CS341: Operating System

Computer System Architetcure from OS Design Prospective

Lect03 : 5th Aug 2014

Dr. A. Sahu
Dept of Comp. Sc. & Engg.
Indian Institute of Technology Guwahati

1

Outline

- **OS in one slide**: Virtualized the Hardware and Manage
- Review of Computer Systems Architecture
- **OS in one Lecture:** Components of OS

2

Operating System in One Slide

- OS act an intermediary between : User and Hardware
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner
- A well-designed useful abstracted interfaces
  - Sockets, file systems, and address spaces into a single convenient package , **Probably best contributions of CS**
- Provides a **virtual execution environment** on top of hardware
  - That is more convenient than the raw hardware interface
  - More Simple, More reliable, More secure, More portable, More efficient............

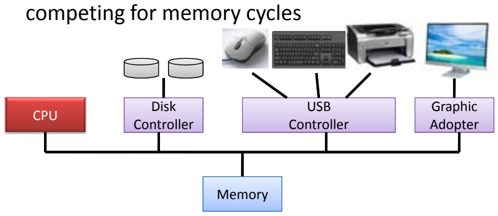
3

Review of Computer System Architecture from OS Prospects

4

Computer System Organization

- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



Every Motherboard Have

- Processor and Memory (Need to Add)
- Controller (Programmable & Now a day -Smarter)
  - DMA Controller (Ex IC 8237) {Simiar to Taxi wala}
  - Interrupt Controller (Ex IC 82C59) {PA to Director}
  - Timer (Example IC 82C54) {An alram }
  - USB Controller (Example IC 8251) $lsusb
  - PCI Bus Controller $lspci
- All these devices have : Address to control , control register and status register

CS321: Computer Peripheral and Interface
How to write monitor or driver program in Assembly Language using 8085 or in C using PIC Microcontroller

Computer-System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an interrupt

Common Functions of Interrupts

- Interrupt transfers control to
  - ISR generally, through the **interrupt vector**
  - **interrupt vector :** addresses of all service routines
- Interrupt architecture must save
  - Address of the interrupted instruction
- A **trap** or **exception**
  - Software-generated interrupt caused either by an error or a user request
  - It do not go to Interrupt controller, but it handle on its own.. (Similar to: if Director get a problem he don't ask to PA but solve himself)
- An operating system is **interrupt driven**

Interrupt Handling

- OS preserves the state of the CPU
  - By storing registers and the program counter
- Determines which type of interrupt has occurred:
  - **polling**
  - **vectored** interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt

I/O Structure

- Blocking I/O methods
- After I/O starts, control returns to user program only upon I/O completion
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing

I/O Structure

- Non-Blocking I/O methods
- After I/O starts, control returns to user program without waiting for I/O completion
  - **System call** – request to the OS to allow user to wait for I/O completion
  - **Device-status table** contains entry for each I/O device indicating its type, address, and state
  - OS indexes into I/O device table to determine device status and to modify table entry to include interrupt

Storage Definitions and Notation Review

- Bit : Basic unit of storage
- Byte 8 bits
  - Smallest convenient chunk of storage
- Notation
  - **kilobyte**, or **KB**, is 1,024 bytes $\approx 10^3$
  - **megabyte**, or **MB**, is $1,024^2$ bytes $\approx 10^6$
  - **gigabyte**, or **GB**, is $1,024^3$ bytes $\approx 10^9$
  - **terabyte**, or **TB**, is $1,024^4$ bytes $\approx 10^{12}$
  - **petabyte**, or **PB**, is $1,024^5$ bytes $\approx 10^{15}$

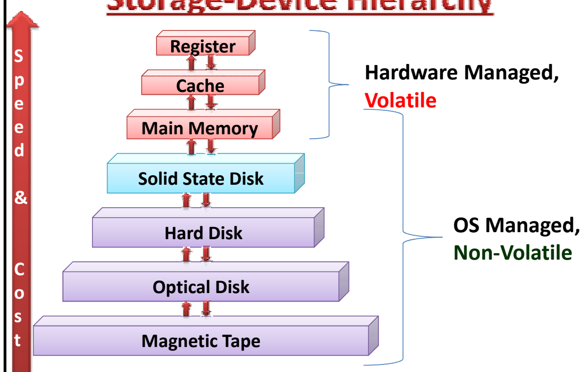
Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - **Random access,** Typically **volatile**
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- Hard disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer
- **Solid-state disks** – faster than hard disks, nonvolatile
  - Various technologies, Becoming more popular

Storage Hierarchy

- Storage systems organized in hierarchy
  - Speed, Cost, Volatility
- **Caching** – copying information into faster storage system; main memory can be viewed as a cache for secondary storage
  - Locality of Reference
  - Locality Principle (Spatial and Temporal Locality)
- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

Storage-Device Hierarchy



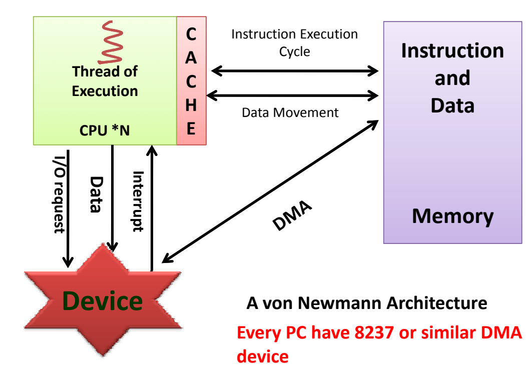
Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied
  - From slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy

Direct Memory Access Structure

- **DMA**
- Used for high-speed I/O devices able to transmit information at close to memory speeds
- ***Device controller*** transfers blocks of data from buffer storage directly to main memory without CPU intervention
  - ***Taxi Wala to handle all Pickup and Drop of Conference***
- Only one interrupt is generated per block, rather than the one interrupt per byte

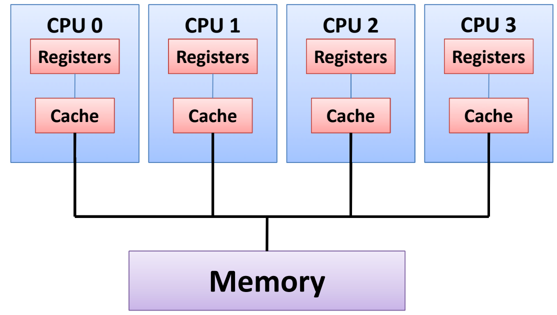
I/O transfer in a Modern Computer



Computer-System Architecture

- Most systems use a general-purpose processor
  - Most systems have special-purpose processors as well
- **Multiprocessors** growing in use and importance
- Also known as **parallel systems**, **tightly-coupled systems** Advantages include:
  - **Increased throughput, Economy of scale**
  - **Increased reliability** – graceful degradation or fault tolerance
- **Asymmetric Multiprocessing** – each processor is assigned a specific task.
- **Symmetric Multiprocessing** – each processor performs all tasks

Symmetric Multiprocessing Architecture



Asymmetric Architecture