OPERATING SYSTEM

Computer System Organization

Dr Rahul Nagpal
Computer Science

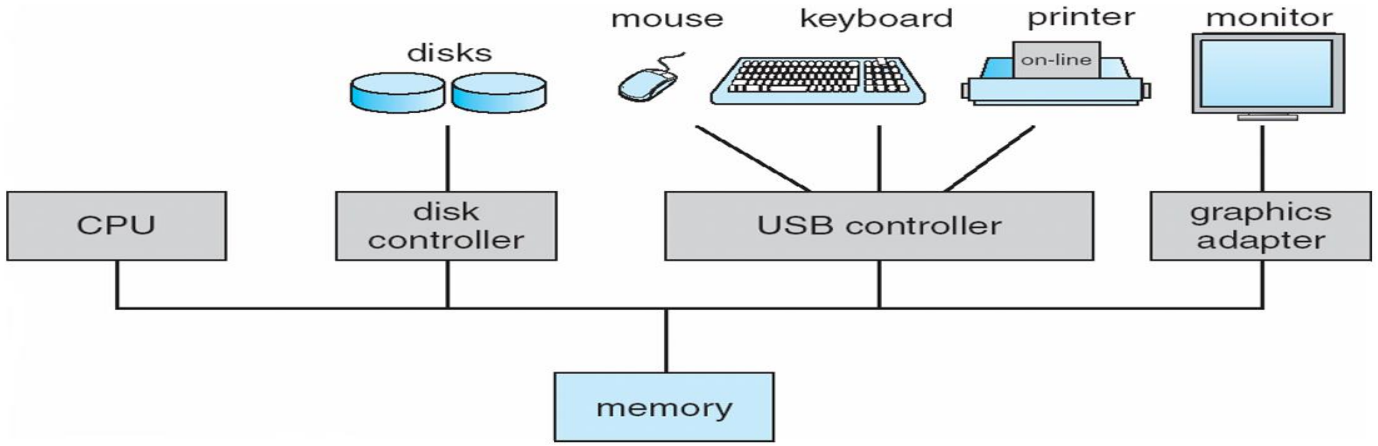
OPERATING SYSTEM

Computer System Organization

Dr. Rahul Nagpal
Computer Science

OPERATING SYSTEMS
Computer System Organization

- Computer-system Organization
  - 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



OPERATING SYSTEMS
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 interrupts CPU on completion

Computer Startup

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

OPERATING SYSTEMS

Common Functions of Interrupts

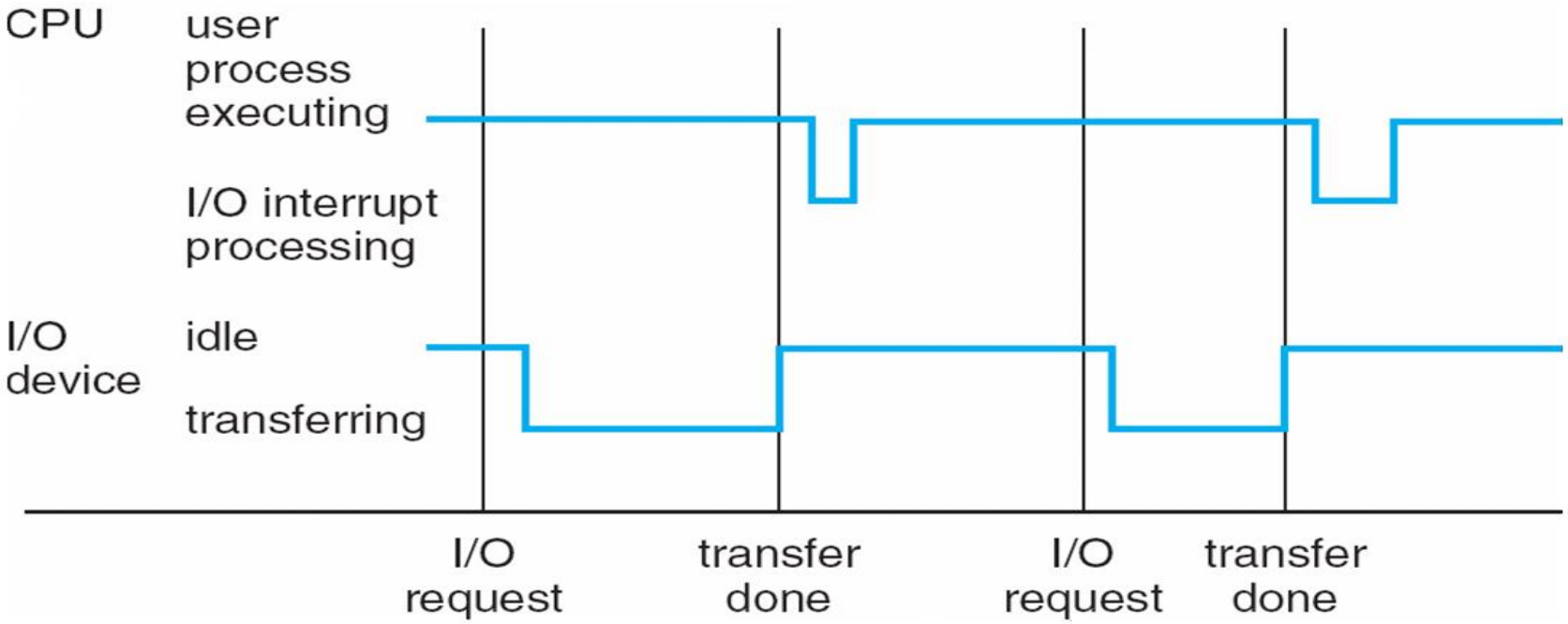
- An operating system is **interrupt driven**
- Interrupt transfers control to the interrupt service routine
- **interrupt vector** contains the addresses of all the service routines for transfer
- Interrupt architecture must save the address of the interrupted instruction
- A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request

OPERATING SYSTEMS

Interrupt Handling

- The operating system 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

OPERATING SYSTEMS
Interrupt Timeline



OPERATING SYSTEMS
Storage Structure

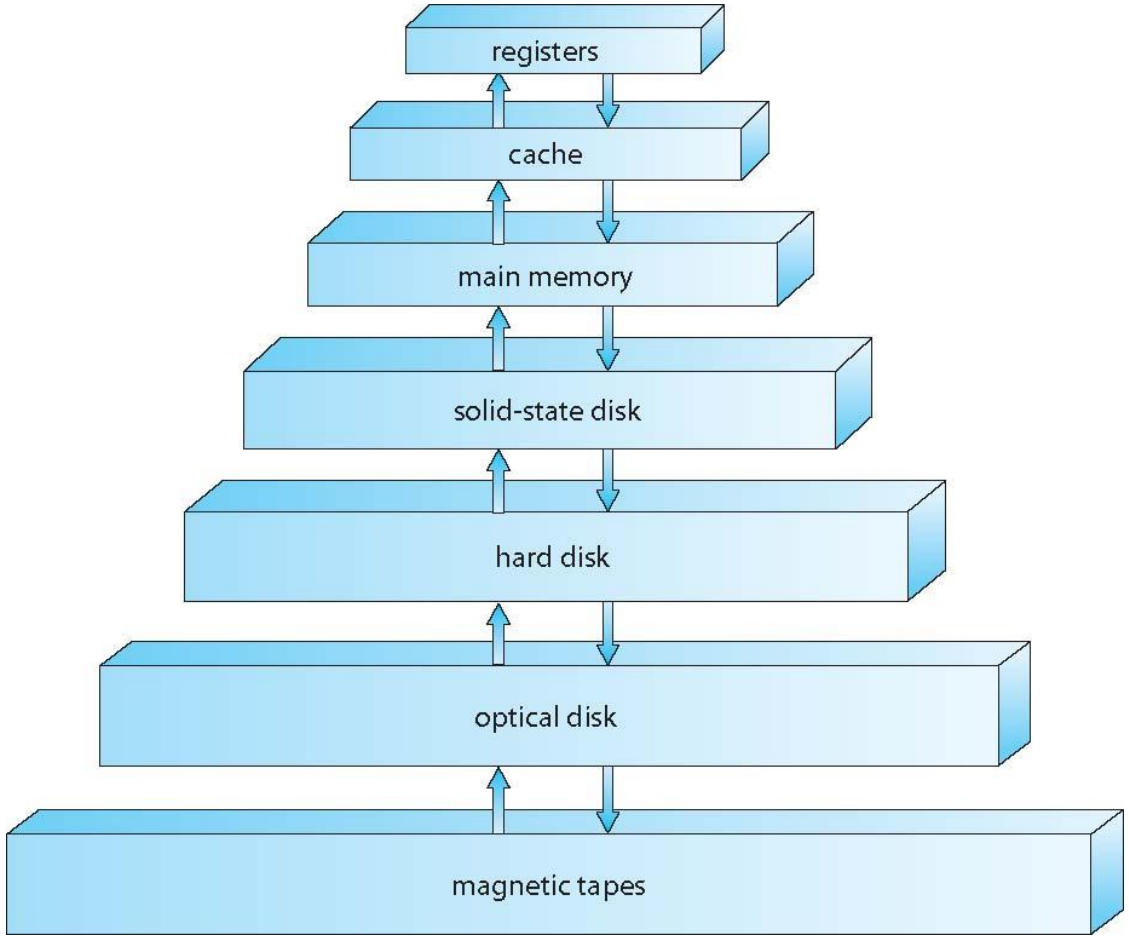
- Main memory – only large storage media directly access by CPU
  - **Random access and typically volatile**
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- Hard disks – rigid metal or glass platters covered with magnetic recording material
  - Disk 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
- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

OPERATING SYSTEMS

Storage-Device Hierarchy



OPERATING SYSTEMS
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

OPERATING SYSTEMS
I/O Structure

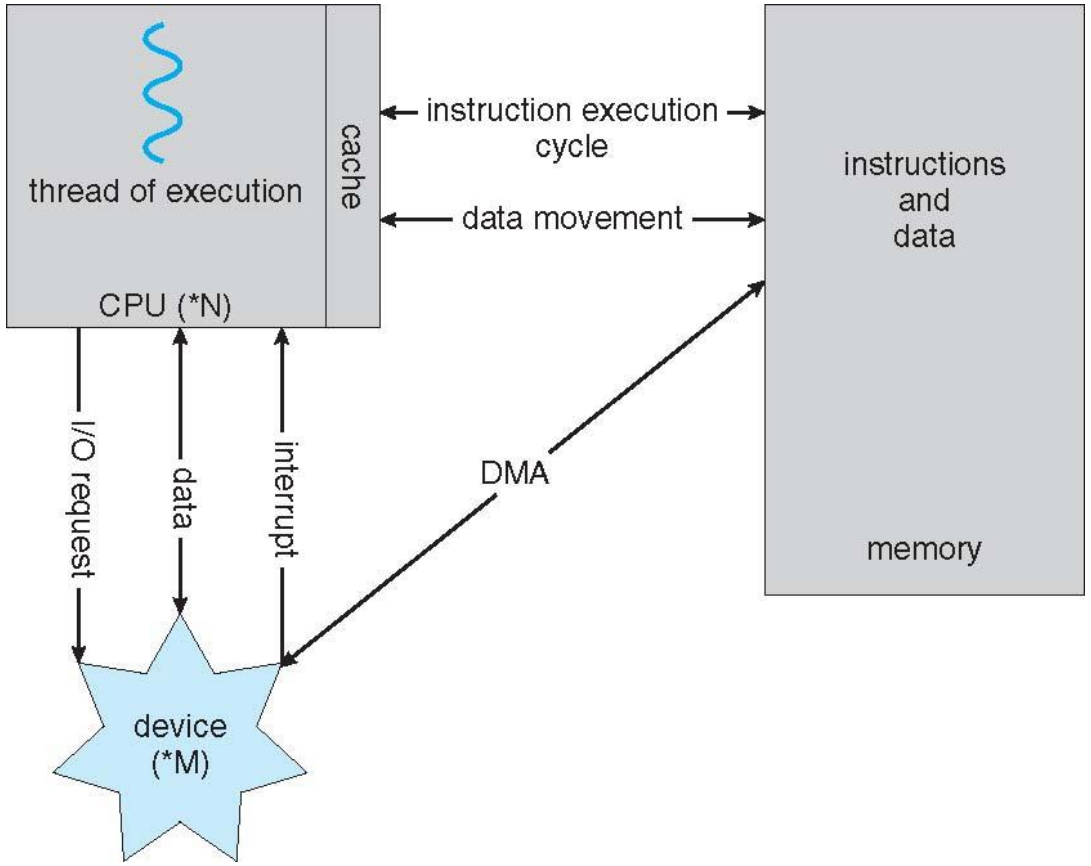
- After I/O starts, control returns to user program only upon I/O completion
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing
- After I/O starts, control returns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

Direct Memory Access Structure

- 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
- Only one interrupt is generated per block, rather than the one interrupt per byte

OPERATING SYSTEMS

How a Modern Computer Works



A von Neumann architecture

THANK YOU

Dr Rahul Nagpal

Computer Science

rahulnagpal@pes.edu