



# Operating System Questions and Answers – CPU Scheduling Algorithms-1

[<< Prev](#)[Next >>](#)

This set of Operating System Multiple Choice Questions & Answers (MCQs) focuses on “CPU Scheduling Algorithms-1”.

1. Round robin scheduling falls under the category of \_\_\_\_\_

- a) Non-preemptive scheduling
- b) Preemptive scheduling
- c) All of the mentioned
- d) None of the mentioned

[View Answer](#)

Answer: b

Explanation: None.

advertisement

2. With round robin scheduling algorithm in a time shared system \_\_\_\_\_
- a) using very large time slices converts it into First come First served scheduling algorithm
  - b) using very small time slices converts it into First come First served scheduling algorithm
  - c) using extremely small time slices increases performance
  - d) using very small time slices converts it into Shortest Job First algorithm

View Answer

Answer: a

Explanation: All the processes will be able to get completed.

3. The portion of the process scheduler in an operating system that dispatches processes is concerned with \_\_\_\_\_
- a) assigning ready processes to CPU
  - b) assigning ready processes to waiting queue
  - c) assigning running processes to blocked queue
  - d) all of the mentioned

View Answer

Answer: a

Explanation: None.

4. Complex scheduling algorithms \_\_\_\_\_
- a) are very appropriate for very large computers
  - b) use minimal resources
  - c) use many resources
  - d) all of the mentioned

View Answer

5. What is FIFO algorithm?

- a) first executes the job that came in last in the queue
- b) first executes the job that came in first in the queue
- c) first executes the job that needs minimal processor
- d) first executes the job that has maximum processor needs

[View Answer](#)

Answer: b

Explanation: None.

advertisement

6. The strategy of making processes that are logically runnable to be temporarily suspended is called \_\_\_\_\_

- a) Non preemptive scheduling
- b) Preemptive scheduling
- c) Shortest job first
- d) First come First served

[View Answer](#)

Answer: b

Explanation: None.

7. What is Scheduling?

d) none of the mentioned

[View Answer](#)

Answer: a

Explanation: None.

8. There are 10 different processes running on a workstation. Idle processes are waiting for an input event in the input queue. Busy processes are scheduled with the Round-Robin time sharing method. Which out of the following quantum times is the best value for small response times, if the processes have a short runtime, e.g. less than 10ms?

a)  $t_Q = 15\text{ms}$

b)  $t_Q = 40\text{ms}$

c)  $t_Q = 45\text{ms}$

d)  $t_Q = 50\text{ms}$

[View Answer](#)

Answer: a

Explanation: None.

9. Orders are processed in the sequence they arrive if \_\_\_\_\_ rule sequences the jobs.

a) earliest due date

b) slack time remaining

c) first come, first served

d) critical ratio

[View Answer](#)

Answer: c

Explanation: None.

advertisement

10. Which of the following algorithms tends to minimize the process flow time?

- a) First come First served
- b) Shortest Job First
- c) Earliest Deadline First
- d) Longest Job First

[View Answer](#)

Answer: b

Explanation: None.

11. Under multiprogramming, turnaround time for short jobs is usually \_\_\_\_\_ and that for long jobs is slightly \_\_\_\_\_

- a) Lengthened; Shortened
- b) Shortened; Lengthened
- c) Shortened; Shortened
- d) Shortened; Unchanged

[View Answer](#)

Answer: b

Explanation: None.

12. Which of the following statements are true? (GATE 2010)

- I. Shortest remaining time first scheduling may cause starvation
- II. Preemptive scheduling may cause starvation
- III. Round robin is better than FCFS in terms of response time

- a) I only
- b) I and III only
- c) II and III only
- d) I, II and III

[View Answer](#)

process never gets CPU.

II) Preemption may cause starvation. If priority based scheduling with preemption is used, then a low priority process may never get CPU.

III) Round Robin Scheduling improves response time as all processes get CPU after a specified time.

advertisement

## Sanfoundry Global Education & Learning Series – Operating System.

Here's the list of Best Reference Books in Operating System.

To practice all areas of Operating System, [here is complete set on 1000+ Multiple Choice Questions and Answers on Operating System](#).

Participate in the Sanfoundry Certification [contest](#) to get free Certificate of Merit. Join our social networks below and stay updated with latest contests, videos, internships and jobs!

[Telegram](#) | [Youtube](#) | [LinkedIn](#) | [Instagram](#) | [Facebook](#) | [Twitter](#) | [Pinterest](#)

« [Prev - Operating System Questions and Answers – CPU Scheduling Benefits](#)

» [Next - Operating System Questions and Answers – CPU Scheduling Algorithms-2](#)

advertisement

## Recommended Posts:

1. [C Programming Examples on Combinatorial Problems & Algorithms](#)
2. [C++ Programming Examples on Computational Geometry Problems & Algorithms](#)
3. [C++ Programming Examples on Numerical Problems & Algorithms](#)
4. [C Programming Examples on Numerical Problems & Algorithms](#)
5. [Java Programming Examples on Combinatorial Problems & Algorithms](#)
6. [Java Programming Examples on Numerical Problems & Algorithms](#)
7. [Data Structures & Algorithms II – Questions and Answers](#)
8. [C Algorithms, Problems & Programming Examples](#)
9. [Java Programming Examples on Set & String Problems & Algorithms](#)
10. [C++ Programming Examples on Set & String Problems & Algorithms](#)
11. [C Programming Examples on Set & String Problems & Algorithms](#)
12. [Java Programming Examples on Hard Graph Problems & Algorithms](#)
13. [C++ Programming Examples on Graph Problems & Algorithms](#)
14. [Java Algorithms, Problems & Programming Examples](#)
15. [C++ Programming Examples on Hard Graph Problems & Algorithms](#)
16. [C++ Algorithms, Problems & Programming Examples](#)
17. [Java Programming Examples on Graph Problems & Algorithms](#)
18. [C Programming Examples on Hard Graph Problems & Algorithms](#)
19. [C Programming Examples on Graph Problems & Algorithms](#)
20. [Operating System Questions and Answers](#)

advertisement



[Manish Bhojasia](#), a technology veteran with 20+ years @ Cisco & Wipro, is Founder and CTO at Sanfoundry. He is Linux Kernel Developer & SAN Architect and is passionate about competency developments in these areas. He lives in Bangalore and delivers focused training sessions to IT professionals in Linux Kernel, Linux Debugging, Linux Device Drivers, Linux Networking, Linux Storage, Advanced C Programming, SAN Storage Technologies, SCSI Internals & Storage Protocols such as iSCSI & Fiber Channel. Stay connected with him @ [LinkedIn](#)

### Subscribe Sanfoundry Newsletter and Posts

Subscribe

[About](#) | [Certifications](#) | [Internships](#) | [Jobs](#) | [Privacy Policy](#) | [Terms](#) | [Copyright](#) | [Contact](#)



© 2011-2021 Sanfoundry. All Rights Reserved.