Chapter 1

- **The four components of a computer system:**

اربع اجزء رئسية بيتكون منهم الكمبيوتر

➢ Hardware:

- provides the basic resources for the system

CPU time ، Memory، I/O devices

القطع الماديه

➢Operating system:

دا لي كذا تعريف انا حبيت ابسطهم وحطهم مرة وحدة

- A software program that manages the hardware of a computer and coordinates its use

بيدير وينسق استخدام الهاردوير

- It acts as an intermediary between the computer user and the hardware

يعني بيكون الوسيط بين المستخدم والهاردوير لاني لو مستخدم عادي مش اقدر اتعامل مع الهاردوير

- It provides basis for application programs to run

بيكون الاساس لتشغيل اي برنامج

➢ Application programs:

- Define the ways in which the resources are used

بتخلي استخدام الكمبيوتر اسهل ونعرف نستخدم الموارد الي فيه

➢ User:

- You ,machines ,other computer

- **Resources of a computer system:**
  - ➢ CPU time
  - ➢ Memory space
  - ➢ File-storage space
  - ➢ I/O devices
- **Operating system:**
  - ➢ Can be viewed as a resource locator as it manages the resources and decides how to allocate them to programs

يدير الموارد ويقرر كيفية تخصيصها للبرامج

  - ➢ Can be viewed as a control program as it controls I/O devices and manages the execution of user programs

ممكن نعتبره برنامج بيدير ويتحكم في تنفيذ برامج المستخدم

  - ➢ Consists of:

مكونات نظام التشغيل

    - **Kernel:**
- Running all time on the computer

دا بيكون شغال طول الوقت من بعد ما تدوس علي زورار البور ودا زي قلب الانسان شغال باستمرار

    - **System programs:**
- Associated with the OS but not necessarily part of the kernel
    - **application programs:**
- **Modern general purpose computer system consists of:**
  - ➢ One or more CPUs:
    - Can execute in parallel to device controllers

➢ Device controllers (connected through a common bus):

▪ Each device controller is in charge of a specific type of device ➢

A shared memory:

▪ A memory controller syncs access to the memory ➢

System bus:

▪ a common bus bus that provides access to a shared memory

- **Bootstrap program:**

➢ A simple program runs when the computer is powered up or rebooted

بيشتغل عند تشغيل الكمبيوتر ومعاه حاجات تاني بشتغل

➢ Stored in the ROM or EEPROM

➢ Known by the term Firmware

➢ Initializes all aspects of the system

➢ loads the kernel and starts its execution

يعني اول ما تدوس علي البور بيبدا الكمبيوتر باشغيل شوية برامج وبعدين بيشغل النواة

- **Interrupts:**

➢ It signals the occurrence of an event from the hardware or software

دا لما بدوس مثلا علي ايقونة برنامج فبيحصل.

➢ Each computer architecture has a set of interrupts

interrupts

أي معماريه للكمبيوتر لازم يكون فيها

➢ Each interrupt has an interrupt service routine (interrupt handler) which executes when the interrupt occurs

- ➢ When an interrupt occurs the CPU stops what it was doing and executes the appropriate interrupt service routine
- ➢ After completing the execution of the interrupt service routine the CPU resumes the interrupted computation
- ➢ Interrupt vector:
  - ▪ an array for holding the addresses of the interrupt service routines

يعني لما بنعمل برنامج بالجافا ونقسم علي صفر اكيد البرنامج ايقولي الاكسبشن انه مينفعش اقسم علي صفر ممكن نقول عليه رسائل الخطا

  - ▪ Stored in low memory (the first locations in memory)

ودي بتبقا مخزنة

  - ➢ Hardware generated interrupt:
  - ▪ occurs by sending a signal to the CPU by the way of the system bus

system bus ← لما الهاردوير يبعت اشار لل CPU

- ➢ Software generated interrupt:
  - ▪ Called: a trap or exception
  - ▪ occurs by executing a special operation called a system call
  - ▪ caused by an error or a user request

- **Storage structure:**
  - ➢ CPU loads instructions from memory
  - ➢ All forms of memory provides an array of bytes each one has a unique address
  - ➢ **Main memory (RAM / DRAM):**
    - ▪ interaction between it and the CPU is achieved through a sequence of instructions:

- load:
    - moves a byte or a word from main memory to a CPU register
- store:
    - moves the contents of a CPU register to main memory
    - Too small to store all programs and data
    - volatile
    - can be viewed as a cache for secondary storage

➢ **Other forms (ROM / EEPROM):**

- Contains mostly static programs such as the bootstrap program

➢ **Secondary storage (ex: HDD):**

- an extension to the main memory
- holds large quantities of data permanently
- nonvolatile
- most programs are stored in it until they are loaded into memory

➢ **Caches:**

- installed to improve performance
- Caching is the use of high-speed memory to hold a copy of recently accessed data, assuming that it will be needed again soon
- in multiprocessor computers every processor has a cache memory
- Cache coherency refers to insuring that an update to some data in a cache is reflected immediately in other caches containing a copy of that data

- **I/O Structure:**

➢ **Device controller:**

▪ maintains some local buffer storage and a set of special purpose registers

▪ responsible for moving data between the devices it controls and its local buffer storage

➢ **Device driver:**

▪ each device controller has a specific device driver

▪ it understands the device controller and provides a uniform interface to the device

➢ **To start an I/O operation:**

1. the device driver loads the appropriate register within the device controller
2. the device controller examines the contents of these registers to determine what action to take
3. the device controller transfers data from the device to its local buffer
4. once the transfer is done, the device controller informs the device driver via an interrupt
5. the device driver returns control to the OS (possibly returning data or a pointer to the data or status information)

▪ one interrupt is generated per byte

▪ this form of interrupt-driven I/O can produce high overhead when used for bulk data movement such as disk I/O

- **To solve this problem:**

◦ Direct Memory Access (DMA) is used:

1. after setting up buffers, pointers, and counters for the I/O device
2. the device controller transfers an entire block of data from its buffer to memory or vice versa with no intervention by the CPU

▪ only one interrupt is generated per block

▪ while the device controller is performing these operations the CPU is available for other work

- **Computer system architecture:**

➢ **Single processor systems:**

▪ one main CPU executes general-purpose instruction set including instructions from user processes

▪ almost all single processor systems have other special purpose processors as well, they: • run a limited instruction set

- do not run user processes
- do not turn a single processor system into a multiprocessor system

➢ **Multiprocessor systems:**

▪ two or more processors in close communication ▪

known as parallel systems or multi-core systems ▪

have three main advantages:

- **Increased throughput:** by increasing the number of processors we expect to get more work done in less time
- **Economy of scale:** multiprocessor systems cost less than equivalent multiple single-processors systems

- **Increased reliability:** the failure of one processor will not halt the system, only slow it down
  - **Graceful degradation:** the ability to continue providing service proportional to the level of surviving hardware
  - **Fault tolerant systems:** can suffer a failure of a component and still continue operation
  - **Fault tolerance:** requires a mechanism to allow the failure to be detected, diagnosed, and, if possible, corrected

▪ **of two types (asymmetric / symmetric):**

- **asymmetric multiprocessing:**
  - defines a boss-worker relationship
  - The boss processor schedules and allocates work to the worker processors
  - Worker processors look to the boss for instruction or have predefined tasks

- **symmetric multiprocessing (SMP):**
  - used by most systems
  - no boss-worker relationship
  - All processors are peers, where each processor can perform any task

- **Multicore Systems:**
  - Multiple computing cores on a single chip
  - More efficient than multiple chips with single cores

        - On-chip communication is faster than between-chip communication
        - One chip with multiple cores uses less power than multiple single-core chips
    - Multicore systems are multiprocessor systems but not all multiprocessor systems are multicore systems
- **Clustered Systems:**
    - known as loosely-coupled systems
    - Composed of two or more individual systems (or nodes)
    - Each node may be a single processor system or a multicore system
    - Clustered computers share storage and are closely linked via a LAN
    - **asymmetric clustering:**
        - One node is in hot-standby mode monitoring the active server
        - If the active server fails, the hot-standby node becomes the active server
        - Supports high-availability service, where a node failure does not stop service
    - **symmetric clustering:**
        - Two or more nodes are running applications and are monitoring each other
        - Supports high performance computing better than multiprocessor systems
        - An application can run concurrently on all cluster nodes using parallelization

- Parallelization divides a program into separate components to run in parallel

- **Dual mode operation:**
  - ➢ Protects the operating system from errant users
  - ➢ Two modes of operation: user mode and kernel mode
  - ➢ Kernel mode = supervisor mode = system mode = privileged mode
  - ➢ User defined code must execute in user mode
  - ➢ Only the operating system can execute in kernel mode
  - ➢ At system boot, the hardware starts in kernel mode, then the OS is loaded and starts user applications in user mode
  - ➢ when a trap or interrupt occurs, the hardware switches to kernel mode
  - ➢ A mode bit indicates the current mode: kernel (0) or user (1)
  - ➢ All instructions that may cause harm are deigned as privileged and can only be executed in kernel mode

- **Multi-Mode Operation:**
  - ➢ The concept of modes can be extended beyond two modes
  - ➢ A CPU that supports virtualization can have a third mode for VMM
  - ➢ VMM mode provide more privileges than user but fewer than kernel

- ➢ VMM needs these privileges so it can create and manage virtual machines
- ➢ Sometimes, different modes are used by various kernel components

- **Timer:**
  - ➢ The OS must prevent a user program from running too long
  - ➢ A timer can be set to interrupt the system after a specified period
  - ➢ A timer is generally implemented by a fixed-rate clock and a counter
  - ➢ The OS sets the counter before turning the control to a user program
  - ➢ Every time the clock ticks, the counter is decremented
  - ➢ When the counter reaches 0, an interrupt occurs and control return to the OS
  - ➢ The OS may treat the interrupt as a fatal error or give the program more time
  - ➢ Clearly, instructions to modify the content of the timer are privileged

- **Computing Environment:**
  - ➢ **Traditional computing:**

▪ Computer systems started as mainframe computers

▪ Mainframes were used primarily by large organizations

▪ Mainframe computers have evolved from batch systems to multiprogramming systems, and then time-sharing systems

▪ Later, desktop computers were introduced and gained popularity

▪ **Batch Systems:**

- Processed jobs in bulk, one job after another
- the user prepares a job and submits it to the operator which then batches jobs with similar needs to reduce setup time
- the main task of OS (resident monitor) : perform automatic job sequencing
- Input devices were card readers and tape drives
- Line printers output results and a memory dump
- If the running job needs to wait for an I/O operation, the CPU remains idle waiting for the job
- Disadvantages:

◦ Low CPU utilization as the CPU is often idle ◦ There is no direct interaction between user and system ▪ **Multiprogramming Systems:**

- Several jobs are kept in main memory
- The CPU is always executing a job (no idle time)
- If the running job needs to wait for an I/O operation, the OS picks another job to start/continue execution
- This requires the OS to provide several features:

◦ I/O routines: only the OS can perform I/O

◦ CPU scheduling: to selecting a job for execution

◦ Memory management: to allocate memory to several jobs

ij

- Resource allocation: no job can use resources of other jobs
- Increased CPU utilization but still no direct interaction

- still does not support direct interaction between the user and the computer system

▪ **Timesharing Systems:**

- An interactive (or hands-on) computer system:
  - Provides direct communication between the user and the system
  - Should be responsive (i.e. response time to user input should be minimal)
- A time-sharing (or multitasking) system:
  - An interactive computer system
  - An extension of multiprogramming systems
- The CPU switches between jobs so frequently that the user can interact with each program while it is running.

▪ **Desktop Computers:**

- A personal computer dedicated to a single user
- I/O devices: keyboards, mice, display screens, small printers
- OS focuses on achieving user convenience and responsiveness
- Less focus on advanced CPU utilization and protection features
- Desktop computer OSs include Windows, Mac OS, UNIX, and Linux. 4

▪ Mobile Computing:

- Computing on handheld smartphones and tablet computers
- has limited memory and processing speed
- These devices are portable, lightweight, and battery-powered
- A mobile device can provide features that are either unavailable or impractical on a desktop or laptop computer, such as:
  - GPS: to determine precise location of the device on Earth
  - Accelerometers: to measure linear acceleration in different directions
  - Gyroscopes: to detect orientation of the device with respect to the ground
- Two OSs dominate mobile computing:
  - Apple iOS and Google Android

▪ Distributed Computing:

- the use of a distributed systems to solve single large problems
- may take the form of client-server computing or peer-to-peer computing
- Computer networks are characterized by distances between nodes:
  - Personal-area network (PAN): wireless links over a short distance
  - Local-area network (LAN): links nodes within a room, building, or campus

- Metropolitan-area network (MAN): connects computer nodes within a city
- Wide-area network (WAN): links nodes within buildings, cities, or countries

- A distributed system is a collection of physically separate, possibly heterogeneous, computer systems that are networked together
- A network is a communication path between two or more systems

▪ Client-Server Computing:

- A model of distributed systems
- Server systems satisfy requests of client systems
- A server can be categorized as a compute-server or a fileserver
- File-servers allows clients to create, update, read, and delete files
- Compute-servers receive client requests, execute actions, and return results
- In client-server systems, a server is a bottleneck

▪ Peer-to-Peer Computing:

- Another model of distributed computing
- A node must first join the peer-to-peer system
- All nodes are peer, where each node may act as
  - A server, when providing services to other nodes

- A client, when requesting services from other nodes

- In P2P systems, several nodes can provide the service

▪ **Virtualization:**

- Allows for creating a virtual machine that acts like a real computer
- A single computer (host) may run several virtual machines (guests)
- The host and the guests share the hardware, but each has its own OS
- The virtual machine manager runs the guest machines, manages their resource use, and protects each guest from the others

▪ **Cloud Computing:**

- Delivers computing as a service and users pay based on usage
- Uses virtualization to run millions of VMS on thousands of servers
- A cloud service can be categorized as public, private, or hybrid cloud
  - A public cloud is available to anyone via the Internet
  - A private cloud is used only by the company owning it
  - A hybrid cloud includes both public and private cloud components
- It may provide infrastructure, platform, or application as a service.
  - Software as a service (SaaS) provides applications (e.g.

Google Docs)

- ◦ Platform as a service (PaaS) provides a software stack (e.g. Google Firebase)
- ◦ Infrastructure as a service (IaaS) provides servers or storage over the internet

- Internet connectivity requires security like firewalls • Load balancers spread traffic across multiple applications

▪ **Real-Time Embedded Systems:**

- Embedded computers are primitive with very specific tasks
- Their OSs provide limited features with little or no user interface
- Embedded systems almost always run real-time operating systems
- A real-time system has well-defined, fixed time constraints
- Processing must be done within the defined constraints, or the system fails
- Examples: weapons systems, nuclear plant systems, medical imaging systems, and industrial control systems