Overview of Architecture

- **Basic Computer Hardware**
- **How CPU handles Input Output?**
- **How Interrupts are supported?**

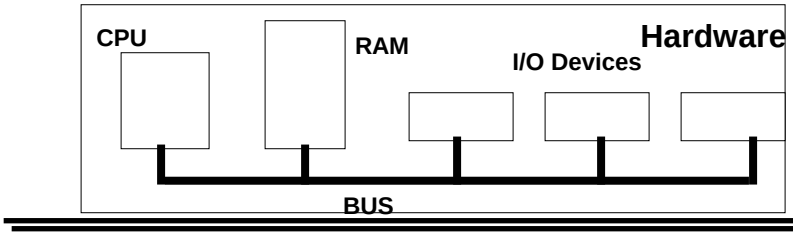
The ability of OS are dictated in part by the architecture of the machine.

- **Architectural support can greatly simplify, or greatly complicate the OS**

OS and Hardware

OS Service	Hardware Support
I/0	Interrupts & DMA
Protection	Protected Instruction
	Kernel and User modes
	Interrupt & Trap Vectors
	Base & Limit Registers
Scheduling	Timers
Synchronization	Atomic Instructions

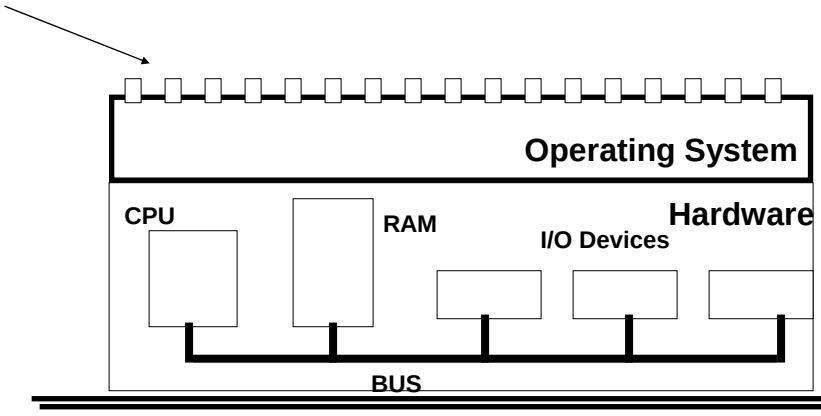
The Computer System



Os-slide#3

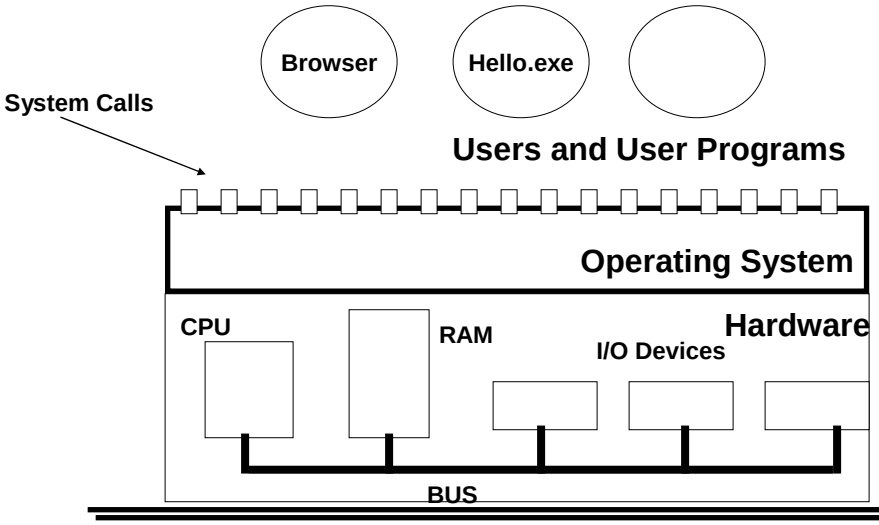
The computer system

System Calls



Os-slide#4

The computer system



Hardware revision

Architecture.
Booting the system.
Instruction execution.
I/O and interrupts.
Memory
Disks
Hardware Protection Mechanisms

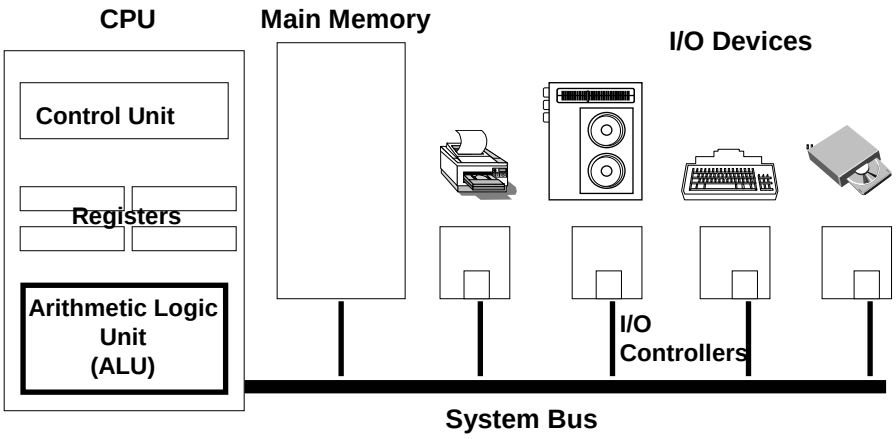
Hardware Concepts.

Most computers are based on the Von Neumann architecture.

Four main elements of hardware

- **Central Processor Unit (CPU)**
- **Main Memory**
- **I/O Modules or Devices (printers, disk drives, key board)**
- **System Interconnections (the bus)**

Basic Hardware Architecture.



CPU Registers.

Program Counter, PC

Address of NEXT instruction.

Instruction Register, IR

Contains the current instruction

Processor Status Word

Contains information about the state of the CPU, Interrupts, condition statements.

Stack Pointer, SP.

Points to the top of the process stack.

Others.

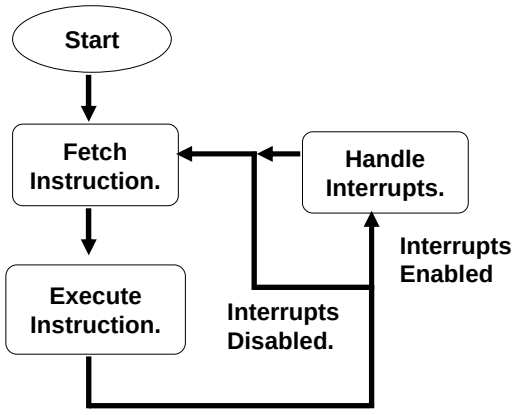
Memory management etc.

User Registers.

Used for various programming purposes.

Instruction Execution.

- **CPU, memory and device controllers connects via common bus**
- **When computer starts, it loads OS**
- **OS runs users programs as requested**
- **Constantly checks for Interrupts:**
  - **From hardware**
  - **From software**



Booting a System.

H/W looks for a boot device. (such as c:\ drive)

Boot block contains a loader program. (generally in a preset location in c:\drive)

Loader program installs OS.

OS performs various initialisation procedures

Some form of process commences (such as init)

Interrupts.

An Interrupt is a mechanism that allows the normal processing of the processor to be interrupted.

Interrupts are used to increase efficiency.

Especially between components that operate at different speeds.

Classes of Interrupts.

Interrupt Class	Cause
program	generated by a s/w condition (e.g. error, system call)
timer	generated by system clock
I/O	generated by I/O controller
hardware failure	power failure, memory parity

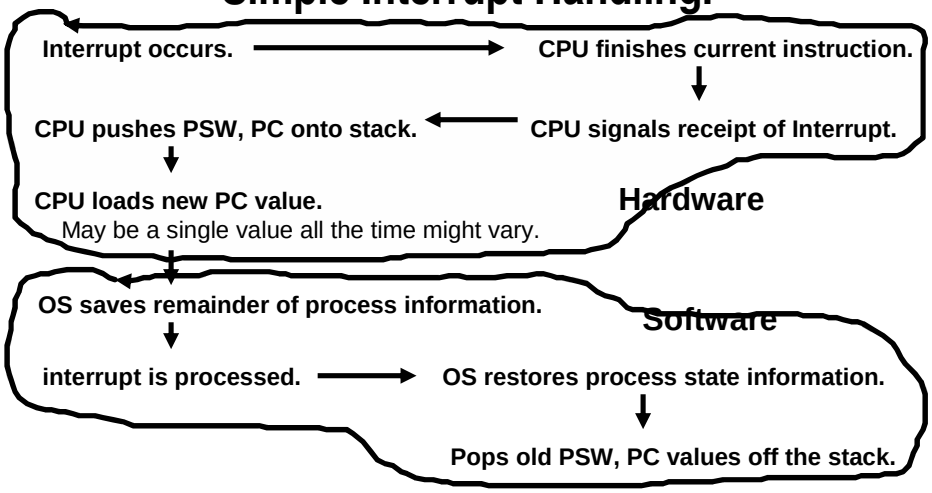
Handling Interrupts.

OS must provide some form of Interrupt handler.

Each hardware platform carries out some particular procedure for an Interrupt.

Eventually the h/w hands over to the s/w.

Simple Interrupt Handling.



Interrupt Handling

- **The Operating System Preserves the state of the CPU by storing registers and program counter**
- **Determine which type of interrupt has occurred:**
  - ♦ ***polling***
  - ♦ ***vectored interrupt***
- **Jumps to appropriate pre-stored code segment to determine what action should be taken to handle each type of interrupt.**

Multiple Interrupts.

What happens when an interrupt interrupts an interrupt?

Disable Interrupts

- **new interrupts remain pending**
- **some interrupts are more important than others**

Interrupt Priorities.

Performing I/O

- **I/O is a major part of a computer's life. Managing I/O efficiently is a major OS responsibility.**
- **Each I/O device is controlled by one device controller: Device controller has its own processor, and executes *asynchronously* with CPU.**
- **Three major mechanisms to perform I/O:**
  - **program driven I/O**
  - **interrupt driven I/O**
  - **direct memory access (DMA)**

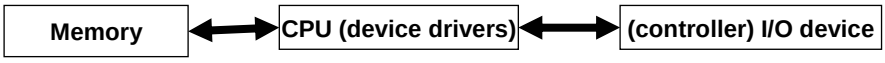
Example.

Programmed Input/Output (I/O)

When I/O must happen

- CPU tells I/O device controller what to do
- I/O device controller does it
- CPU must check to see if I/O device has finished
- If it has the CPU can go on

Problems caused when I/O device is slow.

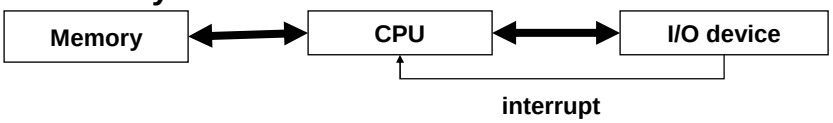


Interrupt Driven I/O.

When I/O is to be done

- CPU tells I/O device controller what to do (data+operation)
- I/O device does I/O
- When each I/O is complete device controller interrupts the CPU
  - » (CPU maintains status table and a link list of jobs per I/O device)
- CPU can perform data transfer (if necessary)

CPU is still responsible for transferring data from memory to the I/O device.

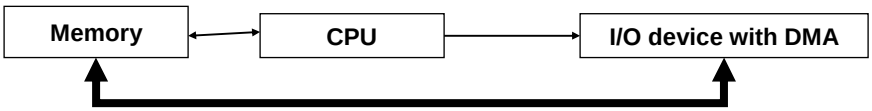


A 9600 baud terminal can transfer 1 character in 1 milliseconds=1000 microseconds
1 interrupt can be processed in 2 microseconds
A device with 1 4-10 microsecond transfer rate can not wait for interrupt even. What to do?

Direct Memory Access.

When I/O is to be done

- CPU tells I/O device controller what to do and where to put/get the information
- CPU continues as normal
- I/O device with the help of DMA unit puts/gets information straight into memory
- DMA interrupts CPU at the end of entire data transfer



Overview: Interrupt Handling and I/O