Operating Systems

Dr. Pritee Parwekar

Operating System

- An interface between applications/users and hardware
- A program that controls the execution of application programs

Main objectives of an OS:
• Convenience • User Friendly • Secure • Efficiency • Ability to evolve

Layers of Computer System

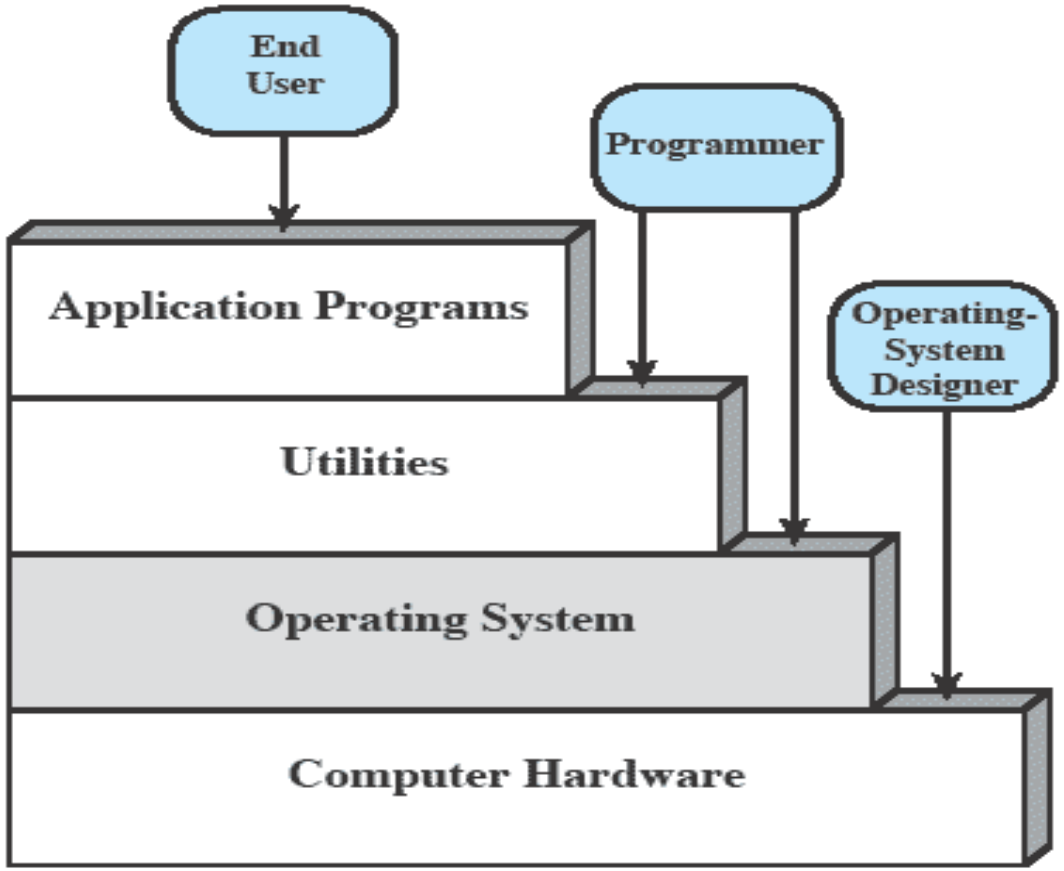
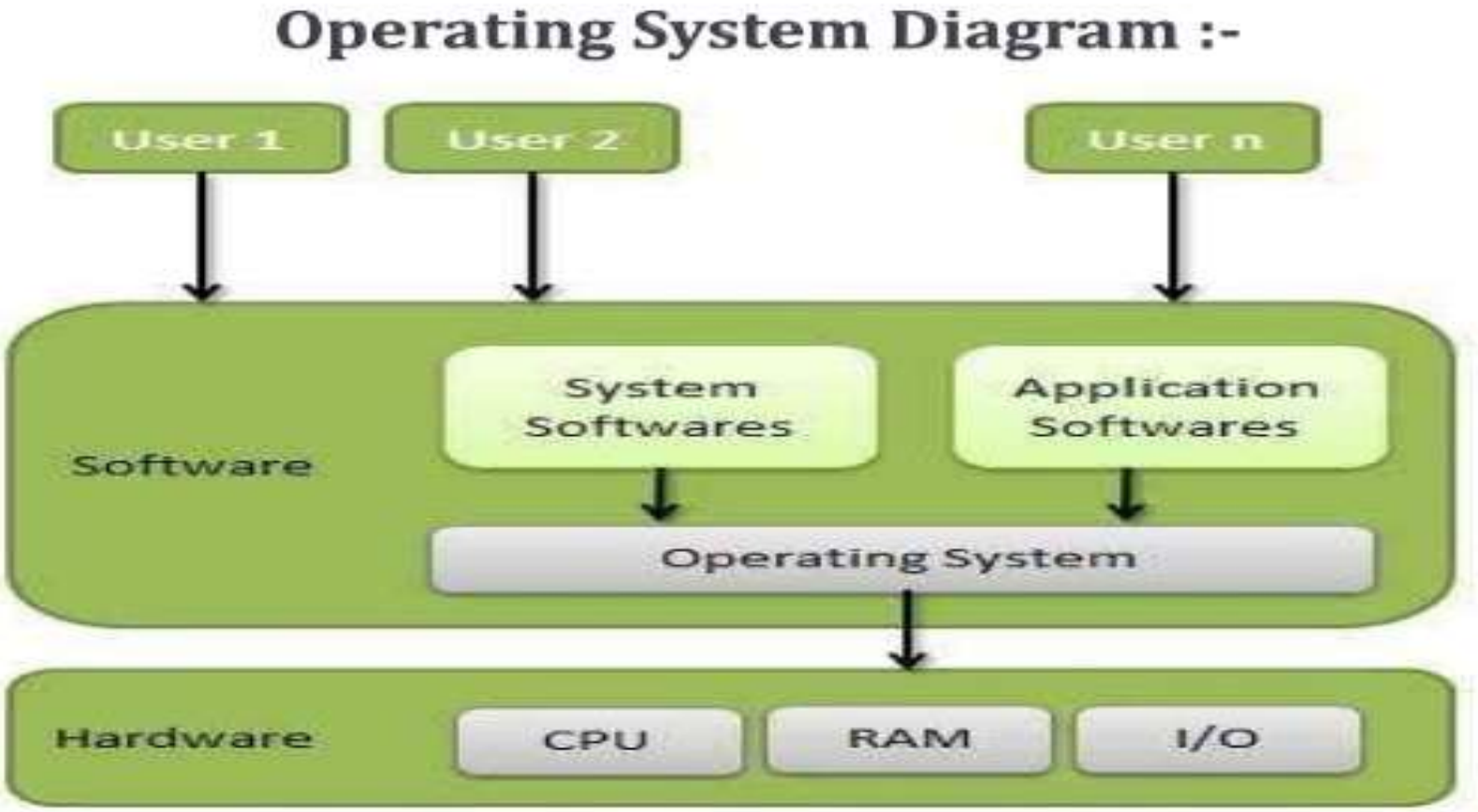


Figure 2.1 Layers and Views of a Computer System

The OS as a User/Computer Interface



Computer Hardware and Software Infrastructure

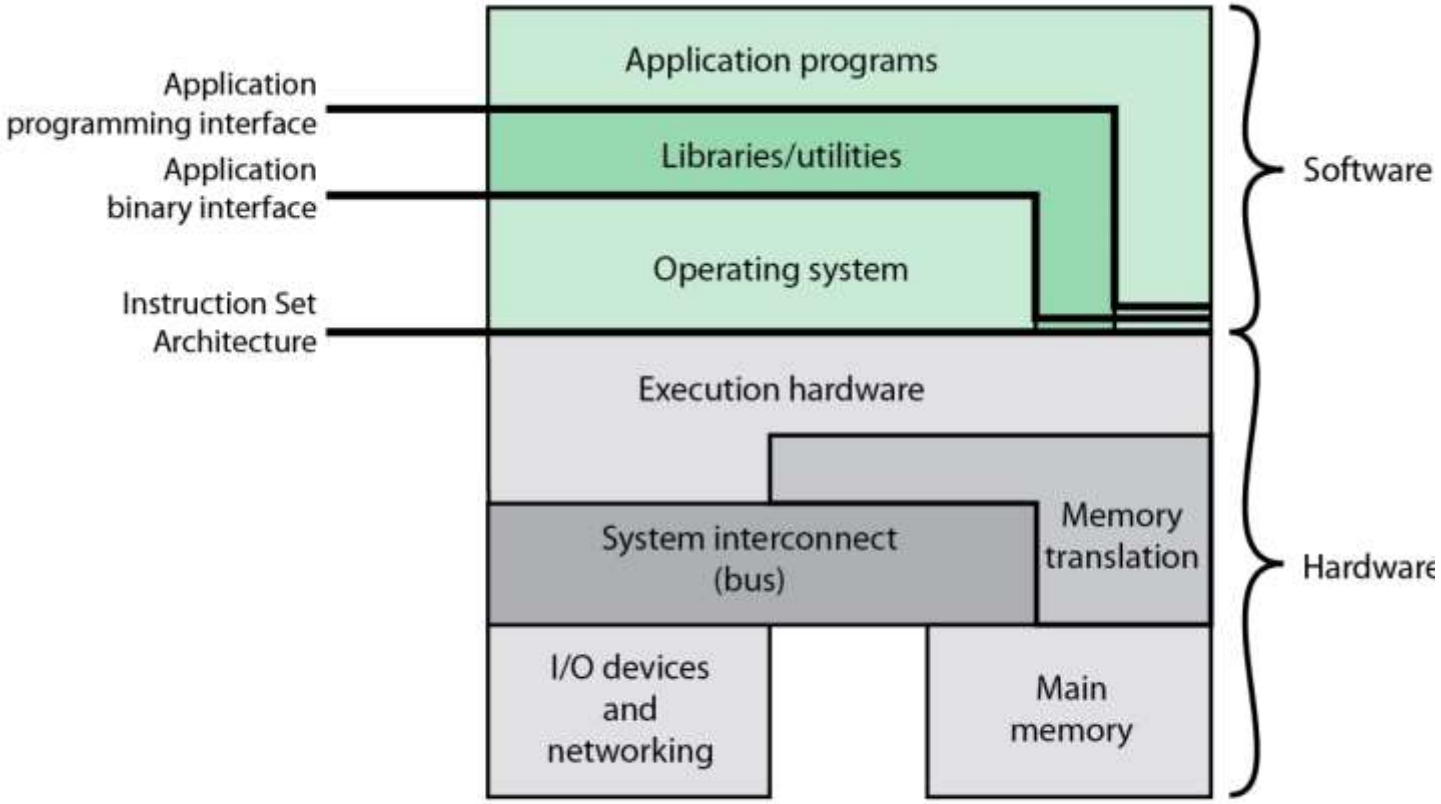


Figure 2.1 Computer Hardware and Software Infrastructure

Operating System Services

- Program development-Editors and debuggers.
- Program execution - OS handles scheduling of numerous tasks required to execute a program
- Access I/O devices - Each device will have unique interface OS presents standard interface to users
- Controlled access to files-Accessing different media but presenting a common interface to users
  - Provides protection in multi-access systems
- System access-Controls access to the system and its resources
- Error detection and response- Internal and external hardware errors
  - Software errors
  - Operating system cannot grant request of application
- Accounting-Collect usage statistics ,Monitor performance

Role of Operating System

- Resource Management
- Process Management
- Memory Management
- I/O Management
- File Management

The Operating System as a Resource Manager

- A computer is a set of resources for moving, storing, & processing data
- The OS is responsible for managing these resources
- The OS exercises its control through software

Operating System as Software

- Functions in the same way as ordinary computer software
- Program, or suite of programs, executed by the processor
- Frequently relinquishes control and must depend on the processor to allow it to regain control

Operating System as Resource Manager

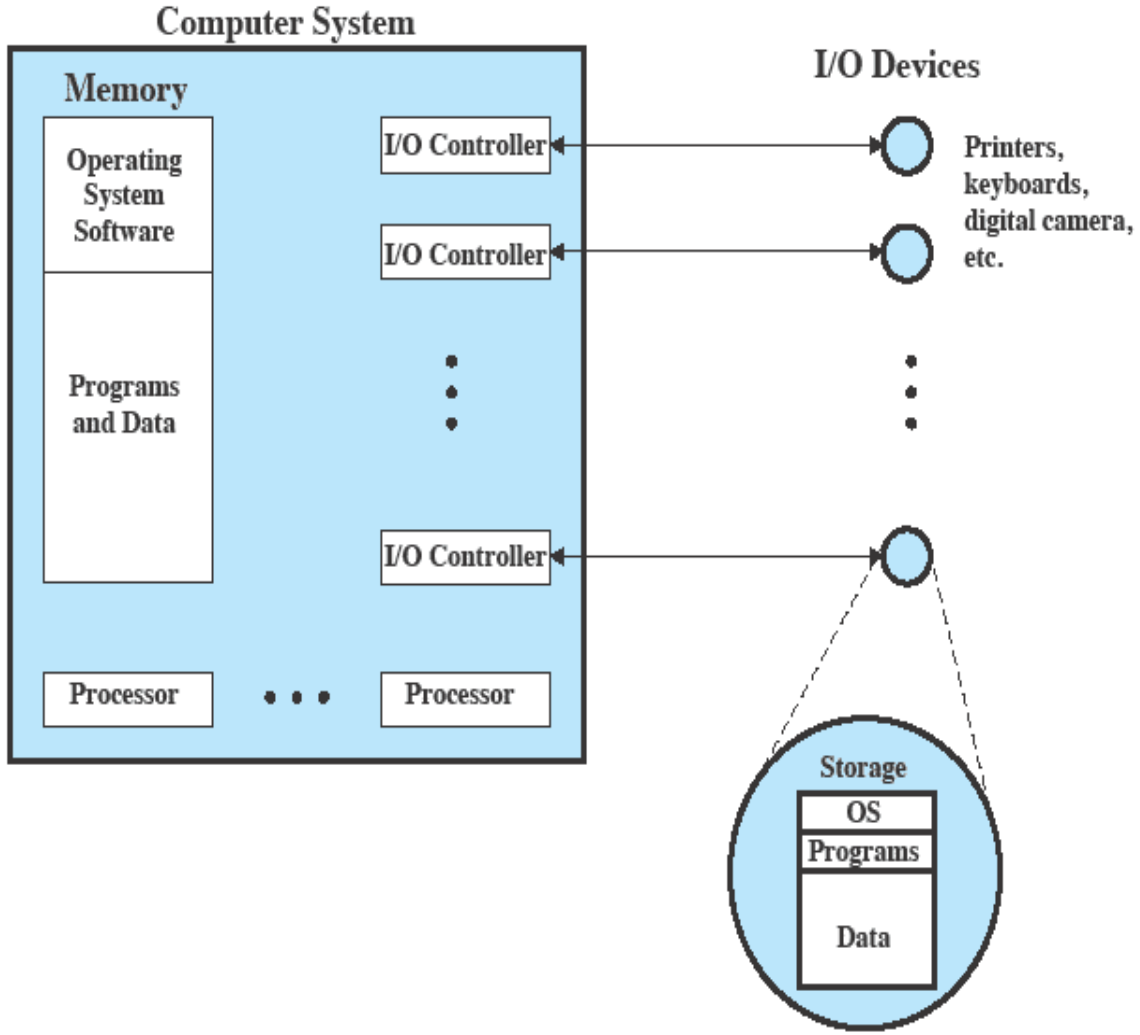


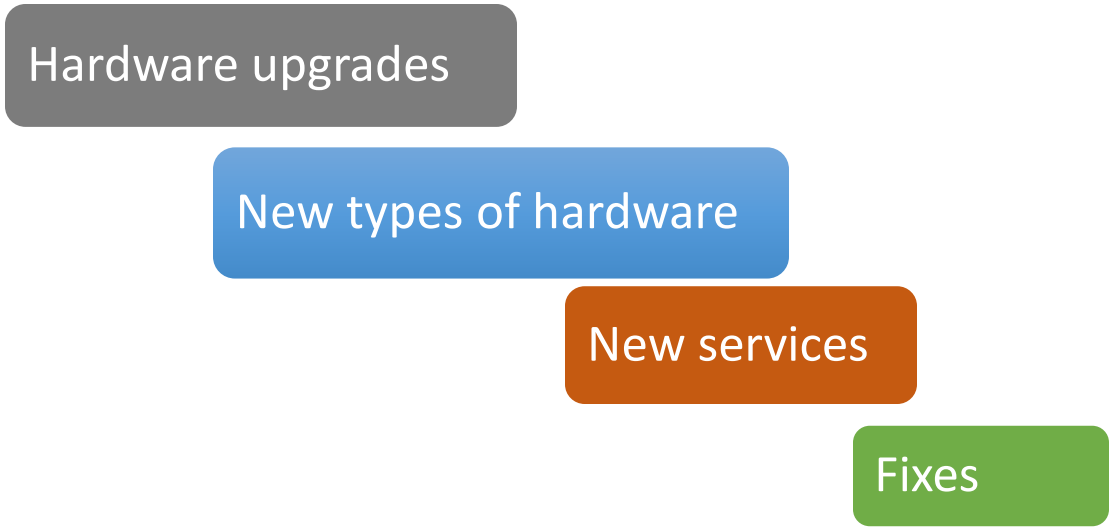
Figure 2.2 The Operating System as Resource Manager

Evolution of Operating Systems

Dr. Pritee Parwekar

Evolution of Operating Systems

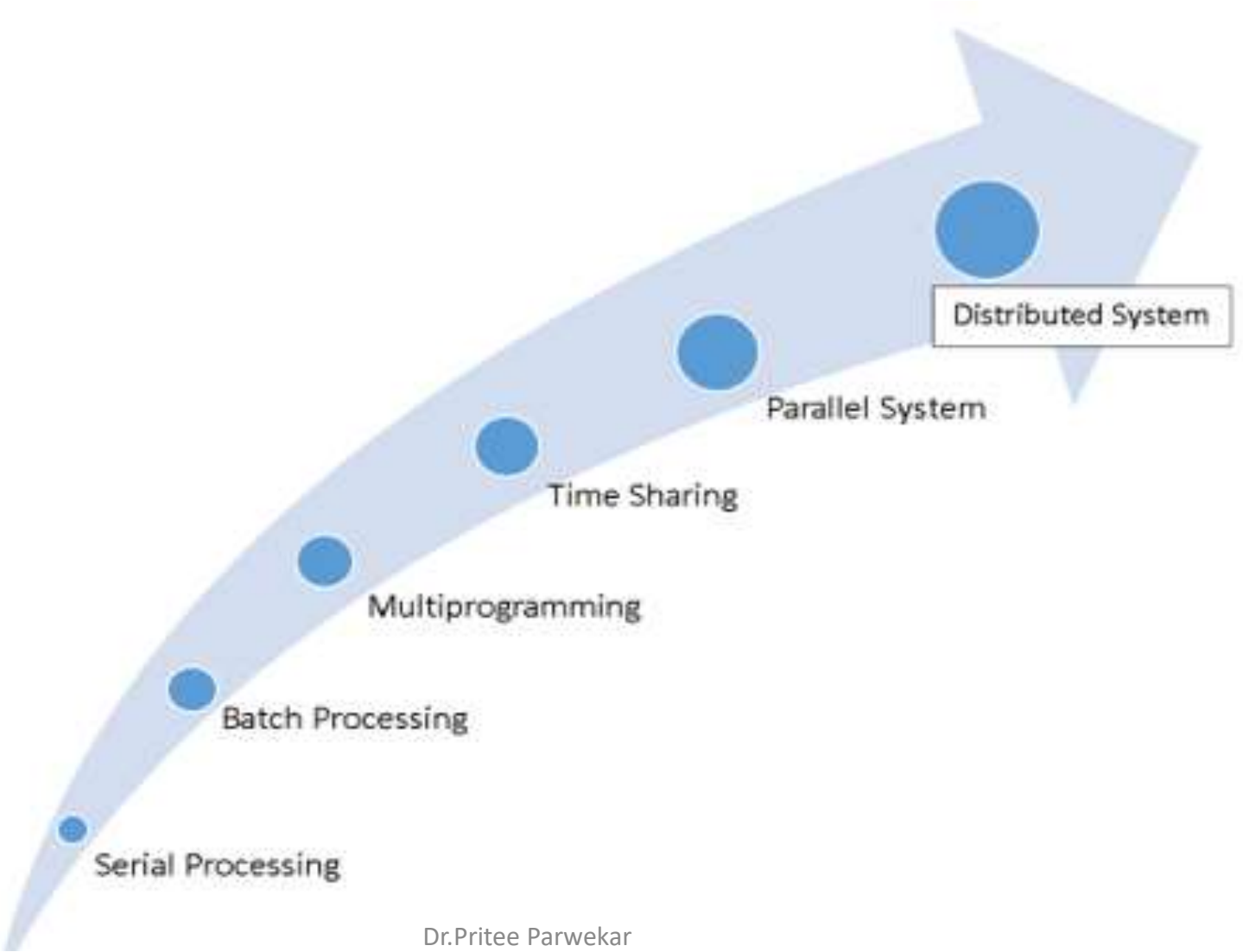
- A major OS will evolve over time for a number of reasons:



Evolution of OS

- Serial Processing
- Batch Processing
- Multiprogramming
- Time Sharing
- Parallel Systems
- Distributed Systems

Evolution of OS



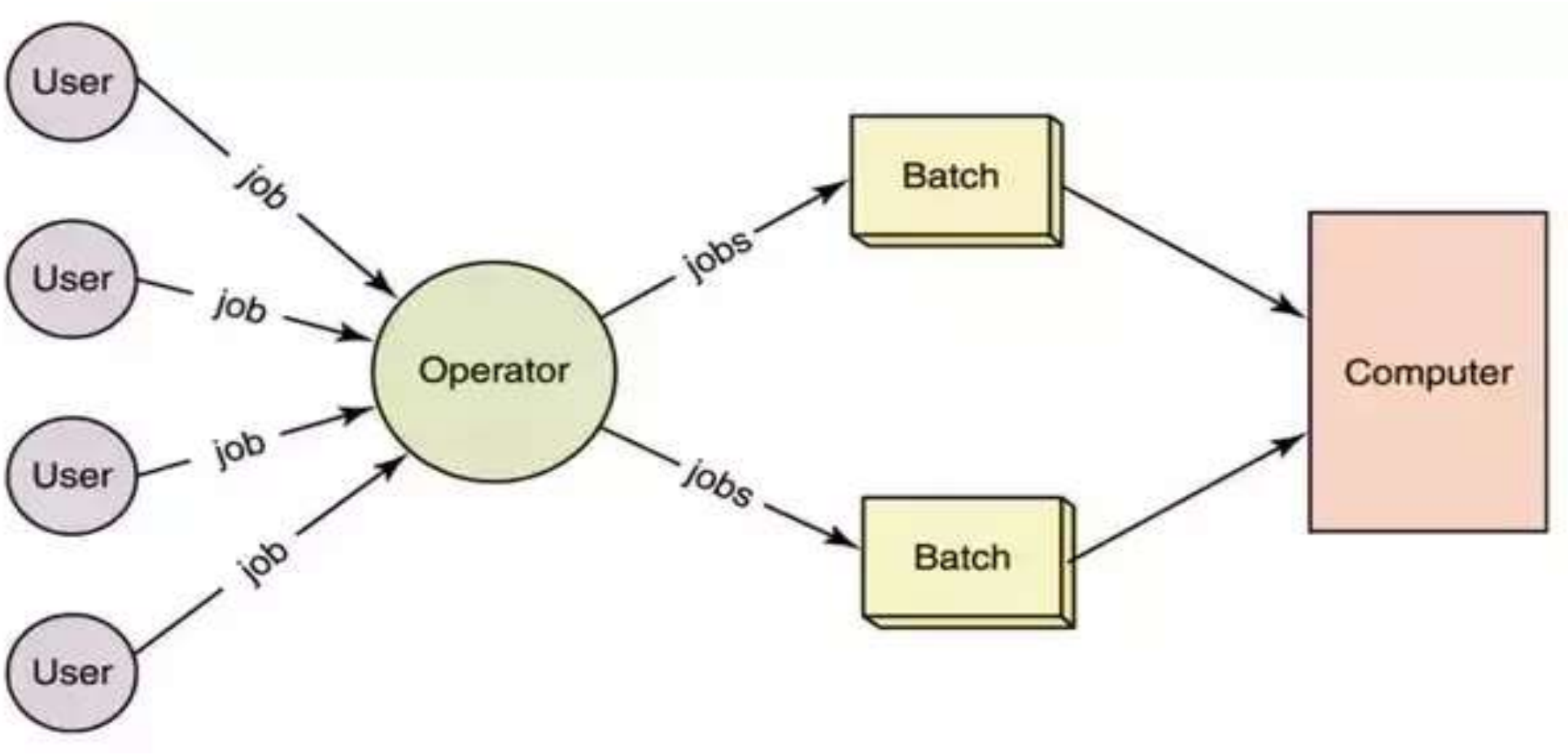
Serial Processing

- **Serial Processing**: The **Serial Processing Operating Systems** are those which Performs all the instructions into a Sequence Manner or the Instructions those are given by the user will be executed by using the FIFO Manner means First in First Out.

Batch Processing

- The users of a batch operating system do not interact with the computer directly.
- Each user prepares his job on an off-line device like punch cards and submits it to the computer operator.
- To speed up processing, jobs with similar needs are batched together and run as a group.
- The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

Batch Processing

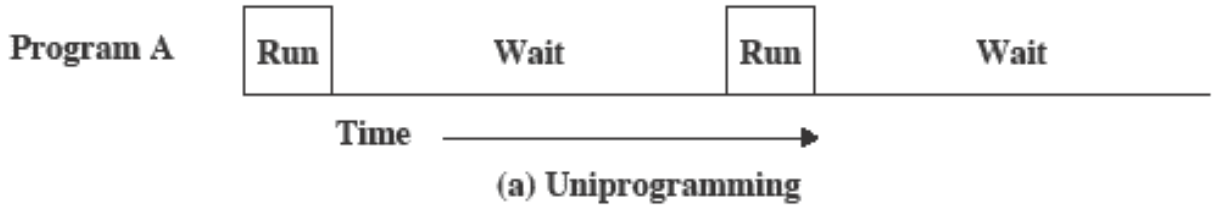


Batch Processing .. contd

The problems with Batch Systems are as follows −

- Lack of interaction between the user and the job.
- CPU is often idle, because the speed of the mechanical I/O devices is slower than the CPU.
- Difficult to provide the desired priority.

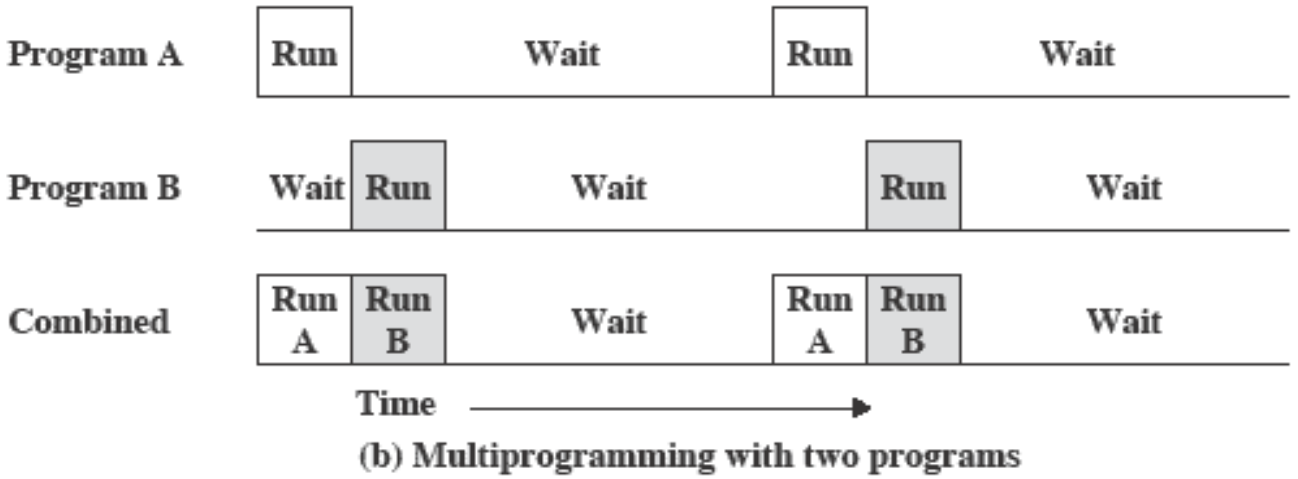
Uniprogramming



(a) Uniprogramming

- The processor spends a certain amount of time executing, until it reaches an I/O instruction; it must then wait until that I/O instruction concludes before proceeding

Multiprogramming



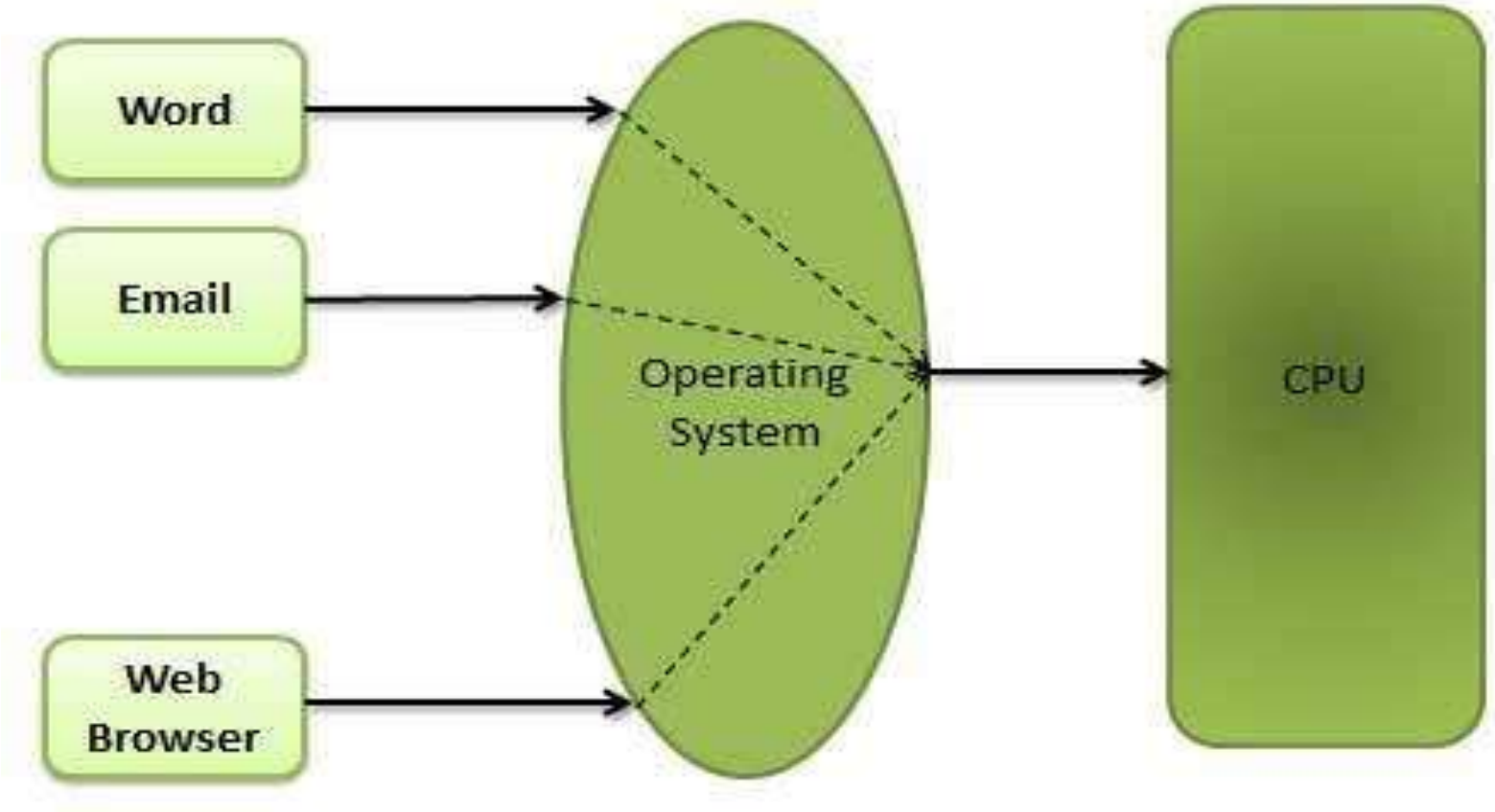
(b) Multiprogramming with two programs

- There must be enough memory to hold the OS (resident monitor) and one user program
- When one job needs to wait for I/O, the processor can switch to the other job, which is likely not waiting for I/O

Multiprogramming Operating Systems

- In multiprogramming system, when one program is waiting for I/O transfer; there is another program ready to utilize the CPU. So it is possible for several jobs to share the time of the CPU.
- But it is important to note that multiprogramming is not defined to be the execution of jobs at the same instance of time. Rather it does mean that there are a number of jobs available to the CPU (placed in main memory) and a portion of one is executed then a segment of another and so on

Multiprogramming Operating Systems



Multiprogrammed Batch Systems

Read one record from file	15 μs
Execute 100 instructions	1 μs
Write one record to file	15 μs
TOTAL	31 μs

Percent CPU Utilization $= \frac{1}{31} = 0.032 = 3.2\%$

Figure 2.4 System Utilization Example

- Processor is often idle
  - even with automatic job sequencing
  - I/O devices are slow compared to processor

Advantages of multiprogramming systems

- CPU is used most of time and never become idle
- The system looks fast as all the tasks runs in parallel
- Short time jobs are completed faster than long time jobs
- Multiprogramming systems support multiply users
- Resources are used nicely
- Total read time taken to execute program/job decreases
- Response time is shorter
- In some applications multiple tasks are running and multiprogramming systems better handle these type of applications

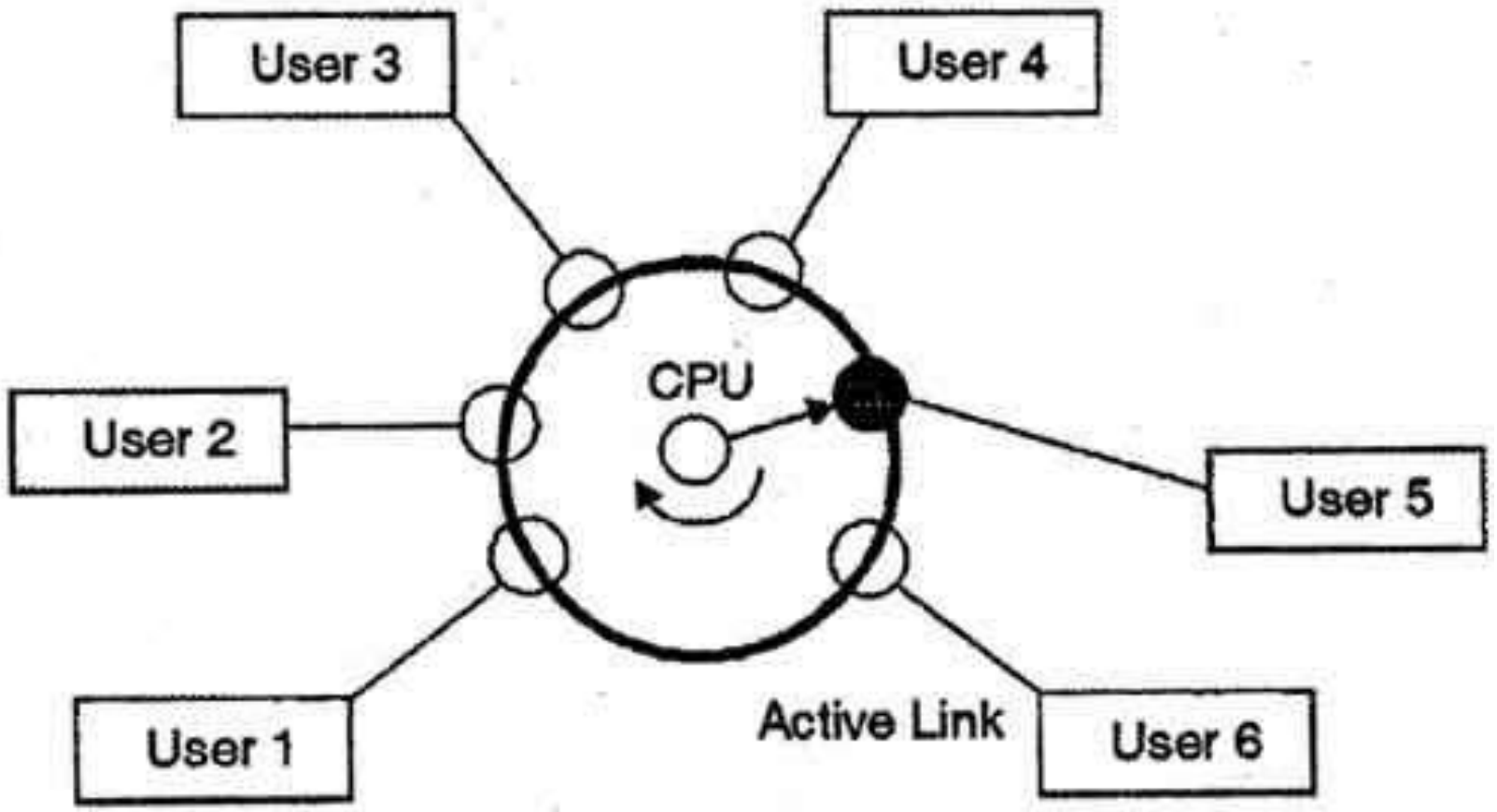
Disadvantages of multiprogramming systems

- It is difficult to program a system because of complicated schedule handling
- Tracking all tasks/processes is sometimes difficult to handle
- Due to high load of tasks, long time jobs have to wait long

Time-sharing operating systems

- Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time.
- Time-sharing or multitasking is a logical extension of multiprogramming.
- Processor's time which is shared among multiple users simultaneously is termed as time-sharing.
- Multiple jobs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response.
- For example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation. That is, if **n** users are present, then each user can get a time quantum. When the user submits the command, the response time is in few seconds at most.

Time-sharing operating systems



Difference between Multiprogramming and Time sharing

- The main difference between Multiprogrammed Batch Systems and Time-Sharing Systems is that in case of Multiprogrammed batch systems, the objective is to maximize processor use, whereas in Time-Sharing Systems, the objective is to minimize response time.

Advantages & Disadvantages of Time Sharing

Advantages −

- Provides the advantage of quick response.
- Avoids duplication of software.
- Reduces CPU idle time.

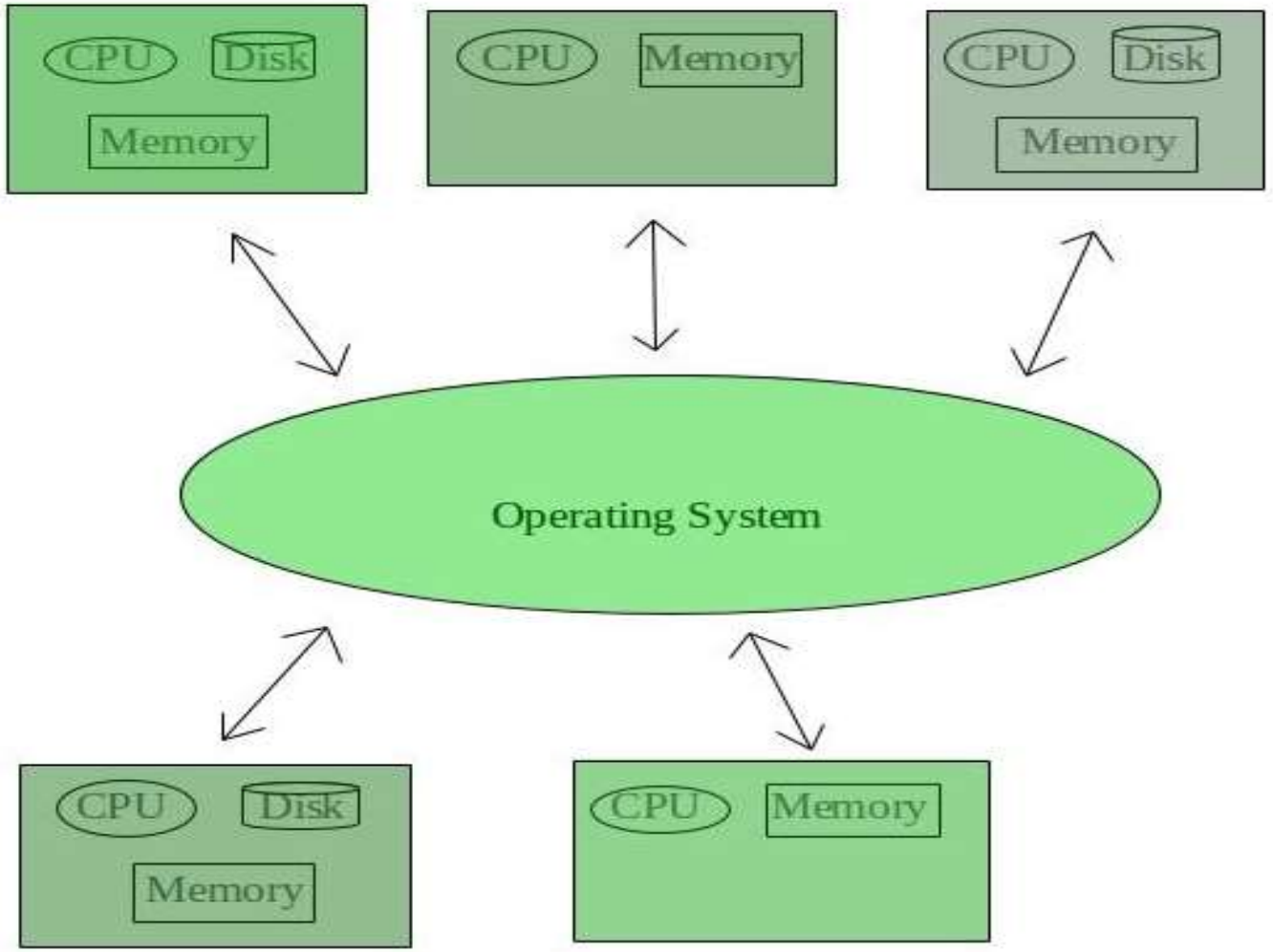
Disadvantages −

- Problem of reliability.
- Question of security and integrity of user programs and data.
- Problem of data communication.

Distributed operating System

- Distributed systems use multiple central processors to serve multiple real-time applications and multiple users.
- Data processing jobs are distributed among the processors accordingly.
- The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines).
- These are referred as **loosely coupled systems** or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on.

Distributed operating System



Advantages of Distributed operating System

- The advantages of distributed systems are as follows –
- With resource sharing facility, a user at one site may be able to use the resources available at another.
- Speedup the exchange of data with one another via electronic mail.
- If one site fails in a distributed system, the remaining sites can potentially continue operating.
- Better service to the customers.
- Reduction of the load on the host computer.
- Reduction of delays in data processing.

Parallel Systems

- Parallel Processing Systems are designed to speed up the execution of programs by dividing the program into multiple fragments and processing these fragments simultaneously.
- Such systems are multiprocessor systems also known as tightly coupled systems.
- Parallel systems deal with the simultaneous use of multiple computer resources that can include a single computer with multiple processors, a number of computers connected by a network to form a parallel processing cluster or a combination of both.

Parallel Systems

- Flynn's has classified the computers system based on parallelism in instructions and data stream

1.SISD – Single Instruction stream Single Data stream

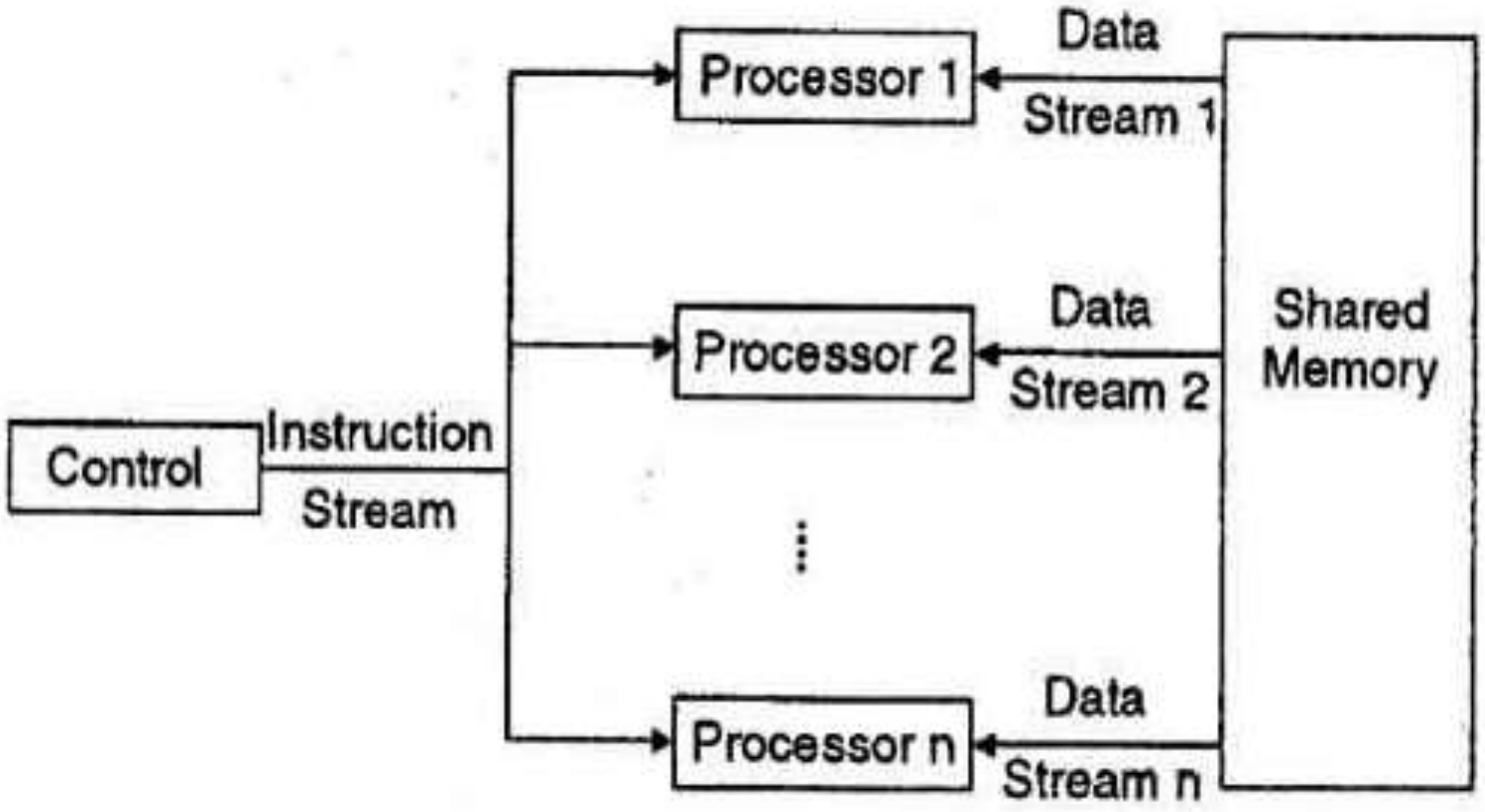
2.SIMD - Single Instruction stream Multiple Data stream

3.MISD - Multiple Instruction stream Single Data stream

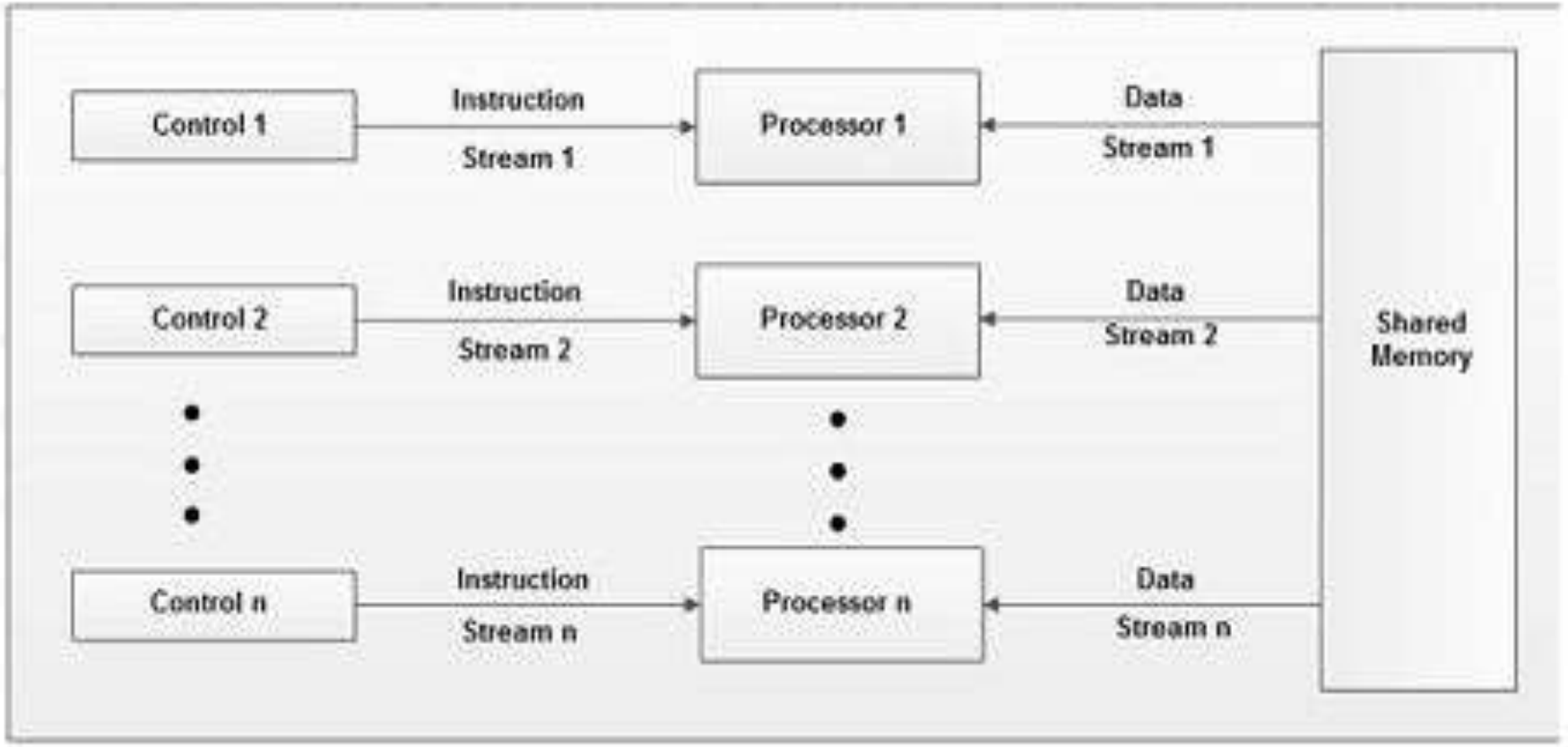
4.MIMD - Multiple Instruction stream Multiple Data stream

Out of above only two are relevant to parallel computers SIMD & MIMD

Parallel Systems - SISD



Parallel Systems - MIMD



Advantages of Parallel Systems

Advantages-

- Parallel computing saves time, allowing the execution of applications in a shorter wall-clock time.
- Solve Larger Problems in a short point of time.
- Compared to serial computing, parallel computing is much better suited for modeling, simulating and understanding complex, real-world phenomena.
- Throwing more resources at a task will shorten its time to completion, with potential cost savings. Parallel computers can be built from cheap, commodity components.
- Many problems are so large and/or complex that it is impractical or impossible to solve them on a single computer, especially given limited computer memory.
- You can do many things simultaneously by using multiple computing resources.

Disadvantages of Parallel Systems

Disadvantages-

- Programming to target Parallel architecture is a bit difficult but with proper understanding and practice you are good to go.
- Various code tweaking has to be performed for different target architectures for improved performance.
- Communication of results might be a problem in certain cases.
- Power consumption is huge by the multi core architectures

Real Time operating System

- A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. The time taken by the system to respond to an input and display of required updated information is termed as the **response time**. So in this method, the response time is very less as compared to online processing.
- Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application. A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail. For example, Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

Types of real-time operating systems

- Hard real-time systems

Hard real-time systems guarantee that critical tasks complete on time. In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM. In these systems, virtual memory is almost never found.

Soft real-time systems

- Soft real-time systems are less restrictive. A critical real-time task gets priority over other tasks and retains the priority until it completes. Soft real-time systems have limited utility than hard real-time systems. For example, multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc.

Advantages of Real Time OS

Advantages -

- Maximum use of devices and system thus gives more output from all the resources
- Time given for shifting tasks is very less
- It Focusses on running applications and gives less importance to queue applications
- Size of programs are small
- Error free
- Memory allocation is well managed

Disadvantages of Real Time OS

Disadvantages-

- Only some task run at the same time
- Sometimes the system resources are not good enough and they are costly as well
- Complex and difficult to write algorithms are used
- It requires specific device drivers
- They are very less prone to switching tasks

Network operating System

- A Network Operating System runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions. The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.

Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

Advantages & Disadvantages of Network operating System

Advantages –

- Centralized servers are highly stable.
- Security is server managed.
- Upgrades to new technologies and hardware can be easily integrated into the system.
- Remote access to servers is possible from different locations and types of systems.

Disadvantages –

- High cost of buying and running a server.
- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

References

- William Stallings, “Operating Systems – internals and design principles”, Prentice Hall, 7thEdition, 2011.(Ch 1 -9,11,12).
- William Stallings “Operating Systems – Internals and design principles”, Pearson Education, 5th Edition
- https://link.springer.com/chapter/10.1007/978-3-662-04303-5_5
- http://ecomputernotes.com/fundamental/disk-operating-system/parallel-processing-systems
- https://www.geekboots.com/story/parallel-computing-and-its-advantage-and-disadvantage
- https://www.includehelp.com/operating-systems/advantages-and-disadvantages-of-different-types-of-operating-systems.aspx
- https://www.tutorialspoint.com/operating_system/os_types.htm
- http://ecomputernotes.com/fundamental/disk-operating-system/parallel-processing-systems