Operating System (OS)

Complete Revision & Quick Reference Notes

1. Operating System Kya Hai?

Operating System (OS) ek system software hai jo:

- User aur Computer Hardware ke beech **interface** ka kaam karta hai.

Examples:

Windows | Linux | macOS | Android | Unix

> ⭐ **STET Point:**
> **OS = Resource Manager + Interface between User and Hardware**

2. Functions of Operating System

💡 **Functions ki Trick: P M F I S E R**

P → Process Management
M → Memory Management
F → File Management
I → I/O Management
S → Security
E → Error Detection
R → Resource Allocation

3. Structures of Operating System ⭐⭐

OS structure ka matlab: *Operating System ke different components ko kis tarah organize kiya gaya hai.*

① Monolithic Structure

Saare major OS services ek large kernel mein hoti hain.

② Layered Structure

- OS ko different layers mein divide kiya jata hai.
- Har layer generally lower layer ki services use karti hai.

③ Microkernel

- Kernel ko minimum essential functions tak rakha jata hai.
- Baaki services user space mein run kar sakti hain.

④ Modular Structure

OS ko different modules mein divide kiya jata hai.

⑤ Hybrid Structure

Different OS structures ka combination.

4. Types of Operating System

- **Batch Operating System**
  - Jobs ko batches mein collect karke execute kiya jata hai.
  - User ka execution ke time direct interaction nahi hota.

- **Multiprogramming OS**
  - Ek se zyada programs memory mein simultaneously rakhe ja sakte hain.
  - CPU ek program ke wait karne par doosre program ko execute kar sakta hai.

⭐ **Goal → CPU utilization increase**

- **Multitasking OS**

User ko ek saath multiple tasks/programs run karne deta hai.

Example: Chrome + MS Word + Music

⭐ **Keyword → Multiple tasks**

- **Multiuser OS**

Multiple users same computer/system resources ko use kar sakte hain.

⭐ **Keyword → Multiple users**

- **Multiprocessing OS**

System mein multiple CPUs/processors hote hain.

⭐ **Keyword → Multiple processors**

- **Real-Time Operating System (RTOS) ⭐⭐⭐**

Tasks ko specified time/deadline ke andar complete karna important hota hai.

- **Distributed Operating System**

Multiple independent computers ko coordinate karke ek single system jaisa appear karwa sakta hai.

- **Network Operating System**

Network par connected computers aur resources ko manage karta hai.

Examples: File sharing, Printer sharing, User management

⭐ **Keyword → Network resources**

STET EXAM STUDY NOTES

Operating System: CPU Scheduling & Process

Quick Revision & Concept Summary for Computer Science

1. Process (प्रॉसेस)

Definition: Process = A program in execution.

> 👉 **MS Word Example:**
> 1. Jab aap MS Word ko open karte ho → Program RAM mein load hota hai.
> 2. CPU usko execute karta hai.
> 3. Ab woh **Process** ban gaya.

2. Process State (प्रॉसेस स्टेट)

Process State: Process abhi kis condition mein hai.

⭐ **STET Ke Liye Ye Table Yaad Karo:**

State	Process kya kar raha hai?
New	Create ho raha hai
Ready	CPU ka wait kar raha hai
Running	CPU par execute ho raha hai
Waiting / Blocked	I/O ya Event ka wait kar raha hai
Terminated	Execution complete ho gaya

3. CPU Scheduling & Scheduler

Jab multiple processes READY queue mein CPU ka wait kar rahe hote hain, OS decide karta hai:

👉 **CPU kisko milega aur kitne time ke liye?**

Isi process ko **CPU Scheduling** kehte hain.

Formula Definition: *CPU Scheduling = Selecting a process from the ready queue and allocating CPU to it.*

Scheduler: Scheduler woh OS component hai jo decide karta hai ki next CPU kis process ko milega.

4. Types of Scheduling ⭐⭐⭐

A. Non-Preemptive Scheduling

Ek process ko CPU mil gaya, to generally woh CPU tab tak nahi chhodega jab tak execution complete/block na ho.

Examples: FCFS, SJF

```
P1 → CPU
     ↓
  Complete
     ↓
P2 → CPU
```

B. Preemptive Scheduling

OS running process se CPU wapas le sakta hai aur doosre process ko de sakta hai.

Examples: Round Robin, SRTF

```
P1 → CPU
     ↓
  Preempt
     ↓
P2 → CPU
```

> 🧠 **Trick to Remember**
>
> **Preemptive** = CPU forcibly taken back
> **Non-preemptive** = CPU normally not taken back

5. Scheduling Algorithms Trick

Algorithms ko ek trick se yaad karo:

F S S P R M M

First Come First Served (FCFS)
Shortest Job First (SJF)
Shortest Remaining Time First (SRTF)
Priority Scheduling
Round Robin (RR)
Multilevel Queue Scheduling
Multilevel Feedback Queue Scheduling

6. Important Scheduling Terms & Formulas

- **Arrival Time (AT):** Process ready queue mein kab aaya.
- **Burst Time (BT):** Process ko CPU par kitna time chahiye (Execution time).
- **Completion Time (CT):** Process kab complete hua.
- **Turnaround Time (TAT):** Total time process system mein raha.
- **Waiting Time (WT):** Process ne ready queue mein kitna wait kiya.

⭐ **Important Formulas** ⭐

TAT = Completion Time – Arrival Time (TAT = CT – AT)

WT = Turnaround Time – Burst Time (WT = TAT – BT)

STET EXAM REVISION NOTES

Process Synchronization & Semaphores

Operating System Quick Revision & Key Concepts

1. Process Synchronization (प्रॉसेस सिंक्रोनाइज़ेशन)

Definition: Jab multiple processes/threads same shared resource or shared data ko access karte hain, tab unko properly coordinate karna zaroori hota hai. Is coordination ko **Process Synchronization** kehte hain.

2. Problem: Race Condition (रेस कंडीशन)

Jab multiple processes bina synchronization ke shared data ko modify karti hain, to data inconsistency ho sakti hai.

> **Maan lo Example:** Initial `Counter = 10`. P1 aur P2 dono counter ko +1 karna chahte hain.
> • **Expected Result:** 12
> • **Actual Problem:** Dono same time par old value 10 read kar sakte hain → **Actual Result:** 11 ❌ (Wrong Result)

> 🧠 **बस याद रखो:**
> **Race Condition** = Multiple processes access shared data at the same time → **Wrong** / **Unexpected Result**.

3. Critical Section (क्रिटिकल सेक्शन)

Definition: Program ka woh part jahan process shared data/resource ko access ya modify karta hai, use **Critical Section** kehte hain.

> ⭐ **STET Rule: Critical Section = Shared Resource / Data Access wala code**

🖨️ **Printer Example:** Computer mein 1 printer (Shared Resource) hai. Do processes hain: P1 (Print A), P2 (Print B).

```
P1 Workflow: Start → Prepare document → Send document to Printer (🔴 Critical Section) → Finish
```

Agar P1 printer use kar raha hai (P1 → 🖨️ `Printing`), to P2 wait karega (P2 → ⏳ `Wait`).

4. Mutual Exclusion (म्यूचुअल एक्सक्लूजन)

Definition: Mutual Exclusion means only one process can execute in the critical section at a time.

Agar P1 Critical Section mein hai, to P2 ko usi time Critical Section mein enter nahi karne denge.

> ⭐ **STET Trick**
> **Mutual Exclusion = Only ONE process at a time**

5. Critical Section vs Mutual Exclusion

Term	Meaning & Example
Critical Section	Shared resource use karne wala code (e.g., 🖨️ Printer use karne wala code)
Mutual Exclusion	Ek time par sirf one process ko allow karna (e.g., 👤 Ek time par ek hi process printer use kare)

6. Semaphore (सेमाफोर) & STET One-Liner Notes

Definition: Semaphore ek synchronization tool/mechanism hai jo processes ko shared resources/critical section access karne mein control karta hai.

Topic	One-line Meaning	Simple Example
1. Binary Semaphore	Semaphore with only 0 or 1 value.	🖨️ 1 printer → 1 = available, 0 = busy
2. Counting Semaphore	Semaphore used to control multiple identical resources.	🖨️🖨️🖨️ 3 printers → Semaphore = 3
3. wait()	Acquires/takes a resource; value decreases.	P1 takes printer → 3 → 2
4. signal()	Releases a resource; value increases.	P1 finishes printing → 2 → 3

> 🧠 **Super Easy Trick**
>
> • `wait()` = लेना 📥 | `signal()` = छोड़ना 📤 | **Binary** = 0/1 | **Counting** = Multiple resources

Deadlock in Operating Systems

Complete Revision Notes & STET Exam Preparation Guide

STET HIGH PRIORITY ⭐⭐⭐⭐⭐

1. What is Deadlock? (डेडलॉक क्या है?)

Deadlock is a situation where two or more processes are unable to proceed because each is waiting for a resource held by the other, resulting in an indefinite wait.

> 💬 ***Simple Explanation:***
> *"Tumhare paas meri cheez hai, mere paas tumhari cheez hai, aur dono ek-doosre ka wait kar rahe hain."*

🔒 **Visual Representation of Deadlock:**

⬇ **DEADLOCK OCCURS** 🔒

2. Deadlock ke 4 Necessary Conditions⭐⭐⭐⭐⭐

Deadlock tabhi possible hai jab ye 4 conditions **simultaneously (ek sath)** present hon (Coffman Conditions):

① Mutual Exclusion

Resource ek time par sirf **one process** hi use kar sakta hai.
Example: Printer → Agaram P1 use kar raha hai to P2 ko wait karna padega.

② Hold and Wait

Process ke paas ek resource already **held** hai, aur woh doosre resource ka **wait** kar raha hai.
State: P1 → R1 held + R2 waiting.

③ No Preemption

Resource ko process se **forcefully nahi liya ja sakta**. Process execution finish hone ke baad khud release karega.

④ Circular Wait

Processes ek **circular chain** mein ek-doosre ke resource ka wait kar rahe hain.
Chain: P1 waits for P2 & P2 waits for P1.

🧠 **STET Memory Trick**

M H N C

Mutual Exclusion | **H**old and Wait | **N**o Preemption | **C**ircular Wait

3. Deadlock Handling Approaches⭐⭐⭐⭐⭐

Operating System deadlock ko handle karne ke liye main 4 approaches use karta hai:

Approach	Key Mechanism / Strategy	Important Note
1. Prevention	Deadlock ki 4 necessary conditions mein se **at least one condition ko eliminate** / **prevent** karo (e.g., prevent Circular Wait).	Ensures zero deadlocks
2. Avoidance	Resource allocate karne se pehle check karo: *"Agar resource diya, to system safe state mein rahega?"* • Safe → Allocate • Unsafe → Wait	⭐ **Banker's Algorithm** is used for Avoidance
3. Detection	OS deadlock ko hone deta hai, fir periodic cycles par check karta hai: *"Deadlock hua hai ya nahi?"*	Uses Resource Allocation Graph (RAG)
4. Recovery	Deadlock detect hone ke baad usse remove/ recover karna. **Methods:** Process termination ya Resource preemption.	Restores System State

4. Quick One-Line Revision Checklist ⚡

Deadlock: Processes permanently waiting for resources held by each other.

4 Conditions: M-H-N-C (Mutual Exclusion, Hold & Wait, No Preemption, Circular Wait).

Prevention: Break at least one of the 4 conditions.

Avoidance: Avoid unsafe state before allocation (Uses **Banker's Algorithm**).

Detection: Allow deadlocks to occur, then periodically find them.

Recovery: Process termination or resource preemption to break the loop.

STET / CS EXAM SPECIAL NOTES

Operating System: File Systems & Allocation Methods

Complete Revision Guide • Concepts, Visual Examples & Key Tricks

1. Topic Overview & Key Concepts

TOPIC	SIMPLE MEANING	STET KEY POINT	EXAMPLE
File System	Files ko store & manage karna	OS manages files	`.pdf`, `.jpg`
File Organization	Files ko arrange karna	Files → Directories/Folders	`Documents → notes.pdf`
File Allocation	File ke blocks ko disk par place karna	3 main methods	Contiguous, Linked, Indexed
Contiguous Allocation	Blocks ek saath/consecutive	**Continuous blocks**	`10 → 11 → 12 → 13`
Linked Allocation	Blocks alag-alag jagah + links	**Pointer / Link**	`10 → 25 → 7`
Indexed Allocation	Index block addresses store karta hai	**Index block**	`Index → 10, 25, 40`
Directory	Files ko organize karta hai	Folder jaisa	`Documents/`
Single-Level Directory	Ek hi directory	Simple structure	`A.txt, B.txt`
Two-Level Directory	Har user ki separate directory	User-based	`User1 → Files`
Tree Directory	Hierarchical structure	**Root → Folder → File**	`C:\Users\...\file`
File I/O	File se data lena/dena	**Input / Output**	Read, Write
Read	File se data lena	File → Memory	`read()`
Write	File mein data save karna	Memory → File	`write()`
Open	File access ke liye open	`open()` system call	Open PDF
Close	File access finish	`close()` system call	Close file

2. Deep Dive: File Allocation Methods

Maan lo ek file ko disk par **4 blocks** chahiye. Disk layout:

```
Disk Blocks: [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]
```

1️⃣ Contiguous Allocation — "Saath-Saath" 📦 Continuous Blocks

File ke saare blocks disk par ek ke baad ek (consecutive) rakhe jaate hain.

```
[1] [2] [3] [4] [5] [6] [7] [8] [9] [10]
        └──────── File A ────────┘
```

Example: File A ko blocks 3, 4, 5, 6 mil gaye (`3 → 4 → 5 → 6`).

🧠 **Memory Key:** Contiguous = Continuous = Saath-saath

2️⃣ Linked Allocation — "Jahan Jagah Mile" 🔗 Pointer Chain

Agar continuous 4 blocks available na ho, toh file ke blocks alag-alag jagah rakhe ja sakte hain. Har block ke paas next block ka link/pointer hota hai.

```
[1] [2] [3] [4] [5] [6] [7] [8] [9] [10]
Chain: Block 2 → Next=4 | Block 4 → Next=7 | Block 7 → Next=9 | Block 9 → END
```

Example Path: `[2] → [4] → [7] → [9]`

🧠 **Memory Key:** Linked = Chain 🔗

3️⃣ Indexed Allocation — "Ek Index List Rakho" 📋 Index Block Pointer

Isme ek special **Index Block** hota hai jiske andar file ke saare scattered blocks ke addresses store hote hain. Blocks ko aapas mein link karne ki zaroorat nahi hoti.

INDEX BLOCK
Block Address: 2
Block Address: 5
Block Address: 7
Block Address: 9

↓ Pointing to disk blocks: [2], [5], [7], [9]

🧠 **Memory Key:** Indexed = List / Index batayegi blocks kahan hain.

💡 **STET Quick Revision Trick**

- 📦 **Contiguous** = Continuous / Together (Fast, but External Fragmentation)
- 🔗 **Linked** = Chain of Pointers (No External Fragmentation, Slow Direct Access)
- 📋 **Indexed** = List / Index (Supports Direct Access, Pointer Overhead)

3. Fast Revision: Exam Comparison Table

ALLOCATION METHOD	MAIN ADVANTAGE	MAJOR DISADVANTAGE	ACCESS TYPE SUPPORTED
Contiguous	Super fast read/write performance	External Fragmentation (file growth difficult)	Sequential & Direct
Linked	Zero External Fragmentation	Slow Random Access (traversal required)	Only Sequential
Indexed	Direct Access + Flexible placement	Index Block Overhead for small files	Sequential & Direct

Operating System Concepts

Protection vs Security vs Recovery — Quick Revision Guide

1️⃣ Protection ⭐⭐⭐⭐⭐

> **Meaning:** System ke resources ko unauthorized access ya misuse se protect karna.

SIMPLE EXAMPLE

Tumhare laptop mein:

```
📁 Personal Folder
    ↓
👤 User A → Access ✅
👤 User B → Access ❌
```

OS decide karta hai: Kaun kya access kar sakta hai?

2️⃣ Security ⭐⭐⭐⭐⭐

> **Meaning:** System aur data ko unauthorized users, attacks aur threats se protect karna.

EXAMPLE FLOW

```
👤 User → 🔑 Password → System
```

Security mein commonly included tools/methods:

- Password
- Authentication
- Authorization
- Encryption
- Malware protection

Authentication vs Authorization ⭐

Authentication	Authorization
"Tum kaun ho?"	*"Tumhe kya karne ki permission hai?"*
Example: Username + Password	**Example:** Admin → Delete ✅ Normal User → Delete ❌

🧠 **STET Key Takeaway: Security = System/Data ko threats se protect karna**

3️⃣ Recovery ⭐⭐⭐⭐

> **Meaning:** System failure/problem ke baad system ko normal state mein wapas lana.

EXAMPLE SCENARIO

```
💻 System → ❌ Power Failure → Data/System problem
```

Recovery Methods:

- Backup
- Checkpoint
- Restore

RESTORE EXAMPLE

Tumne important file ka backup Google Drive/another disk mein rakha.

Original file: ❌ **Lost** | Backup: ✅ **Restore**

🔥 Protection vs Security vs Recovery

Topic	Simple Meaning	Example
Protection	Kaun resource access karega?	Read/Write permission
Security	System ko threats/attacks se bachana	Password, encryption
Recovery	Failure ke baad system ko restore karna	Backup/Restore

🧠 Super Trick

🔐 **Protection** = Permission
🛡️ **Security** = Threat se bachao
🔄 **Recovery** = Failure ke baad Restore