Operating systems

What is an operating system
- Acts as an intermediary between a user and hardware
- Operating systems goals
  - Execute user programs, and solve user problems
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Computer systems structure
- Hardware
  - Basic computing resources
    - Cpu, memory, i/o device
- Operating system
  - Controls and coordinates use of hardware among various applications and users
- Application programs
  - Define ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
- Users
  - People, machines, other computers

What operating systems do
- Users want convenience, ease of use, and good performance
  - Dont care about resource utilization
- Shared computers like mainframe or minicomputer must keep all users happy
- Users dedicate systems such as workstations have dedicated resources but frequently use shared resources from servers
- Handheld computers are resource poor, optimized for usability and battery life
- Some computers have little or no user interface, such as embedded computers in devices and automobiles

Operating system definition
- OS is a resource allocator
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a control program
  - Controls execution of programs to prevent errors and improper use of the computer
- No universal accepted definition
- One program running at all times on the computer is the kernel
- Everything else is either
  - A system program (ships with the operating system)
  - An application program

Computer startup
- Bootstrap program
  - Loaded at power-up or reboot
    - Typically stored in ROM or EPROM, generally known as firmware
    - Initializes all aspects of system
    - Loads operating system kernel and starts execution

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

Memory - connected by bus
- CPU
- Disk controller
- USB controller
- Graphics adapter

Computer system operations
- i/o device 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
- Interrupts transfer control to the interrupt service routine generally, through the interrupt vector
- Interrupt vector contains the addresses of all the service routines
- 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
- An operating system is interrupt driven

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
        - Goes by each device and queries if it started interrupt
    - Vectored interrupt system
        - You know which addresses to go to when interrupt happens
- Separate segments of code determine what action should be take for each type of interrupt

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

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 non volatile 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 non volatile

Structure hierarchy
- Storage systems organized in hierarchy
    - Speed
    - Cost
    - Volatility
- Cashing
    - 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

Storage device hierarchy
- Registers - fastest - most expensive
    - Cache
        - Main memory
            - Solid-state disk
                - Hard disk
                    - Optical disk
                        - Magnetic tapes - slowest - cheapest

Direct memory access structure
- Used for high speed i/o device 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

Computer system architecture

- Most systems use a single general purpose processor
    - Most systems have special purpose processors
- Multiprocessors systems growing in use and importance
    - Also known as parallel systems, tightly coupled systems
    - Advantage include
        - Increased throughput
        - Economy of scale
        - Increased reliability
            - Graceful degradation or fault tolerance
    - Two types
        - Asymmetric multiprocessing
            - Each processor is assigns a special talks
        - Symmetric multiprocessor
            - Each processor is a clone

Clustered systems

- Like multiprocessor systems, but multiple systems working together
    - Usually sharing storage via a storage-area network (SAN)

Multiprogramming (batch system)

- Needed for efficiency

Timesharing (multitasking)

- Is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating interactive computing