جامعة ألأنبار

كلية علوم الحاسوب وتكنولوجيا المعلومات

قسم أنظمة شبكات الحاسوب

المرحلة الرابعه

Operating System

التدريسي: أ.م.د عمر منذر حسين

Introduction

What is an Operating System?

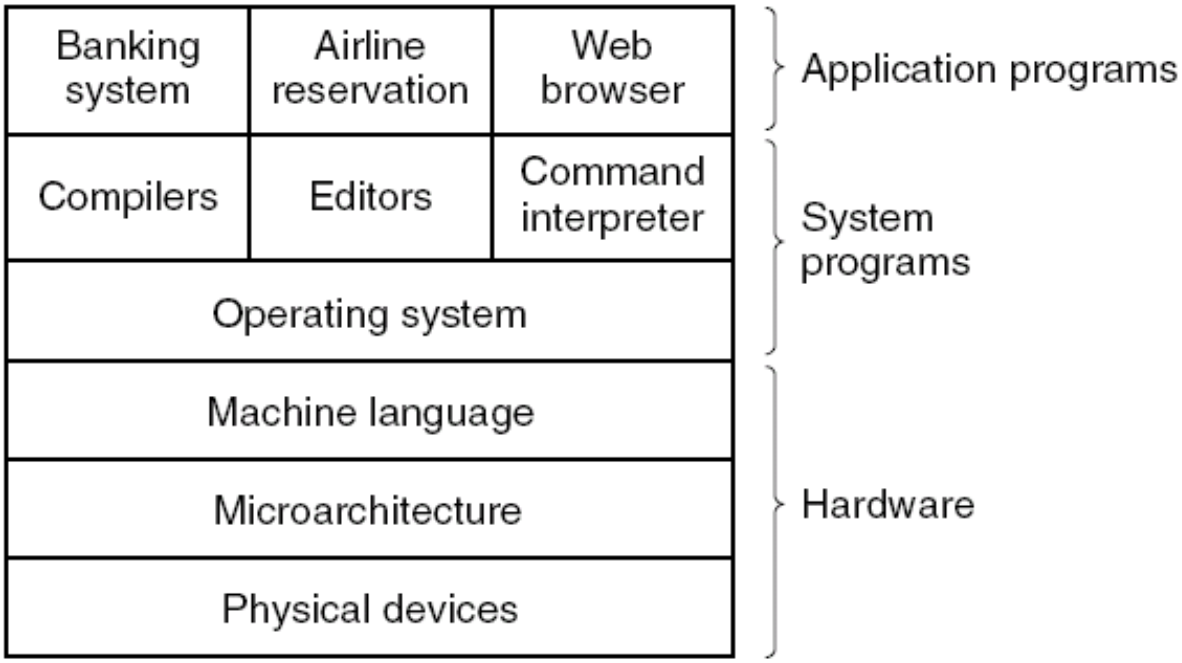
- An **operating system** is a program that manages a computer's hardware.
- It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.
- As a manager, the operating system has two basic functions:

  - OS oversees all hardware resources and allocates them to user and applications as needed.
  - Performs many low-level tasks on behalf of users and application programs.

Types of OS

1. **Mainframe Operating Systems:** The operating systems for mainframes are heavily oriented toward processing many jobs at once, most of which need prodigious amounts of I/O. (e.g., OS/390).
2. **Server Operating Systems:** They run on servers, and allow the users to share H/W and S/W resources. (e.g., Windows server).
3. **Multiprocessor Operating Systems**
4. **Personal Computer Operating Systems**
5. **Handheld Computer Operating Systems**
6. **Embedded Operating Systems: r**un on the computers that control devices that are not generally thought of as computers and which do not accept user-installed software. Typical examples are microwave ovens and TV sets.
7. **Sensor Node Operating Systems.**
8. **Real-Time Operating Systems.**
9. **Smart Card Operating Systems:** The smallest operating systems run on smart cards, which are credit card sized devices containing a CPU chip

The Modern Computer System



A computer system consists of hardware, system programs, and application programs.

Basic Elements of Computer System

- Four main structural elements:
    1. Processor: Controls the operation of the computer and performs its data processing functions. Often referred as **CPU**.
    2. Main memory: Stores data and programs.
        - Volatile: when the computer shut down, contents are lost.
        - Real memory or primary memory.
    3. I/O modules: Move data between computer and its external environment (e.g., disks).
    4. System bus: Provides for communication among processors, main memory and I/O modules.
    5. Memory Address Register (MAR): specifies the address in memory for the next read or write.

6. Memory Buffer Register (MBR): Contain the data to be written into or read from memory.
7. I/O Address Register (I/O AR): Specifies particular I/O device.
8. I/O Buffer Register (I/O BR): used for exchange of data between I/O module and the processor.

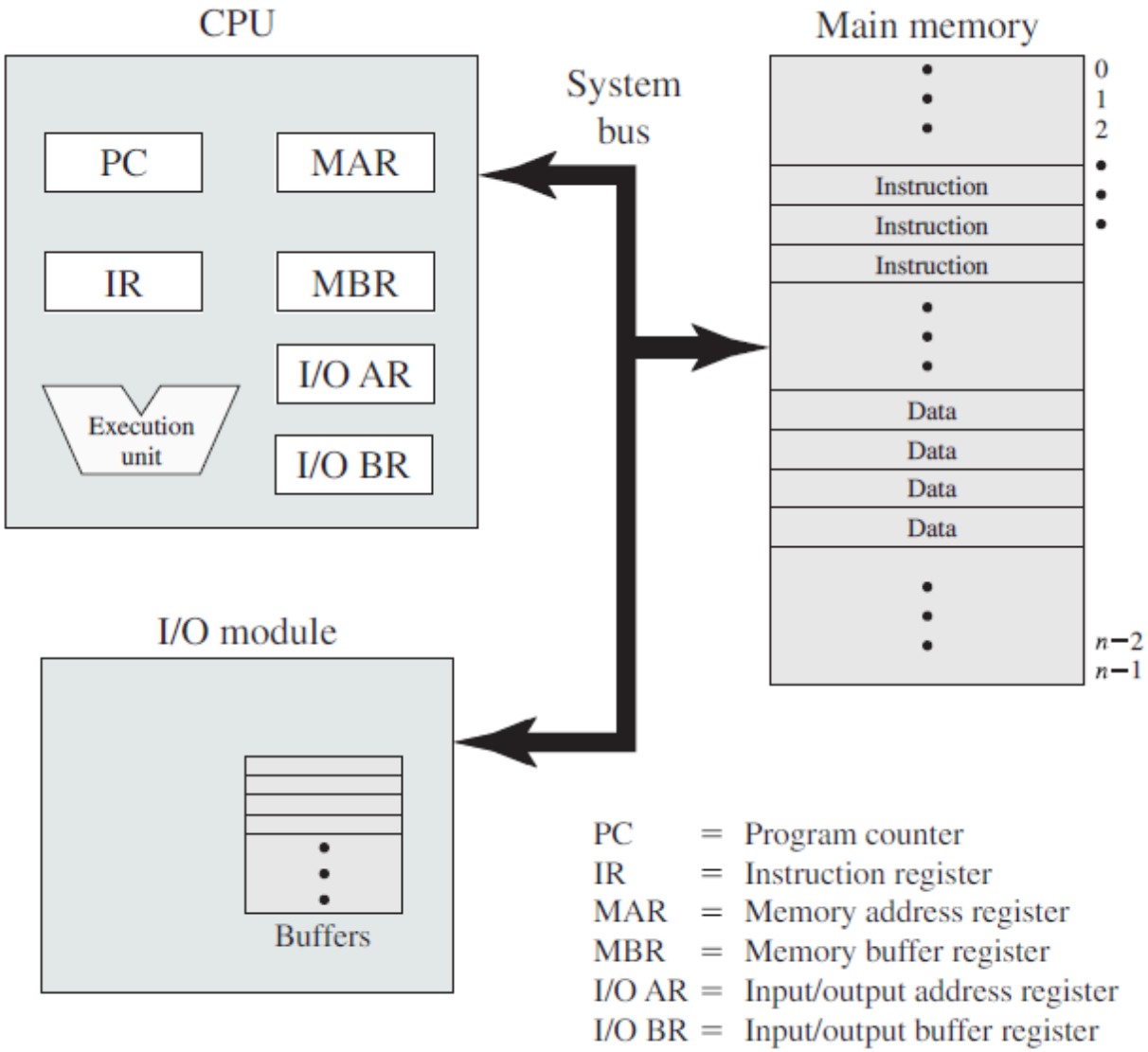


Figure 1.1 Computer Components: Top-Level View

Computer System Components

1. Hardware – provides basic computing resources (CPU, memory, I/O devices).

2. Operating system – controls and coordinates the use of the hardware among the various application programs for the various users.
3. Applications programs – define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
4. Users (people, machines, other computers).
5. I/O devices and the CPU can execute concurrently
6. A number of device controllers connected through a common bus that provides access to shared memory
7. Each device controller is in charge of a particular device type
8. Each device controller has a local buffer
9. CPU moves data from/to main memory to/from local buffers
10. Device controller communicates with CPU by using an interrupt.

Computer-System Operation

- For a computer to start running—for instance, when it is powered up or rebooted—it needs to have an initial program to run.
- This initial program, or **bootstrap program,** is stored within the computer hardware in read-only memory (ROM) or electrically erasable programmable read-only memory (EEPROM), known by the general term firmware.
- It initializes all aspects of the system, from CPU registers to device controllers to memory contents.
- The bootstrap program must locate the operating-system kernel and load it into memory.
- Once the kernel is loaded and executing, it can start providing services to the system and its users.
- Once this phase is complete, the system is fully booted, and the system waits for some event to occur.

Interrupts

- If there are no processes to execute, no I/O devices to service, and no users to whom to respond, an operating system will sit quietly, waiting for something to happen.
- Events are almost always signaled by the occurrence of an interrupt or a trap.
- A trap (or an exception) is a software-generated interrupt caused either by an error (for example, division by zero or invalid memory access) or by a specific request from a user program that an operating-system service be performed.
- The occurrence of an event is usually signaled by an **interrupt** from either the hardware or the software.
- Hardware may trigger an interrupt at any time by sending a signal to the CPU, usually by way of the system bus.
- Software may trigger an interrupt by executing a special operation called a system call.
- When the CPU is interrupted, it stops what it is doing and immediately transfers execution to a fixed location. The fixed location (**interrupt vector)** usually contains the starting address where the **service routine** for the interrupt is located.
- The interrupt service routine executes; on completion, the CPU resumes the interrupted computation.
- Interrupt architecture must save the address of the interrupted Instruction.
- **Thus, operating system is interrupt driven!**

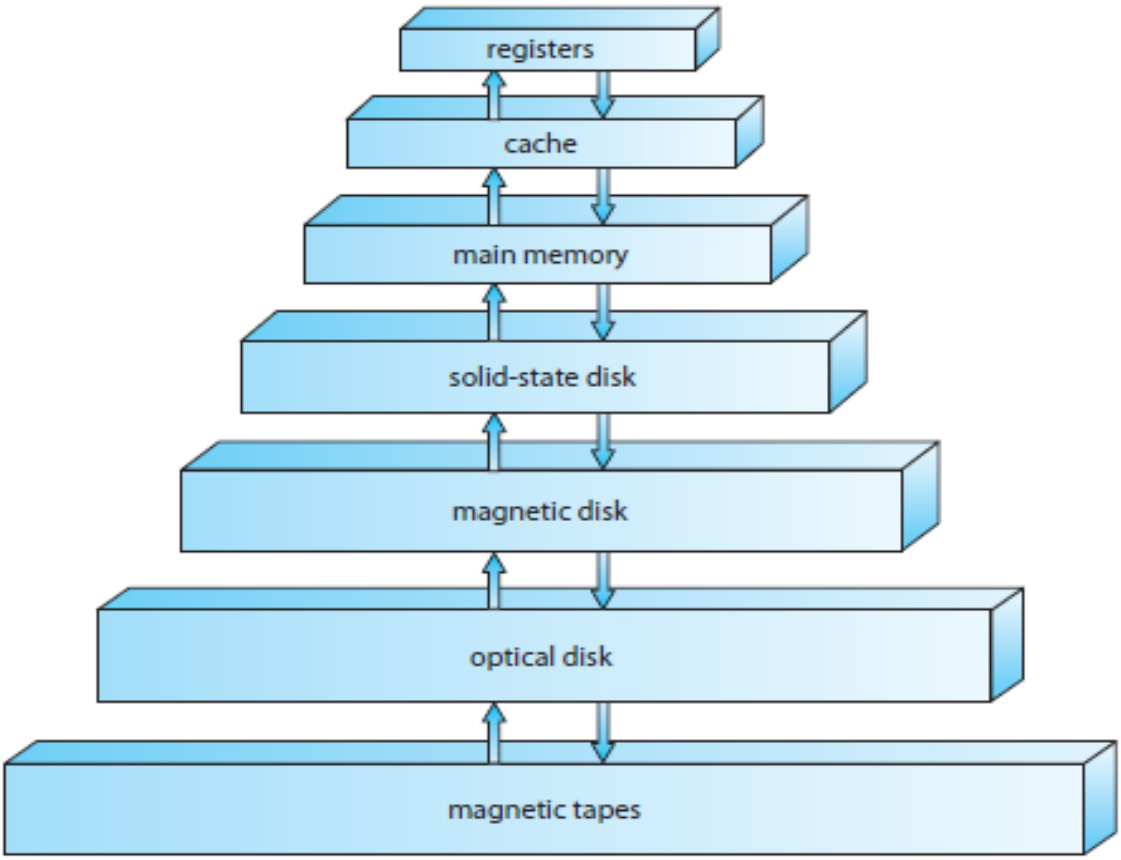
Storage Structure

- **Main memory** – only large storage media that the CPU can access directly. It is **Random access** and **volatile!**
- **Secondary storage** – extension of main memory that provides large **nonvolatile** storage capacity.
- 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.

Storage Hierarchy

- The main differences among the various storage systems lie in **speed**, **cost**, **size**, and **volatility**.
- The wide variety of storage systems can be organized in a hierarchy (Figure) according to speed and cost.
- The higher levels are expensive, but they are fast.
- As we move down the hierarchy, the cost per bit generally decreases, whereas the access time generally increases.



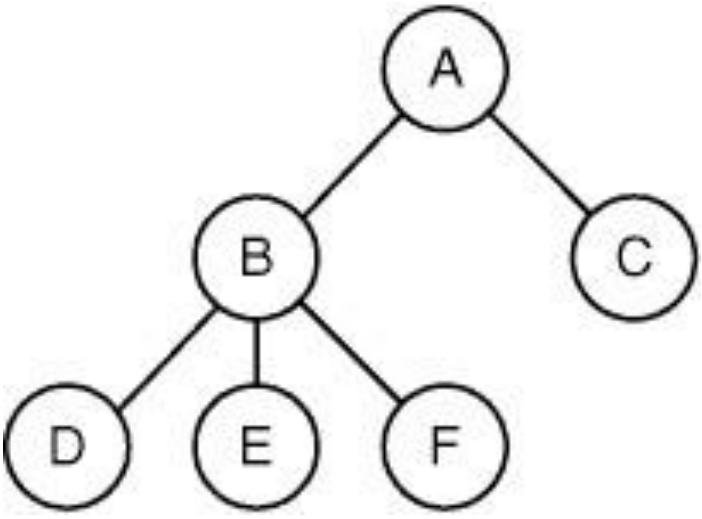
Caching

- Caching is an important principle of computer systems.
- 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.
- Because caches have limited size, cache management is an important design problem.

OS Concepts- Processes

- **Process** is a running program, for each process there is **address space**, which is a list of memory locations. Also, associated with each process is a set of **resources**.
- A process is fundamentally a container that holds all the information needed to run a program.
- Example: video editing program to convert a one-hour video to a certain format
- Gone off to surf the Web. Meanwhile, a
- Background process that wakes up periodically to check for incoming e-mail may have started running.
- Three active processes: the video editor, the Web browser, and the e-mail receiver.
- Periodically, the operating system decides to stop running one process and start running another; for example, because the first one has used up more than its share of CPU time in the past second.
- When a process is suspended temporarily, it must later be restarted in exactly the same state it had when it was stopped.
- This means that all information about the process must be explicitly saved somewhere during the suspension.

- For example, the process may have several files open for reading at once.
- Associated with each of these files is a pointer giving the current position (i.e., the number of the byte or record to be read next). When a process is temporarily suspended, all these pointers must be saved so that a read call executed after the process is restarted will read the proper data.
- In many operating systems, all the information about each process, other than the contents of its own address space, is stored in an operating system table called the **process table**
- For example, the process may have several files open for reading at once.
- Associated with each of these files is a pointer giving the current position (i.e., the number of the byte or record to be read next). When a process is temporarily suspended, all these pointers must be saved so that a read call executed after the process is restarted will read the proper data.
- In many operating systems, all the information about each process, other than the contents of its own address space, is stored in an operating system table called the **process table.**

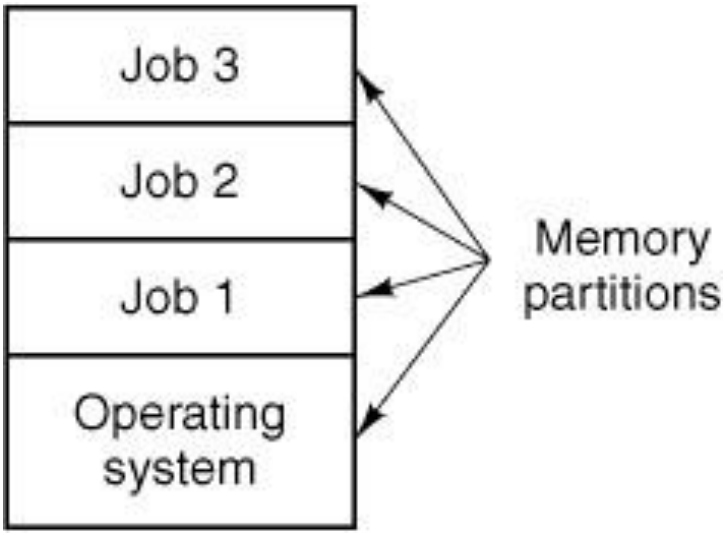


- A process tree. Process A created two child processes, B and C. Process B created three child processes, D, E, and F.

Address Space

- ❖ Each computer has RAM to store executing programs.
- ❖ In a very simple OS, only one program at a time is in memory.

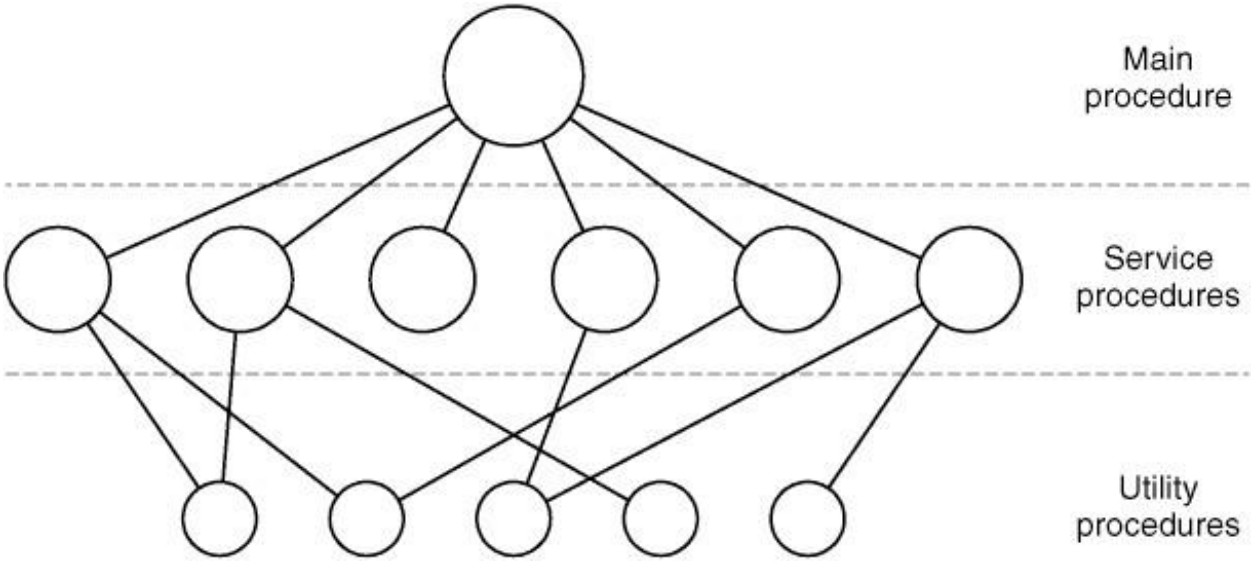
- More sophisticated OS allow multiple programs to be in RAM.
- A multiprogramming system with three jobs in memory.



Operating System Structure

- Six different structures of OS that have been tried.
- **Monolithic Systems, Layered Systems, Microkernels, Client-Server Model, Virtual Machines, Exokernels.**
- **Multiprogramming** needed for efficiency, Single user cannot keep CPU and I/O devices busy at all times.
- Multiprogramming organizes jobs (code and data) so CPU always has one to execute.
- One job selected and run via **job scheduling!**
- When it has to wait (for I/O for example), OS switches to another job.
- **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.

Layered Systems



A simple structuring model