

E-Content Module

Subject	: Operating System
Topic	: Classification of Operating System
Target Learners	: BCA/ MCA
Duration	: 25-30 min

About Author:

Name	: Devi Dayal Sinha
Email-id	: devidayal_ap@impactcollege.in

Classification of Operating System

Introduction

An Operating System (OS) is a collection of controlled programs which when executed produces an interface for the user's interaction with the system and its components. It manages hardware resources such as CPU, memory, storage, and input/output devices, and provides services to application programs.

Operating systems can be classified based on processing method, number of users, task handling, and system architecture.

The classification has been done majorly in three categories which are:

1. On the basis of Evolution
2. On the basis of Tasking
3. On the basis of Advancement

1. On the basis of Evolution

- i. Serial Processing System
- ii. Batch Operating System
- iii. Multiprogramming

2. On the basis of Tasking

- i. Single User Single Tasking
- ii. Single User Multi Tasking
- iii. Multi User Single Tasking
- iv. Multi User Multi Tasking

3. On the basis of Advancement

- i. Multiprogramming
- ii. Time Sharing Operating System
- iii. Real Time Operating System
- iv. Network Operating System(NOS)
- v. Distributed Operating System(DOS)
- vi. Client Server Operating System

1.i. Serial Processing System

Definition:

A serial processing system allows one only task to execute at one time and the next task can only initiate its execution when the current has been completely executed and terminated.

Characteristics:

- Only one job is allowed to enter
- Input, Process and Output cycle is completed then only next job can enter.
- Execution is sequential

Advantages:

- Earliest form of operating system
- No overlapping among tasks

Disadvantages:

- Time consuming
- Sequential in nature
- Multiple jobs not allowed

Examples:

- MS-DOS

1.ii. Batch Operating System

Definition:

A Batch Operating System groups similar jobs (programs) into batches and executes them one by one without user interaction.

Characteristics:

- Jobs are collected and processed in batches
- No direct interaction between user and system
- Execution is sequential
- High turnaround time

Advantages:

- Efficient use of CPU
- Suitable for large repetitive jobs

Disadvantages:

- No user interaction
- Difficult to debug programs

Examples:

- IBM Mainframe OS
- Early UNIX systems

1.iii. Multiprogramming Operating System

Definition:

A Multiprogramming OS allows multiple programs to reside in memory at the same time, and the CPU switches between them to improve utilization.

Characteristics:

- Multiple jobs in memory
- CPU always kept busy
- Improves system efficiency

Advantages:

- Better CPU utilization
- Reduced idle time

Disadvantages:

- Complex memory management
- Requires large main memory

Examples:

- UNIX
- Linux

2.i. Single User Single Tasking

Definition: As like serial processing system allowing one only task to execute at one time.

Characteristics:

- Same as Serial Processing System
- Restricting no. of users up to 1 only
- Max no. of task remains 1 only

Advantages/ Disadvantages:

- Same as Serial Processing system

2.ii. Single User Multi Tasking

Definition: As like serial processing system allowing one only task to execute at one time.

Characteristics:

- Same as Batch Operating System
- Restricting no. of users up to 1 only
- Max no. of tasks is not restricted

Advantages/ Disadvantages:

- Same as Batch operating system

2.iii. Multi User Single Tasking

Definition: MUST use the concept of more number of users working on a single platform where the no of task is still limited up to 1.

Characteristics:

- More no of users can use the same platform by means of common line of connectivity.
- The interface for all the users is the same i.e restricted up to 1 only.

Advantages

- More than one user can use the system
- Sharing of job or task can be implemented

Disadvantages:

- Despite more no. of users using the common line of connectivity, can't use multiple applications on different environment.

2.iv. Multi-User Multi Tasking

Definition:

Allows multiple users to access the system at the same time.

Characteristics:

- Time sharing
- Resource management
- Security mechanisms

Examples:

- UNIX
- Linux

3.i. Multiprogramming:

Same as 1.iii

3.ii. Time Sharing Operating System

Definition:

A Time Sharing Operating System allows multiple users to interact with the system simultaneously by sharing CPU time.

Characteristics:

- Each user gets a small time slice
- Fast response time
- Interactive system

Advantages:

- Multiple users can work at the same time
- Quick response

Disadvantages:

- High overhead
- Security issues

Examples:

- UNIX
- Linux
- Windows Server

3.iii. Real-Time Operating System (RTOS)

Definition:

A Real-Time Operating System responds to input immediately and processes data within a fixed time constraint.

Types:

1. Hard Real-Time OS – Missing a deadline may cause system failure
2. Soft Real-Time OS – Missing deadlines degrades performance but does not cause failure

Characteristics:

- Deterministic response
- Time-critical operations

Advantages:

- Reliable and predictable
- Used in critical systems

Disadvantages:

- Expensive
- Limited user interface

Examples:

- VxWorks
- RTLinux
- QNX

3.iv. Network Operating System

Definition:

A Network OS provides services to computers connected over a network.

Characteristics:

- Centralized server
- User authentication
- File and printer sharing

Advantages:

- Centralized control
- Easy resource sharing

Disadvantages:

- Server dependency
- Requires network infrastructure

Examples:

- Windows Server
- Novell NetWare

3.v. Distributed Operating System

Definition:

A Distributed OS manages a group of independent computers and makes them appear as a single system.

Characteristics:

- Resource sharing
- Load balancing
- Fault tolerance

Advantages:

- High reliability
- Better performance

Disadvantages:

- Complex design
- Network dependency

Examples:

- Amoeba
- LOCUS

3.vi. Client Server

Definition:

A client-server operating system refers to two distinct but interacting OS types: Client OS, for user devices (PCs, phones) to request services (Windows, macOS, Android), and

Server OS, designed to provide those services (Windows Server, Linux, Red Hat) to many clients simultaneously, managing resources centrally for a network. The client OS runs locally for single-user tasks, while the server OS handles complex, multi-user requests, forming the backbone of network interactions like web browsing, email, and file sharing

Characteristics:

- Resource sharing
- Interacting operating systems
- Connection enhancement

Advantages:

- High reliability, still subjected to realization for accuracy
- Better performance

Disadvantages:

- Complex design
- Network dependency and poor connectivity

Examples:

- Windows Server, Linux (Ubuntu Server, Red Hat Enterprise Linux)
-

Summary Table

Type of OS	Key Feature	Example
Batch OS	Batch processing	IBM OS
Multiprogramming OS	Multiple programs in memory	UNIX
Time Sharing OS	Multiple users	Linux
Real-Time OS	Time critical	VxWorks
Distributed OS	Multiple systems	Amoeba
Network OS	Network services	Windows Server

Conclusion

Operating systems are classified based on how they process tasks, support users, and manage resources. Understanding these classifications helps in selecting the appropriate OS for specific applications such as business processing, real-time control, or network management.

Classification of Operating System

Exam Questions & Answers (BCA – 2nd Semester)

PART – A

[Very Short Answer Questions]

1. What is an Operating System?

An Operating System is system software that manages computer hardware and provides services to application programs.

2. Define Batch Operating System.

A Batch Operating System executes jobs in batches without user interaction.

3. What is multiprogramming?

Multiprogramming allows multiple programs to reside in memory and share CPU time.

4. What is a Time Sharing Operating System?

It allows multiple users to use a computer system simultaneously by sharing CPU time.

5. Define Real-Time Operating System (RTOS).

An RTOS processes data and responds within a fixed time constraint.

6. What is a Distributed Operating System?

It manages multiple independent computers and makes them appear as a single system.

7. What is Network Operating System?

A Network OS provides services like file sharing and user management over a network.

8. Give an example of Single User Operating System.

MS-DOS.

9. Give an example of Multi-User Operating System.

UNIX.

10. What is a Hard Real-Time System?

A system where missing a deadline may cause complete system failure.

PART – B

[Short Answer Questions]

1. Explain Batch Operating System.

A Batch Operating System groups similar jobs into batches and executes them sequentially. There is no interaction between the user and the system during execution. It is mainly used in large organizations for payroll and billing systems.

2. Write advantages and disadvantages of Time Sharing OS.

Advantages:

- Quick response time
- Supports multiple users

Disadvantages:

- High system overhead
- Security issues

3. Differentiate between Single User and Multi-User Operating System.

Single User OS

Multi-User OS

One user at a time

Multiple users at a time

Simple design

Complex design

Example: MS-DOS

Example: UNIX

4. Explain Hard and Soft Real-Time Operating System.

- Hard RTOS: Deadlines must be met strictly (e.g., missile systems).

- Soft RTOS: Deadlines can be missed occasionally (e.g., multimedia systems).

5. What are the features of Distributed Operating System?

- Resource sharing
- Load balancing
- Fault tolerance
- Transparency

PART – C

(Long Answer Questions)

1. Explain the classification of Operating System.

An Operating System can be classified into different types based on functionality:

- Batch Operating System: Executes jobs in batches.
- Multiprogramming OS: Multiple programs reside in memory.
- Time Sharing OS: Multiple users share CPU time.
- Real-Time OS: Used in time-critical systems.
- Distributed OS: Multiple systems work as one.
- Network OS: Provides network-based services.
- Single User OS: Supports one user.
- Multi-User OS: Supports multiple users simultaneously.

Each type is designed for specific applications and environments.

2. Explain Real-Time Operating System with its types.

A Real-Time Operating System responds to inputs within a fixed time limit. It is used in systems where timing is critical.

Types:

1. Hard Real-Time OS: Missing a deadline leads to failure.
2. Soft Real-Time OS: Missing a deadline reduces performance but does not fail the system.

Examples: VxWorks, QNX, RTLinux

3. Explain Network Operating System with advantages and disadvantages.

A Network Operating System manages network resources and allows communication between computers.

Advantages:

- Centralized data management
- Resource sharing

Disadvantages:

- Server dependency
- Costly setup

Examples: Windows Server, Novell NetWare

4. Compare Batch OS and Time Sharing OS.

Batch OS

Time Sharing OS

No user interaction

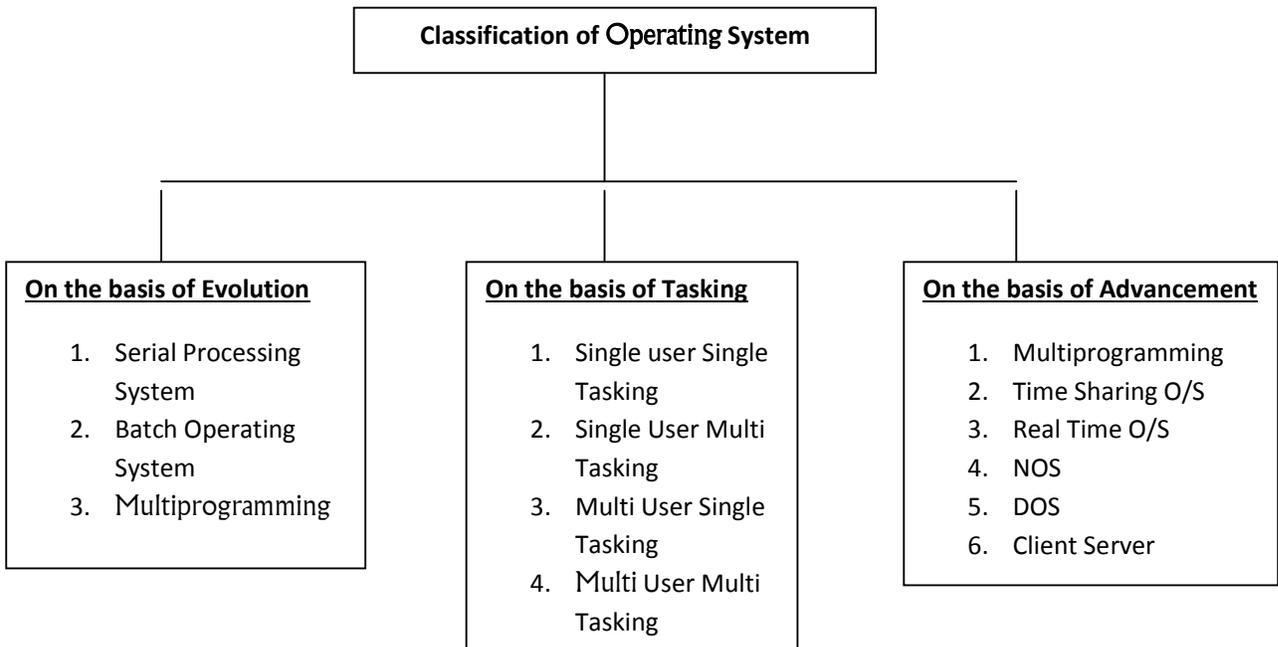
Interactive

Long response time

Short response time

Batch OS
Sequential execution
Example: IBM OS

Time Sharing OS
CPU time shared
Example: UNIX



Authors Note:

- All the information's provided above are for reference only
- Contents have been taken from several resources based on requirement and usability
- Readers suggestion and feedback are welcome
- For reference:

Operating System Concepts by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne (often called the "Dinosaur Book").

Operating Systems: Three Easy Pieces by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau (available free online).

Operating Systems by William Stallings (Internals and Design Principles).

Modern Operating Systems by Andrew S. Tanenbaum and Herbert Bos.

Operating Systems Design and Implementation by Andrew S. Tanenbaum and Albert S. Woodhull.

The Linux Programming Interface by Michael Kerrisk.

Linux Kernel Development by Robert Love.

Advanced Programming in the UNIX Environment by W. Richard Stevens.

The Art of UNIX Programming by Eric S. Raymond.