stack
5. Store all process state (registers)
6. load the PC with the ISR address
7. Execute the ISR
8. Recover the process state
9. Pop PC + PSW

Need to be stored

- PC
- PSW
- Registers

ISR → Interrupt Service Routine

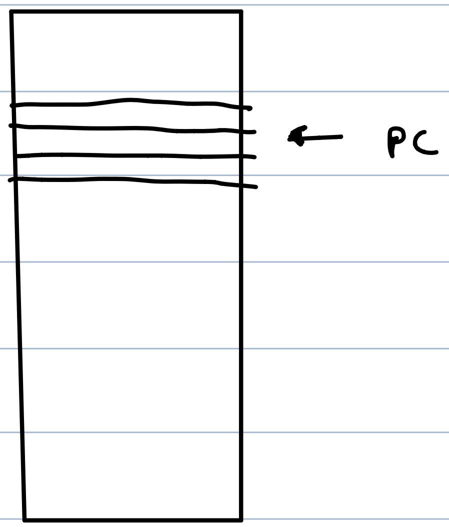


Storage

* CPU only loads instructions (RAM) from memory

* Memory is an array of words each application has an address

The instruction execution cycle



1. load Instruction into IR
2. Opcode is executed
3. Results may be stored in memory or registers

- Can't store all programs in memory
  1. Too small
  2. Volatile

Storage Hierarchy

Registers

Cache

* Wide range of I/O devices

Memory

SSD

Magnetic Disk

Optical

Tapes (Tape Drives)

→ Device drivers allow the OS to integrate such devices

* Direct Memory Access (DMA)

- Encode it in blocks then send interrupt when complete
- Transfer an entire block of data before interrupting the CPU