Types of OS

Evolution of OS:

- The evolution of operating systems went through seven *major phases.*
- Six of them significantly changed the ways in which users accessed computers through the open shop, batch processing, multiprogramming, timesharing, personal computing, and distributed systems.
- In the seventh phase the foundations of concurrent programming were developed and demonstrated in model operating systems.

Evolution of OS (contd..):

Major Phases	Technical Innovations	Operating Systems
Open Shop	The idea of OS Programmers write programs and submit tape/cards to operator. Operator feeds cards, collects output from printer.	IBM 701 open shop (1954)
Batch Processing	Tape batching, First-in, first-out scheduling.	BKS system (1961)
Multi-programming	Processor multiplexing, Indivisible operations, Demand paging, Input/output spooling, Priority scheduling, Remote job entry	Atlas supervisor (1961), Exec II system (1966)

Evolution of OS (contd..):

Timesharing	Simultaneous user interaction, On-line file systems	Multics file system (1965), Unix (1974)
Concurrent Programming	Hierarchical systems, Extensible kernels, Parallel programming concepts, Secure parallel languages	RC 4000 system (1969), 13 Venus system (1972), 14 Boss 2 system (1975).
Personal Computing	Graphic user interfaces	OS 6 (1972) Pilot system (1980)
Distributed Systems	Remote servers	WFS file server (1979) Unix United RPC (1982) 24 Amoeba system (1990)

Batch Systems

- Introduction of tape drives allow batching of jobs
- In this type of system, there is **no direct interaction between user and the computer**.
- The user has to submit a job (written on cards or tape) to a computer operator.
- Then computer operator places a batch of several jobs on an input device.
- Jobs are batched together by type of languages and requirement.

Computer now has a resident monitor :

- initially control is in monitor.
- monitor reads job and transfer control.
- at end of job, control transfers back to monitor.
- The monitor is always in the main memory and available for execution.

Even better: spooling systems.

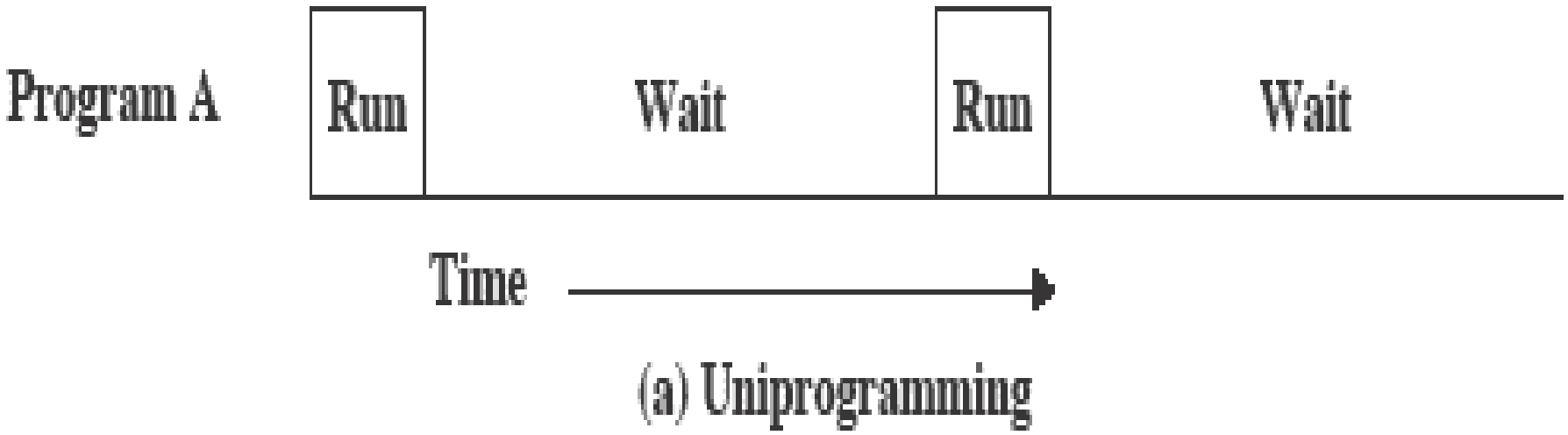
- use interrupt driven I/O.
- use magnetic disk to cache input tape.
- Monitor now schedules jobs. . .

Advantages of Simple Batch Systems

- No interaction between user and computer.
- No mechanism to prioritize the processes.

Uniprogramming

- Processor must wait for I/O instruction to complete before preceding

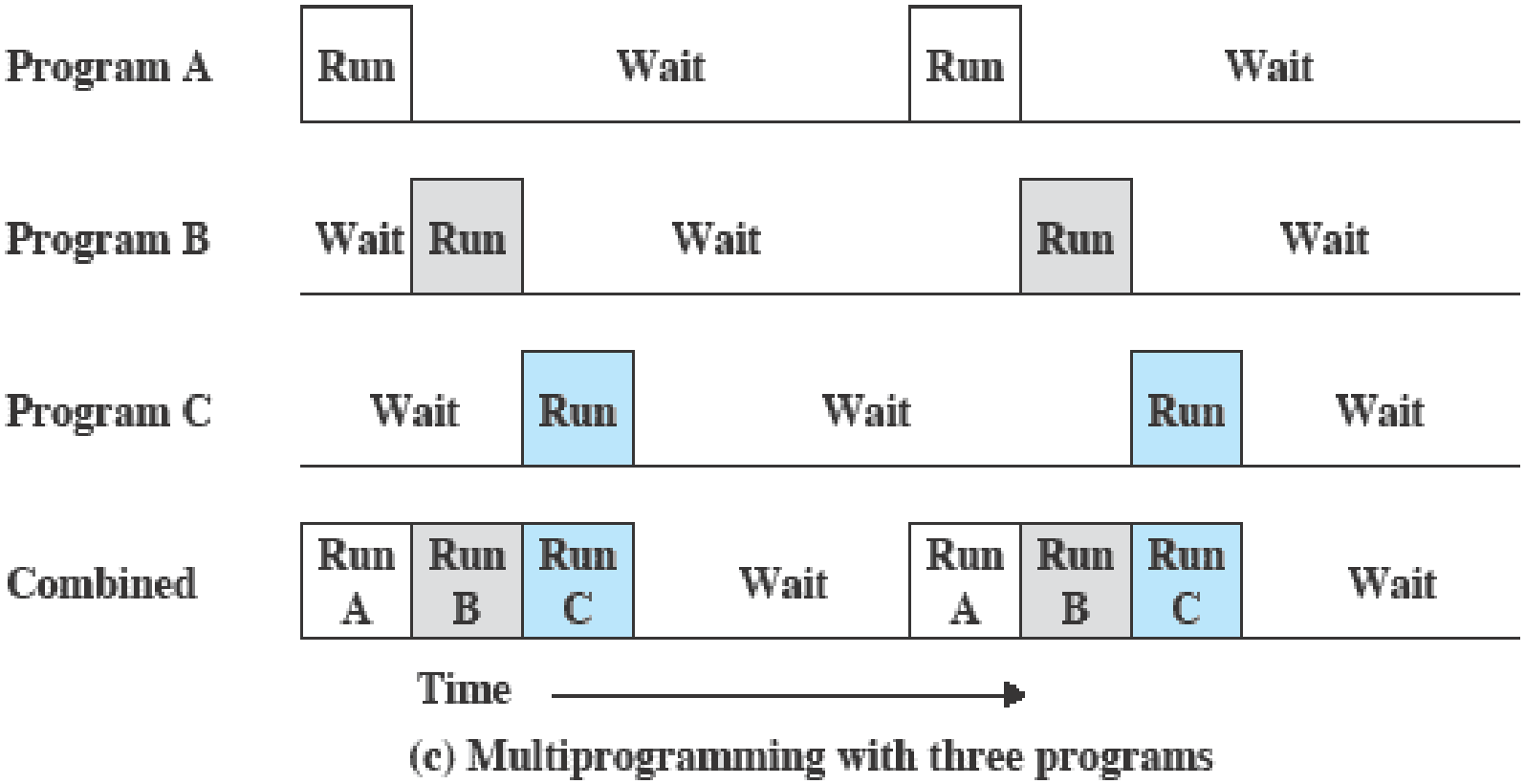


(a) Uniprogramming

Multiprogramming Batch Systems

- 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
- A subset of total jobs in system is kept in memory
- In this, the operating system picks up and begins to execute one of the jobs from memory.
- Once this job needs an I/O operation operating system switches to another job (CPU and OS always busy).
- Jobs in the memory are always less than the number of jobs on disk(Job Pool).

Multiprogramming



(c) Multiprogramming with three programs

- Use memory to cache jobs from disk i.e., more than one job active simultaneously.

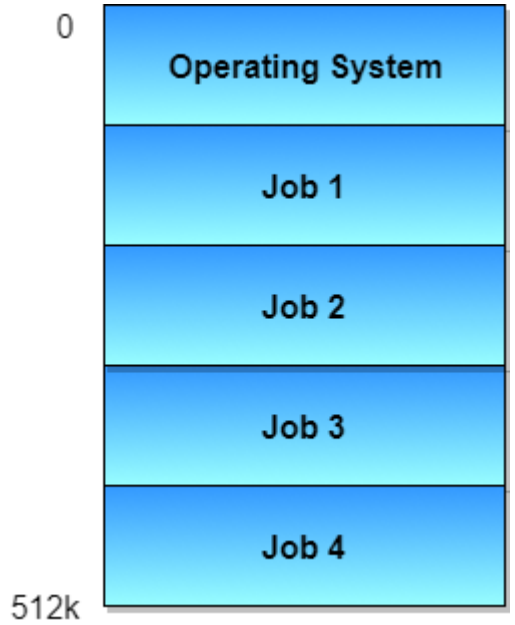
Two stage scheduling:

  - select jobs to load: job scheduling.
  - select resident job to run: CPU scheduling.

- If several jobs are ready to run at the same time, then the system chooses which one to run through the process of **CPU Scheduling**.
- In Non-multiprogrammed system, there are moments when CPU sits idle and does not do any work.
- In Multiprogramming system, CPU will never be idle and keeps on processing.

Advantages:

- Efficient memory utilization
- Throughput increases
- CPU is never idle, so performance increases.



Time Sharing Systems

- **Time Sharing Systems** are very similar to Multiprogramming batch systems.
- **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
- Time slice is defined by the OS, for sharing CPU time between processes.
- Examples: Multics, Unix, etc.,
- In Time sharing systems the prime focus is on **minimizing the response time**, while in multiprogramming the prime focus is to maximize the CPU usage.
  - **Response time** should be < 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 don't fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes not completely in memory

Multiprocessor Systems

- Also known as **parallel systems**, **tightly-coupled systems**
- A Multiprocessor system consists of several processors that share a common physical memory.
- Multiprocessor system provides higher computing power and speed.
- In multiprocessor system all processors operate under single operating system. Multiplicity of the processors and how they do act together are transparent to the others.

Advantages of Multiprocessor Systems

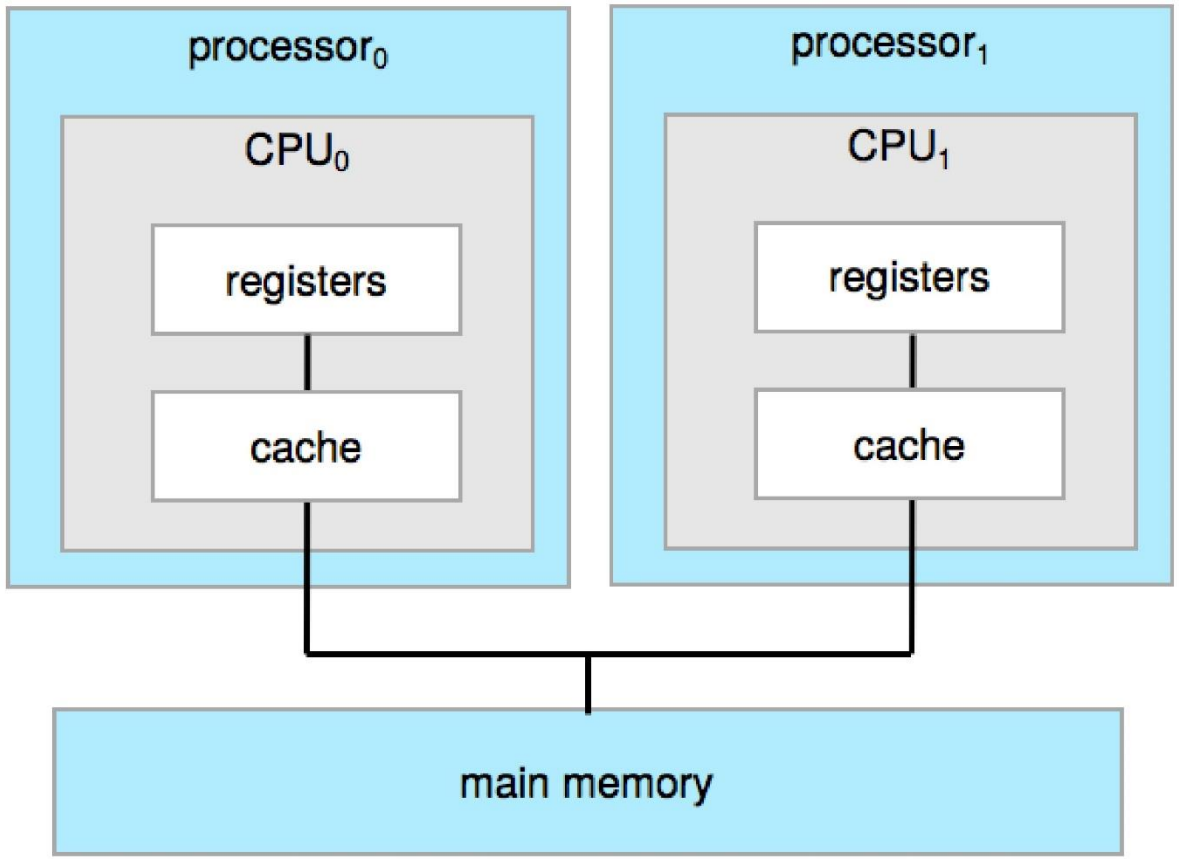
- Enhanced performance
  - Increased throughput **-** Execution of several tasks by different processors concurrently, increases the system's throughput without speeding up the execution of a single task.
  - If possible, system divides task into many subtasks and then these subtasks can be executed in parallel in different processors. Thereby speeding up the execution of single tasks.
- Economy of scale
- Increased reliability – graceful degradation or fault tolerance

Two types:

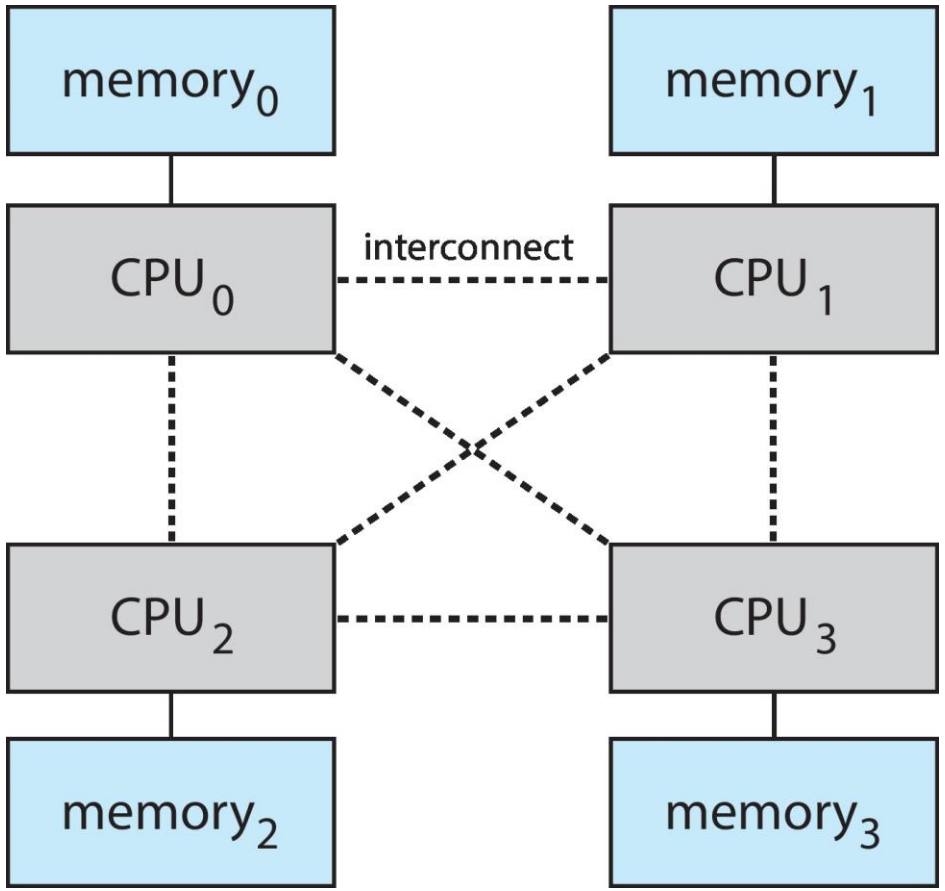
1. **Asymmetric Multiprocessing** – each processor is assigned a specific task.

2. **Symmetric Multiprocessing** – each processor performs all tasks

Symmetric Multiprocessing Architecture



Non-Uniform Memory Access System

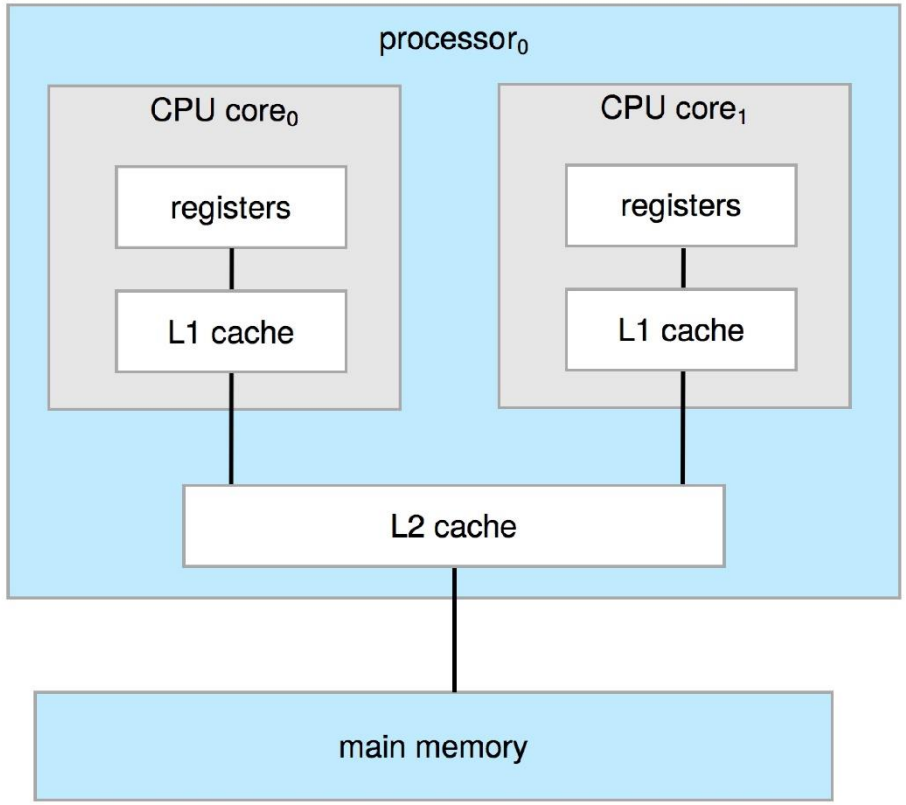


A Dual-Core Design

- A recent trend in CPU design is to include multiple computing **cores** on a single chip.
- Multi-chip and **multicore**
- Systems containing all chips
  - Chassis containing multiple separate systems

Advantages:

- on-chip communication is faster than between-chip communication.
- One chip with multiple cores uses significantly less power than multiple single-core chips.



Blade servers

- These are a relatively recent development in which multiple processor boards, I/O boards, and networking boards are placed in the same chassis.
- The difference between these and traditional multiprocessor systems is that each blade-processor board boots independently and runs its own operating system.
- These servers consist of multiple independent multiprocessor systems.

Desktop Systems

- PC operating systems were neither **multiuser** nor **multitasking**.
- However, the goals of these operating systems have changed with time; instead of maximizing CPU and peripheral utilization, the systems opt for maximizing user convenience and responsiveness. These systems are called **Desktop Systems** and include PCs running Microsoft Windows and the Apple Macintosh.

- Operating systems for these computers have benefited in several ways from the development of operating systems for **mainframes**.
- **Microcomputers** were immediately able to adopt some of the technology developed for larger operating systems.
- On the other hand, the hardware costs for microcomputers are sufficiently **low** that individuals have sole use of the computer, and CPU utilization is no longer a prime concern. Thus, some of the design decisions made in operating systems for mainframes may not be appropriate for smaller systems.

Distributed Operating System

- The motivation behind developing distributed operating systems is the availability of powerful and inexpensive microprocessors and advances in communication technology.
- These advancements in technology have made it possible to design and develop distributed systems comprising of many computers that are inter connected by communication networks. The main benefit of distributed systems is its low price/performance ratio.

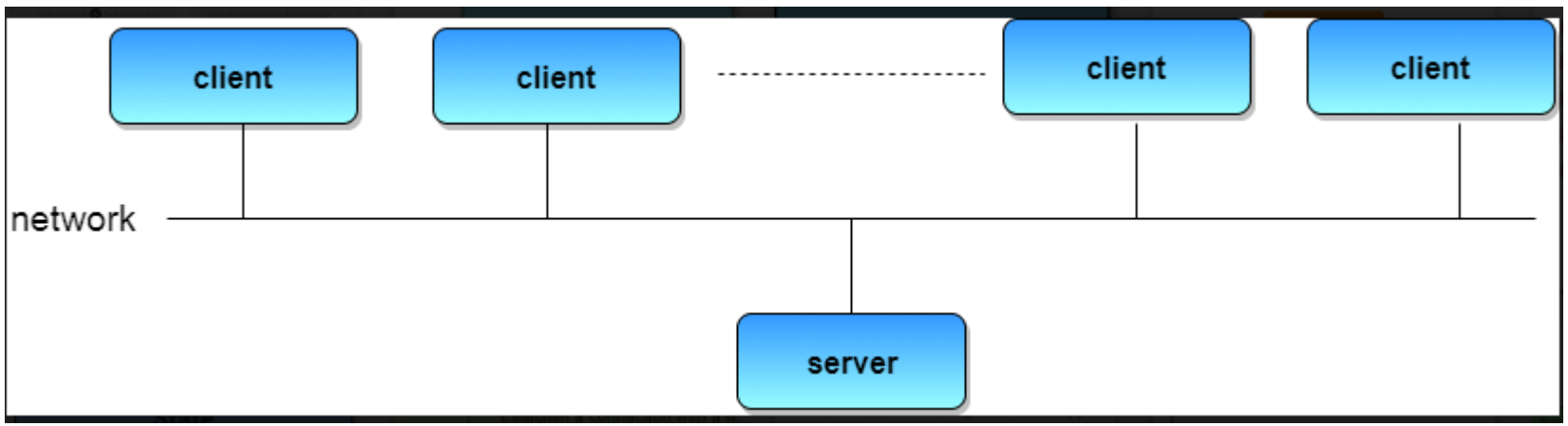
- **Advantages Distributed Operating System**
- As there are multiple systems involved, user at one site can utilize the resources of systems at other sites for resource-intensive tasks.
- Fast processing.
- Less load on the Host Machine.

Types of Distributed Operating Systems

- Following are the two types of distributed operating systems used:
  - Client-Server Systems
  - Peer-to-Peer Systems

Client-Server Systems

- **Centralized systems** today act as **server systems** to satisfy requests generated by **client systems**.
- Server Systems can be broadly categorized as: **Compute Servers** and **File Servers**.
- **Compute Server systems**, provide an interface to which clients can send requests to perform an action, in response to which they execute the action and send back results to the client.
- **File Server systems**, provide a file-system interface where clients can create, update, read, and delete files.


client
client
client
client
network
server

Peer-to-Peer Systems

- The growth of computer networks - especially the Internet and World Wide Web (WWW) – has had a profound influence on the recent development of operating systems.
- When PCs were introduced in the 1970s, they were designed for **personal** use and were generally considered standalone computers.
- With the beginning of widespread public use of the Internet in the 1990s for electronic mail and FTP, many PCs became connected to computer networks.

- In contrast to the **Tightly Coupled** systems, the computer networks used in these applications consist of a collection of processors that do not share memory or a clock. Instead, each processor has its own local memory.
- The processors communicate with one another through various communication lines, such as high-speed buses or telephone lines. These systems are usually referred to as loosely coupled systems (or distributed systems).

Clustered Systems

- Like parallel systems, clustered systems gather together multiple CPUs to accomplish computational work.
- Clustered systems differ from parallel systems, however, in that they are composed of two or more individual systems coupled together.
- The definition of the term clustered is **not concrete;** the general accepted definition is that clustered computers share storage and are closely linked via LAN networking.
- Clustering is usually performed to provide **high availability**.

- A layer of cluster software runs on the cluster nodes. Each node can monitor one or more of the others.
- If the monitored machine fails, the monitoring machine can take ownership of its storage, and restart the application(s) that were running on the failed machine.
- The failed machine can remain down, but the users and clients of the application would only see a brief interruption of service.

- **Asymmetric Clustering -** In this, one machine is in hot standby mode while the other is running the applications. The hot standby host (machine) does nothing but monitor the active server. If that server fails, the hot standby host becomes the active server.
- **Symmetric Clustering -** In this, two or more hosts are running applications, and they are monitoring each other. This mode is obviously more efficient, as it uses all of the available hardware.
- **Parallel Clustering -** Parallel clusters allow multiple hosts to access the same data on the shared storage. Because most operating systems lack support for this simultaneous data access by multiple hosts, parallel clusters are usually accomplished by special versions of software and special releases of applications.

- Clustered technology is rapidly changing. Clustered system's usage and it's features should expand greatly as **Storage Area Networks(SANs)**. SANs allow easy attachment of multiple hosts to multiple storage units.
- Current clusters are usually limited to two or four hosts due to the complexity of connecting the hosts to shared storage.

Real Time Operating System

- It is defined as an operating system known to give maximum time for each of the critical operations that it performs, like OS calls and interrupt handling.
- The Real-Time Operating system which guarantees the maximum time for critical operations and complete them on time are referred to as **Hard Real-Time Operating Systems.**
- While the real-time operating systems that can only guarantee a maximum of the time, i.e. the critical task will get priority over other tasks, but not assured of completing it in a defined time. These systems are referred to as **Soft Real-Time Operating Systems**.

Handheld Systems

- Handheld systems include **Personal Digital Assistants(PDAs)**, such as Palm-Pilots or Cellular Telephones with connectivity to a network such as the Internet.
- They are usually of limited size due to which most handheld devices have a small amount of memory, include slow processors, and feature small display screens.

- Processors for most handheld devices often run at a fraction of the speed of a processor in a PC. Faster processors require **more power**. To include a faster processor in a handheld device would require a **larger battery** that would have to be replaced more frequently.
- The last issue confronting program designers for handheld devices is the small display screens typically available. One approach for displaying the content in web pages is **web clipping**, where only a small subset of a web page is delivered and displayed on the handheld device.