Chapter 1 Summary Operating System Concepts – 9th Edition

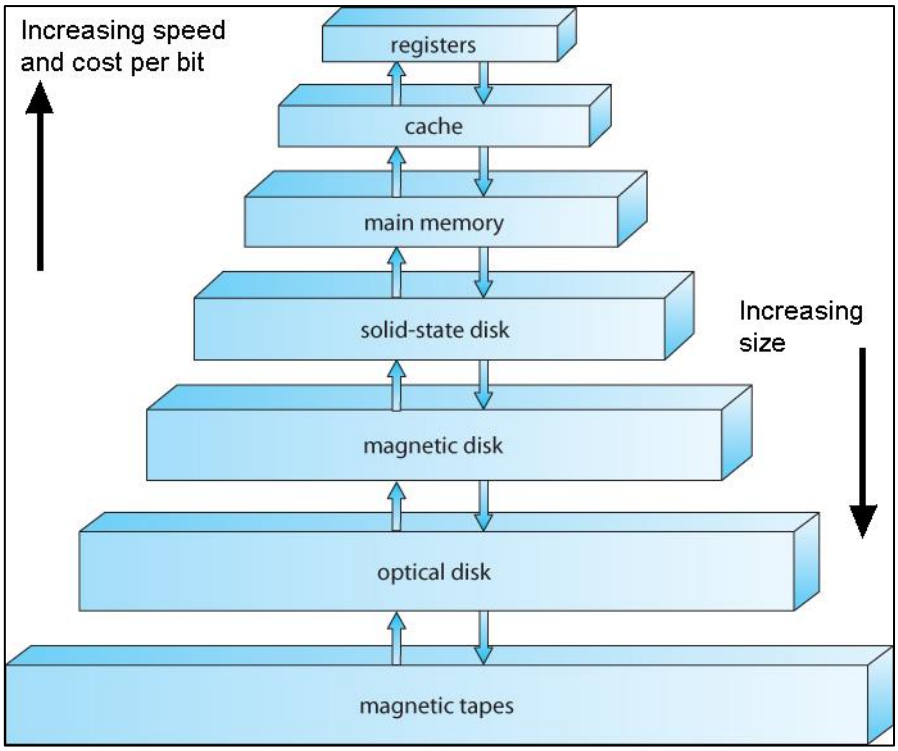
Operating system (جامعة القاهرة)

Chapter 1 summary

- Operating system: a program that acts as an intermediary between a **user of a computer** and the **computer 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.
- Computer system can be divided into **four** components:
    - **Hardware** – provides basic computing resources
        - CPU, memory, I/O devices.
    - **Operating system**
        - Controls and coordinates use of hardware among various applications and users.
    - **Application programs** – define the 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 system does?
    - User view:
        - Convenience, Ease of use.
        - Do not care about **resource utilization**.
        - But shared computer such as **mainframe** or **minicomputer** must keep all users happy.
        - Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers** (internet).
        - Handheld computers (mobile phones) are resource poor, optimized for usability and battery life.
        - Some computers have little or no user interface, such as embedded computers in devices (cars) and automobiles.
    - System view:
        - 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 universally accepted definition of what part of the operating system is.

- **Kernel**: the one program running always on the computer.
- **Bootstrap**: the program is loaded when a computer is powered up or rebooted.
    - Typically stored in **ROM** or **EPROM**, generally known as **firmware** (something between hardware and software)
    - Initializes all aspects of system.
    - Loads **operating system kernel** and starts execution.
- Computer-System Organization:
    - One or more CPUs, device controllers connect through common bus (wire) providing access to shared memory.
    - CPU and device controllers can execute in **parallel**, competing for memory cycles
    - CPU and I/O devices can execute **concurrently** (at the same time).
    - Each device controller oversees 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 the appropriate interrupt service routine, through the **interrupt vector**, which 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**
        - **Vectored** interrupt system

- Storage structure:
  - **Main memory** – only large storage media that the CPU can access directly
    - **Random access**
    - **Volatile**
  - **Secondary storage** – extension of main memory that provides large **nonvolatile** storage capacity
    - **Magnetic 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 magnetic disks
- Storage hierarchy:
  - Storage systems organized in hierarchy:
    - Speed
    - Cost
    - Volatility
    - Storage
  - **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 interface between controller and kernel



- 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.
- Computer-System architecture:
    - Most systems use a single general-purpose processor.
    - Multiprocessors systems growing in use and importance.
        - Also known as **parallel** or **multicore** systems
        - Advantages include:
            - Increased throughput (number of processes per unit of time)
            - Economy of scale
            - Increased reliability – graceful degradation or fault tolerance
        - Two types:
            - **Asymmetric** Multiprocessing (each processor is assigned a task)
            - **Symmetric** Multiprocessing (each processor performs all tasks)
- **Multiprogramming** needed for efficiency:
    - Multiprogramming organizes jobs, so CPU always has one to execute.
    - A subset of total jobs in the system is kept in memory.
    - A jobs set is selected from a pool via **job scheduling**.
    - When a job must wait (for I/O for example), OS switches to another job.

- **Timesharing (multitasking)** is a logical extension to multiprogramming:
    - The CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing.
    - **Response time** should be short (typically < 1 second).
    - Each user has at least one program executing in memory -> **process**.
    - If several jobs ready to run at the same time-> **CPU scheduling**.
    - If processes do not fit in memory, **swapping** moves them in and out to run.
    - **Virtual memory** allows execution of processes not completely in memory.
- Operating-System operations:
    - Interrupt driven by hardware.
    - Software error or request creates **exception** or **trap**.
- Dual-mode operation allows OS to protect itself and other system components.
    - **User** mode and **kernel** mode.
    - **Mode bit** provided by hardware (0,1).
        - Provides ability to distinguish when system is running user code or kernel code.
        - Some instructions designated as **privileged**, only executable in kernel mode.
        - System call changes mode to kernel, return from call resets it to user.
    - Increasingly CPUs support multi-mode operations
        - i.e. **virtual machine manager (VMM)** mode for guest **VMs**
- **Timer** to prevent infinite loop / process hogging resources
    - Set interrupt after specific period (fixed or variable).
    - Operating system initializes and decrements counter.
    - When counter reaches zero generate an interrupt.
    - Set up before scheduling process to regain control or terminate program that exceeds agreed time.
- A **process** is a program in execution
    - It is a unit of work within the system
    - Program is a **passive** entity, process is an **active** entity
- Process needs resources to accomplish its task
    - CPU, memory, I/O, files
    - Initialization data
- Process termination requires reclaim of any reusable resources.
- Single-threaded process has **one program counter** specifying location of next instruction to execute
    - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread.

- Typically, a system has many processes running **concurrently** on one or more CPUs
    - Some user processes and others system processes.
    - Concurrency by multiplexing the CPUs among the processes / threads.
- Memory management allows keeping several programs in memory.
- Memory management activities:
    - Keeping track of which parts of memory are currently being used and by whom
    - Deciding which processes and data to move into and out of memory.
    - Allocating and deallocating memory space as needed.
- File: Abstracts physical properties to logical storage unit.
- Usually disks used to store data that does not fit in main memory or data that must be kept for a "long" period of time.
- **Tertiary storage** includes optical storage, magnetic tape.
- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy.
- Multiprocessor environment must provide **cache coherency** in hardware such that all CPUs have the most recent value in their cache.
- I/O Subsystem:
    - One purpose of OS is to hide individuality of hardware devices from the user.
    - I/O subsystem responsible for:
        - Memory management of I/O including:
            - **buffering** - storing data temporarily while it is being transferred.
            - **caching** - storing parts of data in faster storage for performance.
            - **spooling** - overlapping of output of one job with input of other jobs.
        - General device-driver interface.
        - Drivers for specific hardware devices.
- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS.
- **Security** – defense of the system against internal and external attacks.
- Systems generally first distinguish among users, to determine who can do what.
    - User identities (user IDs, security IDs) include name and associated number, one per user.
    - User ID then associated with all files, processes of that user to determine access control.
    - Group identifier (group ID) allows set of users to be defined and controls managed, then also associated with each process, file.
    - **Privilege escalation** allows user to change to effective ID with more rights.

- **Portals** provide web access to internal systems.
- **Network computers (thin clients)** are like Web terminals.
- Mobile computers interconnect via **wireless networks**.
- Networking becoming ubiquitous - even home systems use **firewalls** to protect home computers from Internet attacks.
- Handheld smart phones and tablet computers have more OS features.
- Network is a communications path, **TCP/IP** most common
    - Local Area Network (LAN)
    - Wide Area Network (WAN)
    - Metropolitan Area Network (MAN)
    - Personal Area Network (PAN)
- **Network Operating System** provides features (as file sharing) between systems across network
    - Communication scheme allows systems to exchange messages.
    - Impression of a single system.
- Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system** provides an interface to client to request services.
    - **File-server system** provides interface for clients to store and retrieve files.
- Peer to peer (P2P) does not distinguish clients and servers.
- Broadcast request for service and respond to requests for service via **discovery protocol**.
- **Emulation** used when source CPU type different from target CPU type.
- **Virtualization** – OS natively compiled for CPU, running guest OSs also natively compiled.
- **Cloud computing** – Logical extension of virtualization as based on virtualization.
    - Internet connectivity requires security like firewalls.
    - Load balancers spread traffic across multiple applications.
- **Public cloud** – available via Internet to anyone willing to pay.
- **Private cloud** – run by a company for the company's own use.
- **Hybrid cloud** – includes both public and private cloud components.
- Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e. word processor)
- Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e. database server)
- Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e. storage available for backup)