

## **Operating System Complete Concepts - Sequence Wise**

### **1. Introduction to Operating System**

**What is Operating System?** Operating system ka matlab hai hardware aur applications ke beech mein ek interface hota hai jo operating system hai. Simple si baat hai - hardware jaise RAM ho gaya, memory ho gayi, screen hai aapki, keyboard - ye saari saari cheezein aapke hardware hain. Aur saath mein application aapke kya honge jo aap chalate ho.

#### **Operating System ka Role:**

- Memory manage karna
- Process ke beech mein scheduling karna
- Disk management hota hai
- Sab kuch memory mein leke aana
- Process management karna

#### **Examples of Operating Systems:**

- Android/iOS (phone ke andar operating system)
- Windows/macOS/Linux (desktop operating systems)
- Real-Time OS (military applications, IoT devices)

### **2. Process, Thread, Program Concepts**

**Program:** Program jaise aap log Java ke andar koi program likhte ho uske baad aap use compile kar do aur then run jab kar rahe hote ho us program ko. Program basically set of instructions hote hain jinko follow karta hai computer to do a specific task.

**Process:** Process is an instance of a program in execution. Jab program chal raha hota hai na to usko kehte hain ki program is under execution to us time pe vo processes kahlate hain. Ek program can consist of multiple processes.

**PCB (Program Control Block):** Har process ka apna-apna khud ka PCB hota hai. Ek chota sa block hota hai jiske andar us process related saari ki saari information hote hai:

- Ye process kitni memory lega
- Iske corresponding registers (registers basically choti si memory hoti hai jo bahut zyada fast memory hoti hai)
- Iska program counter kya hai
- Iska khud ka stack hota hai

**Thread:** Thread bahut hi kamaal ka concept hai. Basically ek process ke andar multiple threads ho sakte hain. Thread alag-alag kaam kar rahe hote hain aur ek process ke andar jo agar 10 threads hain to vo 10 ke 10 threads jo hai vo us process ki memory ko share kar rahe hote hain.

**Multithreading Example:** Jaise video rendering kar rahe hote ho to video rendering ke time pe hota kya hai ki har ek frame ke andar bahut saare multiplication operations chal rahe hote hain. Har ek thread ko kuch-kuch multiplication ka kaam de diya jaata hai aur vo sab milke chal rahe hote hain.

**Inter Process Communication (IPC):** Do processes ko agar aapas mein communicate karna hota hai to use hum kehte hain IPC. Basically ek bus type ki memory hoti hai jiske through ye aapas mein communicate kar rahe hote hain.

### 3. Types of Operating System Environments

**Multiprocessing:** Ek aisa system jahan par multiple processes ek saath chal rahi hain ek system ke andar use kehte hain multiprocess system. Multiple processes chal rahi hoti hai ek saath simultaneous aur vo aapas mein bееch mein communicate bhi kar rahi hoti hain through IPC.

**Kitchen Example (Multiprocessing):** Jaise mummy khana bana rahi hoti hai kitchen ke andar to mummy ko aapne dekha hoga ki vo aata bhi goondh rahi hai, iske alawa vo sabzi bhi kaat rahi hai, iske alawa kuch ho sakta hai already ban bhi raha ho kitchen mein to multiple kaam kar rahi hoti hai. Vo vahan pe basically multiprocessing kaam hota hai.

**Multithreading:** Multithreading basically ek process ke andar multiple threads jab ek saath kaam kar rahe hote hain kisi environment mein to use hum kehte hain multi threaded environment.

**Multiprogramming:** Multiprogramming basically multiple programs ko ek saath memory mein load kar liya so that CPU kabhi khaali na baitha rahe. CPU ki utilization ko badhane ke liye humein multiprogramming karni hoti hai.

**Multitasking:** Multitasking is a technique that allows multiple tasks or processes to run concurrently on a single CPU. Vahi same cheez switching kar kar ke ek alag-alag process ke bееch mein switch kar karke jab vo bahut saare kaam kar raha tha use hum kehte hain multitasking.

### 4. Process States

**New State:** Sabse pehle to process hamari hoti hai ek new state ke andar ki abhi-abhi process hamari bani hi hai. Iske liye jo bhi resources hain vo allocate kar diye jaate hain, iska khud ka ek PCB bhi bana diya jaata hai.

**Ready State:** Uske baad ye process jaati hai ready state ke andar ki now I am ready ab mera kaam ho gaya mera initialization complete ho chuka hai now I am ready to be executed.

**Run State:** Ready ke baad jo scheduler hota hai vo schedule karta hai process ko aur usko run state mein daal deta hai. Run ka matlab ki abhi filhal ye CPU jo hai vo is process ko execute kar raha hai.

**Wait State:** Jab I/O operation hota hai input output operations jo hote hain us time pe vo process to chal nahi rahi hai to CPU to wait karega nahi. CPU kya karta us time pe us process ko daal deta wait state ke andar.

### Suspended States:

- **Suspend Wait:** Agar process bahut der se wait kar rahi hai I/O operation ka, wait queue badi lambi ho jaayegi to process ko suspend state mein daal dete hain
- **Suspend Ready:** Jab suspend wait mein process ne I/O complete kar liya but still suspend state mein hai

**Terminated State:** Agar hamari process puri ki puri complete ho gayi to process puri complete ab vo process hamari fir termination state mein chali aati hai.

## 5. CPU Scheduling Algorithms

**First Come First Serve (FCFS):** Jo process pehle aayi usko process ko pehle chala diya. Lekin is process ki vajah se ek problem ho jaati hai vo ye ki agar ek bahut lambi process aa gayi to uske peeche agar 50 choti-choti processes hain to vo nahi chal paayengi. Isko starvation problem kehte hain.

**Shortest Job First (SJF):** Jo sabse chota jis process ka burst time hai, burst time matlab jitne time ke liye isko CPU chahiye. Agar sabse chota burst time vali koi process aa rahi hai usko jaldi se karke khatam karo. Lekin isme bhi starvation problem ho jaati hai.

**Round Robin:** Round Robin ke andar basically kya hota hai ki kuch quantum quanta mein divide kar dete hain hum apne time ko CPU ke time ko aur hum bol dete hain ki har process ko itna itna time milta rahega circular fashion mein. Har bande ko 2-2 second ka time dete raho round robin fashion ke andar.

**Priority Scheduling:** Priority scheduling ye kehta hai ki main sabse zyada high priority ki process hoon mujhe sabse pehle execute karo. Lekin isme bhi problem hai ki jo bahut kam priority ki processes hain vo kabhi chal hi nahi paayengi.

**Multi Level Queue Scheduling:** Multi level queue scheduling ye kehta hai ki hum multiple queues bana lenge alag-alag priority ke hisab se. P0 vali processes ki ek alag queue banegi, P1 vali processes ki ek queue banegi. Basically hybrid ho gaya aur hybrid ki vajah se hum yahan par multi level queue scheduling kar pa rhe hain.

## 6. Critical Section Problem

**Critical Section Definition:** Critical section represent karta hai portion of code block where processes aur threads access karte hain shared resources such as variables, files, aur database. Ek baar mein sirf ek hi process jaani chahiye critical section mein.

**Bank Account Example:** Aapka bank hai ab bank ke andar jab aapke account se paise kahat rahe hote hain usi time pe suppose karo aapne kisi aur bande ko paise de bhi diye. Agar dono ke dono processes jo hai vo ek saath chal jaaye usi aapke total amount vale variable ke upar to kitni zyada gadbad ho sakti hai.

#### Requirements for Process Synchronization:

1. **Mutual Exclusion:** Agar critical section mein koi process ja rahi hai to ek baar mein ek hi process hogi jo critical section ko access karegi
2. **Progress:** Jo bhi aap mechanism bana rahe ho na synchronization karne ke liye vo aisa hona chahiye ki kam se kam ek process to aapki critical section mein jaani chahiye
3. **Bounded Waiting:** Agar koi process bahar wait kar rahi hai critical section mein jaane ka to it should be bounded

**Traffic Signal Analogy:** Process synchronization is like a traffic signal that helps regulate the flow of vehicles at an intersection. Agar intersection ke upar traffic police vala ya fir proper red light nahi lagi hui hai to gaadi aapas mein thuk ki rahengi.

#### 7. Process Synchronization Mechanisms

**Locks/Mutexes:** Jaise suppose karo P1 ko jaana tha yahan pe P1 ne kya kaha P1 ne laga diya lock. Jab P1 ja raha hai to P1 ne laga diya uske upar lock aur P1 ab ghus chuka hai critical section mein. Ab agar koi aur process ghusne ka try karti hai to us variable ko ya fir us lock ko check kiya jaata hai.

**Semaphores:** Semaphores bhi basically lock lagane ka hi tareeka hota hai lekin iske andar aapke do type ke locks hote hain:

- **Binary Semaphore:** Sirf do processes ko allow karoge ya to locked hai ya fir unlocked hai
- **Counting Semaphore:** Multiple processes ko bol sakte ho ki 5 processes aa sakti hain critical section ke andar

**Read-Write Locks:** Read-write lock kya kehta hai ki hum read karne ke liye multiple logon ko aane denge lekin write karna hai to mutual exclusion hona chahiye.

**Painting Example:** Jaise suppose karo ek painting hai painting lagi hui hai to us painting ko dekh to 10 log sakte hain koi dikkat nahi hai lekin jab uske upar kaam ho raha ho to ek baar mein ek hi painter kaam karega uske upar.

#### 8. Deadlock

**Deadlock Definition:** Deadlock ka matlab aisa ki bas aap atak gaye aur aap kuch bhi kaam nahi kar sakte.

**Placement Example:** Jaise ki suppose karo aapko ek company mein jaana hai placement paani hai aur vahan pe jo interviewer hai vo bolta hai ki suno bhai tumhare paas experience nahi hai pehle experience

gain karke aao tab hum tumhein lenge. Aur fir aap dusri company mein aate vo bhi yahi hota hai ki aap pehle experience gain karke aao tab vo aapko lenge. Toh is tareeke se aapko experience chahiye lekin experience ke liye aapko placement chahiye lekin placement ke liye aapko experience chahiye - deadlock lag gaya.

**Technical Deadlock Example:** Suppose karo ek process hai P1 isne ek resource R1 ko pakda hua hai aur isko nahi chodigi ye kyunki isko jarurat hai iski lekin isko R2 bhi chahiye kaam karne ke liye. Ab P2 ne R2 ko pakda hua hai aur P3 ne R3 ko pakda hua hai. Yahan pe ek tareeke se ek circle ban gaya hai aur koi resource ko chhod bhi nahi raha hai.

**Necessary Conditions for Deadlock:** Ye chaar conditions agar ek saath meet hongi tabhi deadlock hogा:

1. **Mutual Exclusion:** Ek resource ke upar ek baar mein ek hi process aa sakti hai
2. **Hold and Wait:** Ek process ne ek resource ko hold kiya hua hai aur dusri resource ka wait kar rahi hai
3. **No Preemption:** Jab tak process complete nahi ho jaayegi tab tak resource ko chodigi nahi
4. **Circular Wait:** Jab ye wait circular form mein ho jaata hai

#### **Deadlock Handling Techniques:**

**Deadlock Prevention:** Hum kya karenge hum kuch is tareeke ka system laga denge ki chaar conditions hoti hain in chaaron mein se koi ek condition na ho.

**Deadlock Avoidance:** Deadlock avoidance bhi similar sa hi hai lekin kuch complicated algorithm ka use kar raha hota hai jaise banker's algorithm.

**Deadlock Detection:** Kya karta hai ye check karta rahta hai har baar ki ye graphs ki help se ki ye jaisa graph to yahi ban raha hai na to is graph ke andar cycle to nahi hai.

**Deadlock Recovery:** Agar deadlock mein aa gaya system to hamein basically koi ek process ko terminate karna padega.

**Car Intersection Example:** Jaise intersection par chaar gadiyan aake fansi hui hain. Agar main yahan pe kisi ek gadi ko utha dun bahar nikaal dun to deadlock khatam.

**Deadlock Ignorance:** Deadlock ignore karenge kyunki deadlock ye maan ke chalte hain ki deadlock kabhi kabhar hi hogा 100 mein se ek baar ho sakta hai deadlock. Agar deadlock aa jaata hai to hum kya karenge crash kar denge system ko.

#### **9. Memory Management**

**Types of Memory:** Memory ka matlab kya hai RAM main yahan pe memory ka matlab vo nahi kah raha jo aapki hard drive hoti hai 500GB ki 2 terabyte ki vo nahi vo to secondary memory hai. Main kaam hota hai aapka RAM se jiske andar aapka program load hota hai aur RAM ke andar hi aapka program chal raha hota hai.

**Fixed Partitioning:** Fixed partitioning ka matlab ye hai ki aap kya karoge apni memory ko fixed partitions mein divide kar doge jinko size fixed hai. Problem: Internal fragmentation ho jaata hai ki aapki jo memory hai vo fragmentation ki aapki memory mein khaali jagah to hai lekin vo use nahi ki ja sakti.

**Dynamic Partitioning:** Dynamic partitioning kya kehta hai ye kehta hai ki jo process aayegi suppose karo P1 aayi to P1 ko hamne ye space de diya. P1 ka agar 4 byte ka space tha to hamne 4 byte ka space P1 ko pakda diya. Problem: External fragmentation ho jaata hai.

#### Algorithms for Dynamic Partitioning:

**First Fit:** Jo pehla space mil gaya which is large enough for a process vahi pe us memory ko us process ko daal do.

**Best Fit:** Ek baar pure ke pure jitne bhi khaali spaces hain na memory ke andar unko ek baar iterate karke dekh lo ki kaun sa hai jo ki just isse bada hai ya fir just iske barabar hai.

**Worst Fit:** Jo sabse badi khaali space hai na uske andar processes ko daalna start karo so that agar 8 byte ka space hai khaali to sabse bada space hai isi mein daal do P4 ko.

### 10. Paging

**Paging Definition:** Paging is a storage mechanism used to retrieve processes from secondary storage into main memory in the form of pages. Ab hum yahan par bol dete hain ki hamara jo main memory hai usko hamne divide kar diya bahut chote-chote pages ke andar.

**Page Table:** Ab humein contiguous fashion mein rakhne ki jarurat nahi hai. Humein ye vali mapping jo hai vo Memory Management Unit jo hai vo manager apna kaam karta hai. Is manager ke liye jo manager hai ye ek database ka use leta hai jiska hum naam hum bolte hain page table.

**Memory Management Unit (MMU):** Ek manager assign karna padega operating system ko. Operating system bolega suno bhai mujhe in sabki tension nahi leni mujhe to is tareeke se chahiye ki jaise ki process P5 contiguous fashion mein rakhi ho. Main ek manager assign kar deta hun iske liye to vo delegate kar deta hai is manager ko memory management unit ko.

#### Logical vs Physical Address:

- **Logical Address:** Operating system ko jo chahiye hota hai
- **Physical Address:** Ye jo hai vo yahan par pada hua hai

Page table ke andar ye mapping ho rakhi hoti hai aur is mapping ki help se MMU basically memory management unit operating system ko physical address pe pahucha paati hai.

### 11. Virtual Memory

**Virtual Memory Definition:** Virtual memory is a concept that lets computer use more memory than it actually has. It creates an imaginary memory space by combining physical memory and secondary memory.

**Working Principle:** Jaise aapko bataya ki multiple processes chal rahi hoti hain ek baar mein ek processor multiple processes ko chala raha hota hai. Lekin hamein pata hai ki hamara jo game hai vo 56GB ka hai lekin vo sirf 16GB RAM ke andar aa gaya.

**90-10 Rule:** 90% of the time ek process sirf 10% of the memory hi use kar rahi hoti hai. 90% of the time sirf 10% of the memory use kar rahi hai baki 90% of the memory vo use nahi kar rahi hoti hai.

**Demand Paging:** Jis cheez ki demand hai jis program ke andar process ke andar jis code ki demand hai sirf usi piece of code ko hum load karenge apni main memory ke andar. Jab iske ki demand khatam ho jaayegi to isko nikaal denge memory ke andar aur agar kisi aur process ko koi aur page chahiye to us page ko load kar denge.

**Impact of RAM Size:** Agar RAM aapki badi hogi to degree of multiprogramming badh jaayegi aapki. Agar RAM aapki 8GB hai to aap kam programs load kar paaoge. Vahi pe agar aapke paas 16GB RAM hai to aap usmein aur zyada programs ko load kar paaoge.

## 12. Page Replacement Algorithms

**Page Fault vs Page Hit:**

- **Page Fault:** Jab aapki memory ko jab aapke program ko ek memory piece of code ki requirement hai aur vo aapke memory ke andar nahi pada hua hai primary memory ke andar nahi hai to us time pe hota hai page fault
- **Page Hit:** Jis page ki requirement thi vo primary memory ke andar pada hua hai

**FIFO (First In First Out):** FIFO page replacement algorithm basically hota kya hai ki hamake paas kuch limited number of pages hote hain. Jo pehle aaya usko pehle bahar nikalo. Problem: Belady's anomaly ho sakti hai.

**Belady's Anomaly:** Hona to ye chahiye hai na ki jab aap number of frames ko badha doge to automatic mere page faults kam ho jaane chahiye lekin agar aap FIFO algorithm use karoge to kai baar dekha jaata hai aisa ki frames ko badha bhi dete hain tab bhi page faults zyada ho rahe hain.

**Optimal Page Replacement:** Ye ekdam best hai sabse badhiya page replacement algorithm hai. Ye algorithm basically kya kehti hai ki jo page sabse last mein use hogi na jo ekdam end mein use hogi usko sabse pehle bahar nikalo. Problem: Ye feasible nahi hai isko implement nahi kar sakte.

**LRU (Least Recently Used):** Ye kafi heavily use ki jaati hai. Ye kya kehti hai basically ki jo page abhi recently use kiya gaya tha na usko mat nikalo jo least recently use kiya gaya tha usko bahar nikalo. Jo abhi filhal mein use hua hai us page ko mat nikalo.

## 13. Thrashing

**Thrashing Definition:** Jab aapka jo computer hai operating system hai vo computation kam kar raha hai aur bas page replacement kiya ja raha hai pure time. Bas page replacement mein laga hua hai aur computation kam kar raha hai to us time par thrashing ho jaati hai.

**When Thrashing Occurs:** Thrashing kab hogi jab aapne computer ke andar bahut saari degree of multiprogramming ko bahut zyada high kar diya hai. Aap degree of multiprogramming high karte jaate ho to aapki obviously CPU utilization badhti jaati hai.

**Thrashing Graph:** Jaise-jaise aap degree of multiprogramming badha rahe ho aapki throughput improve ho rahi hai. Lekin isse zyada jab aap program load karne ka try karte ho to hota kya hai CPU utilization kam hone lag jaati hai. CPU utilization kam kyun ho rahi hai kyunki ab hoga kya ki page hatenge page aayenge page hatenge page aayenge.

**How to Avoid Thrashing:** Isko avoid karne ka tareeka yahi hai bas degree of multiprogramming ko kam karo. Ek optimal state mein rakho aur dusra RAM ko badhao.

#### 14. Segmentation

**Segmentation Concept:** Paging hamne dekha tha paging ke andar aap logon ki jo primary memory hai usko multiple pages mein divide kar dete hain lekin iske saath ek issue hai. Iske saath issue ye hai ki aapne fixed size ke pages bana rakhe hain.

**User View vs Computer View:** Is segmentation mein computer ke view ke hisab se nahi user ke view ke hisab se chala jaata hai. User ka jo program hai vo kehta hai ki mujhe ye vala procedure to pura ka pura execute karna hai to kyun na is pure ke pure procedure ko ek saath load kar diya jaaye.

**Function-wise Loading:** Aapka jo code hai vo suppose karo procedure one hai uske andar procedure two hai ek stack hai isme ek library function hai isme ek main function hai to isko functions mein tod diya jaata hai code ko aur ye pura ka pura function fir memory mein load kar diya jaata hai.

**Segmentation + Paging:** Segmentation aur paging ko ek saath chala loge matlab aap log paging bhi kar rahe ho lekin paging kis basis pe kar rahe ho aap segment ke basis pe kar rahe ho ki aap pura ka pura segment load kar de rahe ho.

#### 15. Disk Management

##### Types of Secondary Storage:

**Hard Disk Drive (HDD):** Aapne dekha hogा aapki solid state drive hoti hai ek hoti hai HDD hard disk drive hoti hai. Hard disk drive aajkal thodi kam use hoti hai kyunki iske andar mechanical pieces bahut zyada hote hain. Isme ye sab rotational vagerah laga hota hai.

**Solid State Drive (SSD):** SSD jo hai vo much faster hoti hai HDD se vo isliye kyunki uske andar moving parts vagerah nahi hote hain. SSD ke andar sun ke dekhoge koi aavaaz nahi aati. HDD ke andar bahut aavaazen aati hain vo kyunki usmein cheezein ghoom rahi hoti hain.

**Data Storage Mechanism:** Saara ka saara khel aapke electron ki density ke upar hai. Electron ki density high hai to vahan par one maan liya jaayega electron ki density low hai to vahan pe zero maan liya jaayega. To is tareeke se aapke bits define hote hain.

**Disk Structure:** Iske andar bahut saare tracks hain ye ek track hai uske baad ye ek track hai is tareeke se ek CD ke andar bahut saare tracks hote hain ek ke baad ek ke baad ek. In tracks mein bhi data jo hai vo alag-alag sectors mein hai.

**Seek Time:** Seek time is the time taken by locating the disk arm to specify the track where the read/write request will be satisfied. Ye jo head hai ye head track ko seek karega ki mujhe kaun si track pe jaana hai.

**Rotational Latency:** Seek time ke baad jab use mil jaata hai ki theek hai mujhe isi vale track pe jaana hai ab mujhe dekhna hai ki mujhe kaun se vale sector pe jaana hai to uske liye vo disk ghumti hai. Uske liye vo disk ghoom ke head vali position pe aa jaati hai use hum kehte hain rotational latency.

## 16. Disk Scheduling Algorithms

**FCFS (First Come First Serve):** Jo data pehle chahiye us jagah par ghuma do. Suppose karo 2 aaya pehle to 2 vali jagah pe ghuma do uske baad aaya 100 to 100 vali jagah pe ghumaao.

**SSTF (Shortest Seek Time First):** Shortest seek time first kehta hai ki hum zero se challenge aur aage badhte rahenge aur jo jo hamein milta jaayega us usko hum dete jaayenge. Isme average response time decrease ho jaata hai.

**SCAN Algorithm:** SCAN algorithm kya karti hai SCAN algorithm bolti hai ki main yahan se pehle right mein jaungi fir main left mein jaungi fir main right mein jaungi. Pehle 82 mil gaya idhar fir uske baad 142 chahiye tha fir pura ka pura fir ekdam end tak leke aata hai.

**C-SCAN Algorithm:** C-SCAN bhi SCAN ki tarah hi hota hai lekin C-SCAN basically karta kya hai C-SCAN bolta hai ki suno abhi main right side mein gaya uske baad ab main pura ka pura left side mein chale jaunga. Left side mein aate time main cheezon ko scan nahi karunga.

**LOOK Algorithm:** LOOK algorithm ke andar it is similar to SCAN disk algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last requested. Ye 190 se hi retrace back kar lega aur retrace back karte time dobara se beech-beech mein scan karta chalega.

**C-LOOK Algorithm:** C-LOOK is similar to C-SCAN aur bas ye kya hai ki ye end tak nahi jaata ye beech mein rakhta hai jahan jahan par data scan karna hai utne tak jaata hai pura end tak nahi ghumaata hai.

## 17. File Management (Brief)

File management bhi aata hai disk management mein lekin zyada pucha nahi jaata. Disk scheduling algorithms zyada puche jaate hain.

## 18. Complete Revision Summary

**Operating System Role:** Hardware aur software ke beech interface, memory management, process scheduling, disk management

**Process Management:** Program → Process → Thread hierarchy, PCB, IPC, process states (New→Ready→Run→Wait→Terminated)

**CPU Scheduling:** FCFS, SJF, Round Robin, Priority, Multi-level Queue

**Synchronization:** Critical section, mutual exclusion, locks, semaphores, deadlock (4 conditions), deadlock handling

**Memory Management:** Fixed/Dynamic partitioning, paging, virtual memory, page replacement algorithms (FIFO, Optimal, LRU), thrashing, segmentation

**Storage:** HDD vs SSD, tracks/sectors, seek time, rotational latency, disk scheduling algorithms (FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK)

Ye saare concepts operating system ke andar jo aapke interviews ke andar puche jaate hain aur engineering ke andar bhi kafi had tak samajh jaooge isko concept ko padh ke.