IAE ACADEMY INSTITUTE FOR ENGINEERS

CORE JOBS · INTERVIEW QUESTIONS

IAE Academy free study material · Updated 28 October 2026

Operating Systems Interview Questions

INSIDE

- 25 questions with short answers
- 6 numericals with the working
- Grouped by topic

Companies do not publish their interview questions; these are ours.

Online, with links to the notes: **iaeacademy.in/study/operating-systems/interview-questions**

FREE COUNSELLING

Preparing for GATE or a core job? Talk to IAE Academy.

Or call **+91 84204 36524**

Every day, 10:30 am to 7:30 pm

Chat on WhatsApp

Free. Not for sale. Pass it on.

Processes, threads and system calls

1. What is a system call, and why does a user program need one?

A system call is the controlled entry point through which a user program asks the kernel to do something privileged, such as reading a file. The CPU switches from user mode to kernel mode, runs the kernel code, then returns. This keeps user code from touching hardware or other processes' memory directly.

2. What is the difference between a process and a thread?

A process has its own address space and kernel record (the PCB). A thread is a unit of execution inside a process. Threads of one process share code, data, heap and open files, but each keeps its own stack, registers and program counter. So threads are cheaper to create and switch, but one faulty thread can corrupt shared data.

3. What is a zombie process, and why does the system keep one?

A zombie is a process that has finished but whose process-table entry remains because its parent has not yet called wait(). The entry holds the exit status the parent may want to read. It uses no CPU and almost no memory, and it is removed once the parent waits or exits and the system reaps it.

4. A program calls fork() three times one after another, with no conditions around the calls. How many processes exist at the end, counting the original?

There are 8 processes in total, the original plus 7 new ones, because each fork() call doubles the number of processes that exist at that point.

> **WORKING**
>
> 1. Each fork() call makes every existing process create one child.
> 2. So each call doubles the number of processes.
> 3. After 1 call: 2. After 2 calls: 4. After 3 calls: 8.
> 4. In general, n unconditional calls give 2^n processes.

Inter-process communication, concurrency and synchronisation

5. What is a race condition, and what is a critical section?

A race condition occurs when the result depends on the order in which concurrent processes access shared data. The code that touches that shared data is the critical section. It must run with mutual exclusion, otherwise interleaved steps, such as two increments of one counter, can lose an update.

6. What is the difference between a mutex and a semaphore?

A mutex is a lock owned by the thread that locked it, and only that thread should unlock it. It is used for mutual exclusion around a critical section. A semaphore is an integer counter with wait and signal operations that any process may call. A counting semaphore can admit several holders, and a semaphore can also signal between processes.

7. Why does a lock built from a plain shared flag, checked and then set, fail to give mutual exclusion?

Checking the flag and setting it are two separate steps. A context switch between them lets two processes both see the flag free, and both enter the critical section. Mutual exclusion needs the test and the set to be

one indivisible step, as in a hardware test-and-set or compare-and-swap instruction.

8. How do semaphores solve the bounded-buffer producer and consumer problem?

Use a counting semaphore empty, started at the buffer size, for free slots. Use another, full, started at 0, for filled slots. Use a mutex around the buffer itself. The producer waits on empty, then signals full after adding an item. The consumer waits on full, then signals empty after removing one. Each takes its counting semaphore before the mutex, since the reverse order can deadlock.

Deadlock

9. What are the four necessary conditions for deadlock?

Mutual exclusion, hold and wait, no preemption and circular wait. All four must hold together for a deadlock to occur. Breaking any one prevents it. For example, making every process request resources in one fixed global order removes circular wait.

10. What is the difference between deadlock prevention and deadlock avoidance?

Prevention designs the system so that at least one of the four necessary conditions can never hold, whatever the requests. Avoidance allows all four to be possible, but grants a request only if the resulting state is safe. It needs each process's maximum demand in advance, as the Banker's algorithm does.

11. A system has 3 processes, and each needs at most 4 units of one resource type, all units being identical. What is the smallest number of units that guarantees no deadlock?

A minimum of 10 units guarantees that no deadlock can occur, because 9 units can still leave all three processes waiting while holding 3 units each.

> **WORKING**
>
> 1. Worst case: every process holds one unit fewer than its maximum, so 3 units each.
> 2. Total held in that case: $3 \times 3 = 9$ units, and all three wait for one more.
> 3. With one extra unit, some process can get its last unit, finish and release its units.
> 4. So the minimum is $n(k-1)+1 = 3(4-1)+1 = 10$.

CPU scheduling

12. What are turnaround time, waiting time and response time of a process?

Turnaround time is completion time minus arrival time. Waiting time is turnaround time minus CPU burst time, which is the time spent in the ready queue for a process that does no I/O. Response time is the time of first getting the CPU minus arrival time. A scheduler usually tries to lower the averages of these.

13. What is the difference between SJF and SRTF scheduling?

SJF is non-preemptive: a running process finishes its burst, and the waiting process with the shortest burst runs next. SRTF is the preemptive form: a newly arrived process with a shorter remaining time takes the CPU at once. For the same processes, SRTF's average waiting time is never higher than SJF's, but it needs more context switches. Both can starve long processes.

14. Why does the choice of time quantum matter in round robin scheduling?

If the quantum is very large, round robin behaves like FCFS and response time suffers. If it is very small, too much time goes on context switches. A good quantum is longer than the typical short burst, so short processes finish in one turn, while every process still gets regular turns.

15. Three processes P1, P2 and P3 all arrive at time 0 with CPU bursts of 6 ms, 8 ms and 3 ms. Using non-preemptive SJF, what is the average waiting time?

The average waiting time under non-preemptive SJF is 4 ms, since the three processes wait 0 ms, 3 ms and 9 ms before they start.

WORKING

1. Order by burst: P3 (3 ms), P1 (6 ms), P2 (8 ms).
2. P3 starts at 0, so its wait is 0 ms.
3. P1 starts at 3, so its wait is 3 ms. P2 starts at 9, so its wait is 9 ms.
4. Average $= (0 + 3 + 9)/3 = 4$ ms.

Memory management

16. What is the difference between internal and external fragmentation?

Internal fragmentation is wasted space inside an allocated block, such as the unused end of a process's last page. External fragmentation is free memory split into small separate holes, none big enough for a request. Paging has only internal fragmentation, while variable partitions suffer external, which compaction can fix.

17. Why do operating systems use paging instead of contiguous allocation?

Paging splits a process's logical space into fixed-size pages that can sit in any free frames, so no contiguous physical block is needed. That removes external fragmentation and makes allocation simple. The costs are a page-table lookup on each access and some internal fragmentation in each process's last page.

18. A system has a single-level page table in memory and a TLB. A TLB lookup takes 10 ns, a memory access takes 100 ns, and the TLB hit ratio is 90%. What is the effective memory access time?

The effective memory access time is 120 ns: the 110 ns hit time and the 210 ns miss time, weighted by the 90% hit and 10% miss chances.

WORKING

1. On a hit: TLB lookup plus one memory access, $10 + 100 = 110$ ns.
2. On a miss: TLB lookup, page-table access, then the data access, $10 + 100 + 100 = 210$ ns.
3. $\text{EAT} = 0.9 \times 110 + 0.1 \times 210$.
4. $= 99 + 21 = 120$ ns.

19. A system has a 32-bit logical address, a page size of 4 KB and a page-table entry of 4 bytes, with a single-level page table. What is the size of the page table?

The single-level page table is 4 MB in size, because the 12-bit offset leaves 20 bits of page number, giving 2^{20} entries of 4 bytes each.

WORKING

1. Page size $= 4\ \text{KB} = 2^{12}$ bytes, so the offset takes 12 bits.
2. Page-number bits $= 32 - 12 = 20$, so there are 2^{20} pages.
3. Table size $= 2^{20} \times 4\ \text{B} = 2^{22}$ B.
4. 2^{22} B $= 4$ MB.

Virtual memory

20. What is demand paging, and what happens on a page fault?

Demand paging loads a page into memory only when it is first referenced. A reference to a page not in memory raises a page fault. The OS traps, finds the page on disk, loads it into a free frame (evicting a victim if none is free), updates the page table and restarts the instruction.

21. Why can giving a process more frames sometimes increase its page faults under FIFO replacement?

This is Belady's anomaly. FIFO evicts by arrival order, not by use, so the pages held with more frames are not always a superset of those held with fewer. LRU and optimal are stack algorithms: the pages held with n frames are always inside those held with n+1, so they never show it.

22. A memory access takes 200 ns when the page is present. A page fault takes 8 ms to service. The page-fault rate is 1 in 10,000 accesses. What is the effective access time?

The effective access time is about 1000 ns (999.98 ns), roughly five times the 200 ns of an access with no fault, even though only 1 access in 10,000 faults.

WORKING

1. Fault rate $p = 10^{-4}$. Fault service time $= 8\ \text{ms} = 8{,}000{,}000$ ns.
2. $\text{EAT} = (1 - p) \times 200 + p \times 8{,}000{,}000$.
3. $= 0.9999 \times 200 + 800$.
4. $= 199.98 + 800 = 999.98$ ns.

File systems and disk scheduling

23. What is the difference between contiguous, linked and indexed file allocation?

Contiguous allocation stores a file in adjacent blocks: fast access, but external fragmentation and hard growth. Linked allocation chains blocks by pointers: no external fragmentation, but slow direct access. Indexed allocation keeps an index block of pointers: direct access and no external fragmentation, at the cost of the index block's space.

24. Why does the operating system schedule disk requests, and how does SCAN work?

Seek time, the head moving between cylinders, dominates disk access time, so ordering requests to cut head movement raises throughput. SCAN, the elevator algorithm, moves the head one way serving requests on the path, then reverses at the end. Unlike shortest-seek-first, it does not starve distant requests.

25. What is an i-node, and what does it store?

An i-node is a per-file structure holding the file's metadata (size, owner, permissions, timestamps) and pointers to its data blocks: direct pointers, plus single, double and triple indirect ones. The file's name is kept in the directory, not in the i-node. Small files need only the direct pointers.

FROM IAE ACADEMY

Preparing for GATE or a core job?

IAE Academy teaches GATE in seven papers in Ballygunge, Kolkata, and online. Counselling is free: ask us about batches, fees or your preparation.

FREE COUNSELLING

Chat on WhatsApp

Call +91 84204 36524

Open every day, 10:30 am to 7:30 pm.

Scan to chat on WhatsApp

GATE COURSES

Classroom batches in Ballygunge and online batches. Fees and details are on the course pages.

Scan: courses

See the courses

FREE ON IAEACADEMY.IN

Notes and solved previous-year questions for Operating Systems, topic by topic.

Scan: the notes

Open the notes

Found this useful? Send it to a friend preparing for core-job interviews.

Send to a friend on WhatsApp

IAE Academy free study material. Companies do not publish their interview questions; these are ours. Prepared and published by IAE Academy, Kolkata. Free to read and free to share, unaltered and with attribution.

IAE ACADEMY INSTITUTE FOR ENGINEERS

GATE coaching in Kolkata since 2012 · EC, EE, IN, CS and IT, DA, BT and CE · in the classroom and online

227A Rash Behari Ave, Ballygunge, Kolkata 700019

+91 84204 36524 · iaeacademy.in · Updated 28 October 2026

Get directions