Summary of chapter 1

* An operating system is an software that manages the computer hardware as well as providing an environment for the application program to run.
* Main memory is the only large storage area that processor can access directly.for a computer to do its job of processing programs, that programs must be in the main memory

-each byte of the memory has its own address.  
 -the main memory is a volatile storage device that usually lost its contents when the power is turned off/lost.

* To solve this problem, computer system provides secondary storage ,extension of the main memory, which is nonvolatile storage that is capable of holding large quantities of data permanently.
* Most common secondary storage is MAGNETIC DISK- provides storage of both programs and data
* The wide range of storage system of computer can be organized in a hierarchy(on basis of speed and cost) where-

-higher level of storage -> expensive but fast

- lower level storage-> cost per bit DECREASES but access time INCREASES

* Different strategies for designing a computer system—
* Single processor system→ have only 1 processor.
* Multiprocessor system→contain 2/more processor, shares physical memory and peripheral devices

→ Most common multiprocessor design in SMP-Symmetric multiprocessing-where all processors are considered peers and run independently of one another.

→ Specialized form of Multiprocessor system is CLUSTERED systems,consists of multiple computer systems connected by a local area network

* To utilize the CPU, modern operating systems employ MULTIPROGRAMMING, which allows several jobs to be in the memory at the same time, ensuring that CPU always has a job to execute!

→ Time sharing systems are extension of MULTIPROGRAMMING, where CPU scheduling algorithm rapidly switch between the jobs-giving an illusions that each jobs running concurrently

* To ensure correct operation of computer system, operating system uses-

→ dual mode and privileged instructions— preventing user programs form interfering with proper operation of computer system, hardware has 2 modes-

* KERNEL mode → Instructions like(I/O) instructions and Halt instructions privileged and executed in this mode
* USER mode

→ Memory Protection- memory in which OS resides need to be protected by the modification by the user

→ Timer interrupt- timer prevents Infinite loops

* Process/Job fundamental unit of work in OS. Process management includes-
* Creating or deleting process
* Providing mechanism for processes to communicate and synchronize with each other.
* Memory management of OS do it by

-keeping track of which parts of memory being used and by whom

-dynamically allocating and freeing memory space

-storage space is also managed by providing file system for representing file and directories ; managing space on mass-storage devices

* Protection and securing OS and user by
* Protection measures Control access of process or user to the resource made available by Computer system
* Security measures responsible for protecting the computer system from internal and external attacks
* Several data structure that used in OS- list queue stack hash trees map bitmap
* Computing in diff environments-

-mobile computing refers to computing on smartphones tablet

-Distributed computing systems allow to share resources on geographically dispersed hosts connected through computer networks

-Service 2 types : peer to peer, client-server

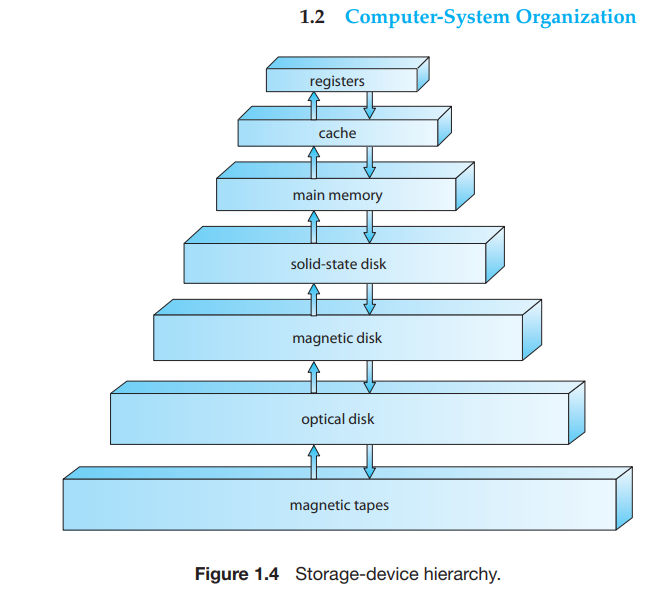
-Cloud computing uses distributed system to abstract system into “cloud”

Where user access the service from remote location

-real time OS designed for embedded environments- consumer devices, automobiles, robotics

* GNU\Linux & BSD UNIX open source operating system
* Kernel
* Middleware
* Bootstrap program
* System processes /system daemons
* System call/monitor call
* Interrupt vector
* Dram (dynamic random access memory)
* Static program like Bootstrap program are stored in ROM
* CPU automatically loads instruction from main memory to execute
* VON-NEUMANN architecture, instruction register
* We want the programs and data to reside in the main memory permanently but itsnot possible for 2 reasons-

1. Main memory is usually too small to store all needed programs and data permanently.
2. Main memory is Volatile storage device that loses its contents when power is turned off.

* Main requirement for secondary storage is that it be able to hold large quantities of data permanently.
* Magnetic disk provides storage for both programs and data.most program are stored on disk until they are loaded to memory.thats why many programs use the disk as the source and destination of their processing
* Some other storage system is cache, CD-ROM, Magnetic tapes etc
* 
* Hierarchy is (lower level) Magnetic tapes→optical disk→magnetic disk→solid state disk→main memory→cache→register(higher level)
* Top 4 level of this hierarchy can be constructed using semiconductor memory
* In absence of generator backup system and expensive battery data must be written into non-volatile storage for safekeeping
* The storage system above Solid state is volatile and from solid state rest are non-volatile
* PDA personal digital assistant
* NVRAM another form of nonvolatile storage which is DRAM with battery backup power→ as fast as DRAM (as long as the battery lasts)

I/O structure(1.2.3)

* In a general purpose computer system consists cpus and multiple decide controllers connected through common bus
* 7 or more devices can be attached to the Small Computer-System Interface (SCCI)controller
* A device controller maintains→ some local buffer storage and a set of special purpose registers. Device controller responsible for moving data between the peripheral device (which its controlling) and its local buffer storage.
* OS have a device driver for each device controller→ it understands the divide controller and provides rest of the operating system with a uniform interface to the device.
* How I/O operation works in the computer system ?  
    
  → to start an I/O operation , device driver loads necessary registers within the device controllers. Device controller check the registers to determine what operation it needs to do( such as “read a character from the keyboard”). The controller starts the transfer of the data from the device to its local buffer.Once the transfer of the data is complete, the device controller informs the device driver via an Interrupt that it finished its operation. The device driver thn returns its control to the operation system , possibly returning the data or pointer to the data if the operation was a read, for other operations it returns status information

For bulk data movement such as disk I/O DMA( direct memory access) is used. After setting up buffer pointers counters for the I/O device, the device controller transfers an entire block of data directly to or from its own buffer storage to memory with no intervention by the CPU. only 1 interrupt is generated per block, to tell the device that the operation has been completed.

1.3.1 Single-Processor Systems

* In a single processor system there is one main CPU which executes general purpose instruction and user processes .
* Almost all single processor system have Special\_Purpose Processor
* Come in device-specific processors-disk,keyboard,graphics controller
* On mainframe they take form of more General purpose processors (i/o processors that move data rapidly among the component of system)
* specia l purpose processors run a limited instruction set and do not run user processes.
* They are managed through the operating system,OS sends them information abt their next task and monitors them.

E.g- disk-controller microprocessor receives sequence of requests from CPU and implements its own disk queue and scheduling algorithm

* The operating system cant communicate with these special- purpose processors-so they do their job autonomously. The use of special purpose microprocessor is common and doesnt turn a single processor into multi process
* If there is ONLY 1 GENERAL PURPOSE CPU-then system is single-processor.

1.3.2 Multiprocessor Systems/parallel system/ multicore system

overhead-

In computer science and technology, overhead can refer to the extra time, memory, processing power, or communication needed to manage tasks such as coordination, communication, or resource management. For example, when multiple processors work together on a task, there is overhead involved in synchronizing their actions, passing data between them, and ensuring they don't interfere with each other. Similarly, in networking, overhead can refer to the extra data transmitted for error checking or protocol management

contention-

Additionally, if both people need to reach for pieces from the same box (shared resource), they might accidentally slow each other down as they both try to grab pieces simultaneously. This is a contention for shared resources.

* Such processors have 2 or more processors (Or, CPUs) in close communication sharing the computer bus and sometimes clock, memory and peripheral devices.

\*\*\*\*\*\* multiprocessor system have 3 main advantages-

1. Increased throughput-The speed up ration with N processors is not N, it’s less then N.because of certain amount of overhead incurred in keeping all parts working properly.plus Contention for shared resources, lowers the expected gain from additional processors
2. Economy of scale - multiprocessor systems can cost less than equivalent multiple single processor systems, bcz they share peripherals, mass storage and power supplies.
3. Increased reliability- if functions can be distributed properly among several processors then the failure of one processor will not halt the system, only it will slow down

Graceful degradation and fault tolerant

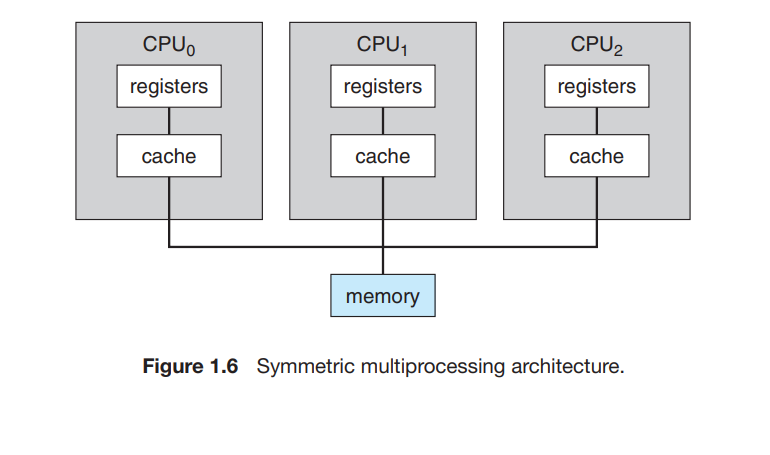
* Graceful degrad means even if some parts of the processor or components fail, the processor will still operate but with reduced speed, capability or functionality
* FAULT TOLERANCE refers to a systems capability to continue functioning properly even when the components fails or errors occur, while also ensuring data integrity and accurate results.It requires a mechanism to allow to failure to be detected , diagnosed and if possible corrected

Types of multiprocessor( symmetric and asymmetric )

* Asymmetric multiprocessing , in which each processor assigned a specific task. It defines boss-worker relationship, where the boss processor controls the system and schedules and allocates work to the worker processors.
* Symmetric Multiprocessing( the most common systems use this)\*\*

Each processor performs all the tasks under OS, all processors are peers.each processor have its own register, private/local cache.All processors share physical memory.

Benefit -> many processors can work simultaneously -N processes can run if there are N cpus without causing performance deteriorate significantly but need to control i/o to ensure the data reach the appropriate processor.



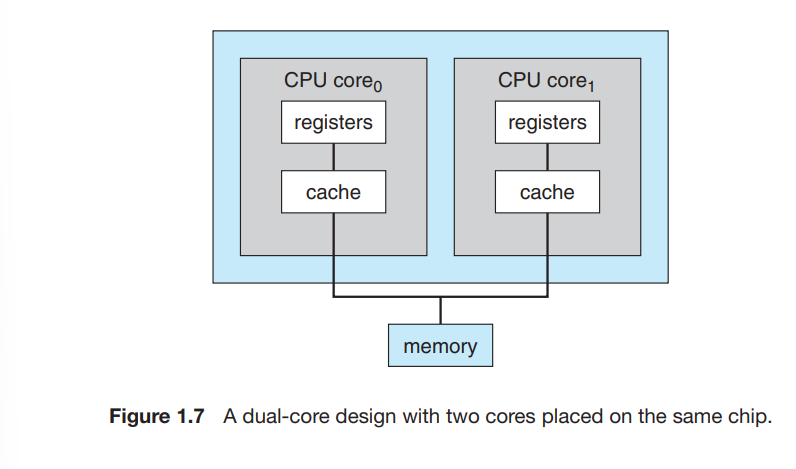
Multiprocessor vs multicore

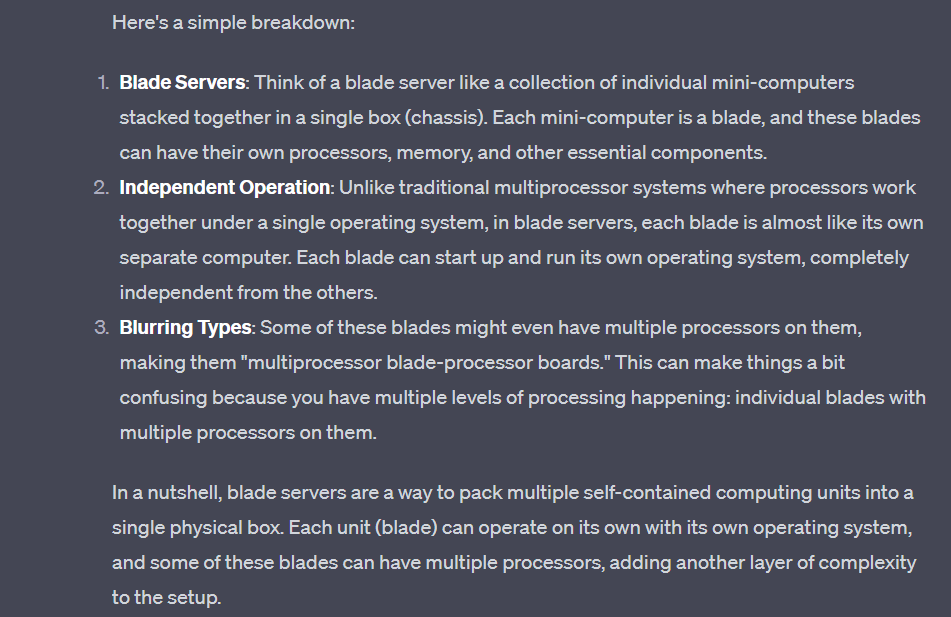
\*\*\*all multicore system are multiprocessor but not all multiprocessor systems are multicore.

* Multiple processor cores are integrated on a single physical chip, each core have its own execution resource and pipeline, just like separate processors in the multiprocessor.
* As they on the same chip, they communicate more efficiently and share resources like cache at higher speed compared to separate processors in a multiprocessor system.
* In addition one chip with multiple core uses significantly less power than multiple single core chips

Blade servers

Modern type of server setup which have multiple processor boards i/o boards and networking boards placed in the same chassis. Here each blade processor board boots independently and run its own operating system.





Cluster (from chatgpt)

Cluster systems are composed of 2 or more individual node joined together. These nodes are storage and are connected via LAN.can be used for high availability services and high performance computing

Are designed to provide continuous services even if one or more nodes fail.cluster software monitors nodes.

Clusters are 2 types; asymmetric and symmetric

* In asymmetric clustering one machine is on standby while the other runs applications the standby one take over if the active one fails
* In symmetric clustering multiple hosts run applications and monitor each other, utilizing hardware efficiently

Operating system operations

Process management

Program is a passive entity like the content in the file stored on disk, and process is an active entity

He operating system is responsible for the following activities in the connection with process management

* Scheduling process and threads in the cpu
* Creating and deleting user and system process
* Suspending resuming process
* Providing mechanism for process synchronization
* Providing mechanisms for process communication

File management

* Creating deleting file
* Creating and deleting directories to organize files
* Supporting primitives for manipulating files and directories
* Mapping files onto secondary storage
* Backing up file on stable(non volatile) storage media

1.7 memory management

OS responsible for

* Keeping track of which parts of memory are currently being used and who is using thm
* Deciding which process and data to move into and out of memory
* Allocating and deallocating memory space

Peer-to-peer (P2P) computing is a distributed computing model where participants in the network, known as peers, collaborate directly with each other to share resources, provide services, or exchange information without relying on a central server. In P2P systems, there is no clear distinction between clients and servers; every peer can act as both a client and a server, contributing resources and services to the network.

Client-server computing is a distributed computing model in which tasks and functions are divided between two types of systems: clients and servers. These systems work together to achieve various computing goals in a networked environment. The fundamental idea behind client-server computing is to offload certain tasks to specialized servers, allowing clients (user devices) to focus on presentation, user interface, and user interactions.

Exercise questions practice

1. What are the three main purposes of an operating system?

→ the main 3 purposes are-

* To provide an environment for a computer user to execute program on the computer hardware in a convenient and efficient manner.
* To allocate different resources of the computer to solve the given problem. The allocation process should be fair and efficient as possible
* As a control program, OS serves 2 major functions-1. Supervision of the execution of the user programs to prevent errors and improper use of computer and 2. Management of the operation and the control of the I/O devices

2.We have stressed the need for an operating system to make efficient use of the computing hardware. When is it appropriate for the operating system to forsake this principle and to “waste” resources? Why is such a system not really wasteful?

→single user systems should maximize use of the system for the user.A GUI might “waste” cpu cycles but it optimizes the user interaction with the system,

Form chatgpt-

User Experience and Responsiveness: Sometimes, it's more important to provide a smooth and responsive user experience than to optimize hardware resource usage. For instance, an operating system might allocate additional memory to keep frequently used applications in memory, even if it means leaving some memory unused. This can result in faster application launches and smoother multitasking, enhancing the user experience

Background Tasks: Operating systems often have background tasks that run to maintain system health, security, and updates. These tasks might run even if they temporarily consume additional resources, as they contribute to the overall stability and security of the system

In these cases, the system isn't truly wasteful because the resources that might seem unused or "wasted" are contributing to overall system goals, user experience, reliability, and other critical aspects. Balancing efficiency with these considerations ensures that the operating system serves the larger purpose of providing a functional, reliable, and user-friendly computing environment

Quiz question 2018

1. Priviledged instruction

Privileged instructions are instructions that can only be executed in kernel mode (privileged mode). These instructions control and manage hardware resources and critical OS functions, such as modifying memory protection settings, enabling or disabling interrupts, and accessing hardware peripherals directly.

If a user program attempts to execute a privileged instruction while in user mode, a hardware exception (usually a trap or interrupt) is generated. This exception transfers control to the OS, which can then decide how to handle the unauthorized access attempt. Typically, the OS terminates the offending program or returns an error code to the user application.