

1. What is the input and output of Pass I and Pass II of Assembler and Macro?

Pass I: Input – Assembly source code. Output – Symbol table & intermediate code. It assigns addresses and identifies labels. Pass II: Input – Intermediate code & symbol table. Output – Final machine code. Macro: Expands macro calls into full assembly instructions.

2. What is Preemptive and Non-preemptive Scheduling?

Preemptive: CPU can be taken away from a running process (e.g., Round Robin, SRTF).

Non-preemptive: Once a process starts, it runs till completion (e.g., FCFS, SJF).

3. What is Turnaround Time, Response Time, and Waiting Time?

Turnaround Time = Completion – Arrival time. Response Time = First response – Arrival time. Waiting Time = Turnaround – Burst time.

4. Which algorithm is best when FCFS, RR, Priority, and SJF are compared?

SJF gives best average waiting & turnaround time. RR is fair for time-sharing. Priority depends on importance; FCFS is simple but inefficient.

5. Which algorithm is best when LRU, Optimal, and FIFO are compared?

Optimal gives least page faults (ideal). LRU performs well in practice. FIFO is simple but may cause Belady's anomaly.

6. What is Macro and Assembler?

Macro: A single instruction that expands into multiple instructions. Assembler: Converts assembly language to machine code and maintains a symbol table.

7. Why is Banker's Algorithm required?

Used to avoid deadlocks by ensuring system remains in safe state before resource allocation.

8. What is Deadlock? What are the four necessary conditions for Deadlock?

Deadlock: Processes wait indefinitely for each other's resources. Conditions: Mutual Exclusion, Hold & Wait, No Preemption, Circular Wait.

9. Explain examples of FCFS, RR, SJF, and Priority Scheduling.

FCFS – Based on arrival order. SJF – Shortest job first. RR – Time-sharing. Priority – Based on process priority.

10. Explain examples of FIFO, Optimal, and LRU Page Replacement.

FIFO – Replace oldest page. Optimal – Replace page not used for longest future time. LRU – Replace least recently used page.

11. What is the difference between Pass I and Pass II of Assembler?

Pass I builds symbol table & intermediate code. Pass II generates object code using the table.

12. What is the drawback of Pass I of Assembler?

Cannot handle forward references; only produces intermediate code.

13. What is the difference between Compiler and Interpreter?

Compiler: Translates full program at once (faster execution after compile). Interpreter: Translates line-by-line (slower, executes directly).

14. In which different parts is Assembly Language divided?

1. Imperative Statements – Perform operations. 2. Declarative Statements – Define storage. 3. Assembler Directives – Control assembler actions.

15. What is a Process? What are the different states of Processes?

Process: A program in execution. States: New, Ready, Running, Waiting, Terminated.

16. What is Paging?

Memory divided into pages (logical) and frames (physical). Pages are mapped to frames to avoid fragmentation.

17. What is Segmentation?

Memory divided into variable-sized segments like code, stack, data. Represents logical structure of program.

18. What are Pages and Frames?

Page: Logical memory block. Frame: Physical memory block. Pages are mapped to frames using page tables.

19. Why is there a need for CPU Scheduling Algorithms?

To improve CPU utilization, reduce waiting time, and increase throughput and responsiveness.

20. What are different types of Schedulers? Explain Long-term, Middle-term, and Short-term.

Long-term: Selects processes from job pool. Short-term: Allocates CPU to ready processes. Medium-term: Swaps processes in/out of memory.