UnCodemy

Operating System Important Questions PDF

The most important Operating System questions are CPU scheduling numericals (FCFS, SJF, SRTF, priority, round robin), the critical section problem, semaphores and classic synchronisation problems, the Banker's algorithm, paging and segmentation, page replacement (FIFO, LRU, optimal) with Belady's anomaly, thrashing, disk scheduling (SSTF, SCAN, C-SCAN) and file allocation methods.

Key takeaways

- Scheduling, Banker's algorithm, page replacement and disk scheduling numericals carry the most marks.
- Draw a Gantt chart for every scheduling question and show waiting and turnaround time in a table.
- Learn the pseudocode for producer-consumer, readers-writers and dining philosophers.
- These are practice questions, not past papers; check your own syllabus and official previous papers.

About these important questions

This page lists the most important Operating Systems questions for B.Tech CSE and IT exams, arranged unit by unit. The units follow the standard syllabus that most Indian technical universities share, so the order may differ slightly from yours.

Please note: these are practice questions based on common syllabus topics. They are not verified past papers, and we do not claim that any question will appear in your exam. Check your own university syllabus first, then solve the official previous year papers. AKTU students can download them from the AKTU old question paper archive.

Unit 1: Introduction and OS Structure

Covers OS functions, types, structure, system calls and interrupts.

Short questions (2 marks)

- Define an operating system and list its main functions.
- Differentiate between multiprogramming and multitasking.
- What is a system call? Give two examples.
- What is dual-mode operation?

Long questions

1. Explain the types of operating systems: batch, time-sharing, distributed, real-time and network.
2. Compare monolithic, layered and microkernel structures with diagrams.
3. Explain the types of system calls with examples.
4. Explain spooling, buffering and interrupts.
5. Explain the boot process and the role of the kernel, the shell and system programs.

Unit 2: Concurrent Processes and Synchronisation

Covers the critical section problem, semaphores, monitors and classic problems.

Short questions (2 marks)

- What is a race condition?
- State the three requirements of a critical section solution.
- Differentiate between a binary and a counting semaphore.

Long questions

1. Explain Peterson's solution and prove that it meets mutual exclusion, progress and bounded waiting.
2. Solve the producer-consumer (bounded buffer) problem using semaphores.
3. Solve the readers-writers and dining philosophers problems with semaphores.
4. Explain monitors and how they differ from semaphores.
5. Explain inter-process communication: shared memory and message passing.

Unit 3: CPU Scheduling and Deadlocks

Covers process states, PCB, threads, scheduling algorithms and deadlock handling. This is the biggest numerical unit.

Short questions (2 marks)

- Draw the process state diagram.
- Differentiate between preemptive and non-preemptive scheduling.
- What is starvation and aging?

Long questions

1. For a given set of processes with arrival and burst times, draw Gantt charts for FCFS, SJF, SRTF, priority and round robin (quantum 2) and find average waiting and turnaround times.
2. Explain multilevel queue and multilevel feedback queue scheduling.
3. State the four necessary conditions for deadlock.
4. Apply the Banker's algorithm: find the Need matrix, check for a safe sequence and decide whether a request can be granted.
5. Explain deadlock detection using a resource allocation graph and recovery methods.
6. Compare user-level and kernel-level threads.

Unit 4: Memory Management

Covers allocation, paging, segmentation and virtual memory.

Short questions (2 marks)

- Differentiate between internal and external fragmentation.
- What is a TLB?
- What is thrashing?

Long questions

1. Explain first fit, best fit and worst fit with a numerical.
2. Explain paging with a diagram, and calculate effective access time with a TLB hit ratio.
3. Explain segmentation and segmentation with paging.
4. For a reference string, count page faults with FIFO, LRU and optimal for 3 and 4 frames. Explain Belady's anomaly.
5. Explain demand paging and the steps to handle a page fault.
6. Explain the causes of thrashing and the working-set model.

Unit 5: I/O Management and File Systems

Covers disk scheduling, file allocation and directory structures.

Short questions (2 marks)

- Define seek time and rotational latency.
- What is RAID?

Long questions

1. For a disk queue such as 98, 183, 37, 122, 14, 124, 65, 67 with the head at 53, find total head movement for FCFS, SSTF, SCAN, C-SCAN and LOOK.
2. Explain contiguous, linked and indexed file allocation with diagrams.
3. Explain single-level, two-level and tree directory structures.
4. Explain free-space management methods: bit vector and linked list.
5. Explain RAID levels 0, 1 and 5.
6. Explain file access methods (sequential, direct and indexed) and file protection using access control lists.

Sample answer outline: Round robin scheduling numerical

Round robin is the scheduling question students lose most marks on, because of the ready queue order. Use these steps:

1. **Copy the data into a table.** Process, arrival time, burst time and remaining time.
2. **Keep a ready queue on the side.** At each time step, add newly arrived processes to the queue before you put the preempted process back at the end. State this rule in your answer, since some textbooks differ.

3. **Draw the Gantt chart.** Each slot is the time quantum or the remaining burst, whichever is smaller. Write the time under every boundary.
4. **Compute the results.** Completion time from the chart, turnaround time = completion minus arrival, waiting time = turnaround minus burst. Show a final table and the averages.
5. **Comment briefly.** A very large quantum makes round robin behave like FCFS, while a very small one adds context-switch overhead. One line on this shows understanding.

The same table-plus-Gantt-chart format works for FCFS, SJF, SRTF and priority scheduling.

Marks weightage and how to write answers

A common pattern, used by AKTU and many other universities, has a compulsory short-answer section that covers every unit, followed by long-answer questions with a choice inside each unit. Marks per question change between universities and years, so confirm the scheme on a recent official paper.

- **Short answers (about 2 marks):** a crisp definition, one example or diagram, and nothing more. Three to five lines is enough.
- **Long answers (7 to 10 marks):** start with a definition, add a labelled diagram or table, explain the steps, and finish with an example, advantages or a comparison.
- Draw a neat Gantt chart with time markings and a results table. Most marks in scheduling questions come from these two.
- In the Banker's algorithm, show the Work and Finish values at every step, not only the final safe sequence.

A 7-day revision plan for Operating System

1. **Days 1 to 5:** one unit a day. Read your notes, then write the long answers for that unit from memory and check them against your notes.
2. **Day 6:** redo one full numerical each of scheduling, Banker's algorithm, page replacement and disk scheduling.
3. **Day 7:** solve one full official previous year paper in three hours, then check which units cost you time.

Keep a single sheet of formulas, definitions and diagrams for the last hour before the exam.

Frequently asked questions

Which Operating System topics are most important?

CPU scheduling, process synchronisation with semaphores, deadlock and the Banker's algorithm, paging and page replacement, and disk scheduling are asked most often.

How do I avoid mistakes in CPU scheduling numericals?

Write the arrival and burst times in a table, draw the Gantt chart, then compute completion time, turnaround time (completion minus arrival) and waiting time (turnaround minus burst) for each process.

Is Operating System asked in placement interviews?

Yes. Interviewers often ask about processes versus threads, deadlock, paging, virtual memory and scheduling, especially for software developer roles.

Are these Operating System questions from previous year papers?

No. They are practice questions built from topics that appear in most B.Tech Operating Systems syllabi. Use them with your own university's official previous papers, for example the AKTU archive at https://aktu.ac.in/old-question-paper.html.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- AKTU: old question papers (official archive)
- Operating Systems: Three Easy Pieces (free textbook)
- NPTEL: online courses from IITs (search for Operating System)

About this PDF

These are practice questions for revision, prepared by the Uncodemy trainer team from common B.Tech syllabus topics. They are not past papers, a certificate or a promise of marks. Spot an error? Write to info@uncodemy.com and the team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550