**SCHEDULING ALGORITHMS**

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

Ans: b)

1. 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

Ans: c)

1. 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

Ans: b)

1. 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

Ans: b)

1. The real difficulty with SJF in short term scheduling is \_\_\_\_\_\_\_\_\_\_\_\_  
   a) it is too good an algorithm  
   b) knowing the length of the next CPU request  
   c) it is too complex to understand  
   d) none of the mentioned

Ans: b)

6. Consider the following set of processes, the length of the CPU burst time given in milliseconds.

Process Burst time

P1 6

P2 8

P3 7

P4 3

Assuming the above process being scheduled with the SJF scheduling algorithm.  
a) The waiting time for process P1 is 3ms  
b) The waiting time for process P1 is 0ms  
c) The waiting time for process P1 is 16ms  
d) The waiting time for process P1 is 9ms

Ans: a)

7. Which of the following statements are true?

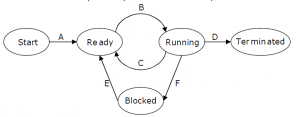
i) Shortest remaining time first scheduling may cause starvation

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

Ans: d)

8. In the following process state transition diagram for a uniprocessor system, assume that there are always some processes in the ready state: Now consider the following statements: [](http://www.geeksforgeeks.org/wp-content/uploads/gq/2014/01/gate2009.png)

I. If a process makes a transition D, it would result in

another process making transition A immediately.

II. A process P2 in blocked state can make transition E

while another process P1 is in running state.

III. The OS uses preemptive scheduling.

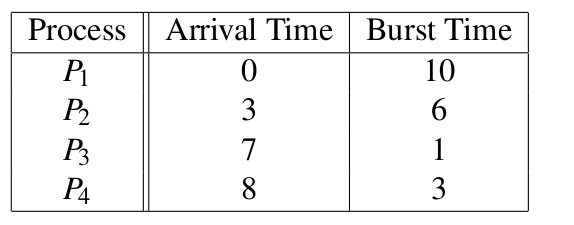
IV. The OS uses non-preemptive scheduling.

Which of the above statements are TRUE?

a)i and ii  
b) i and iii   
c) ii and iii   
d) i, ii and iv

Ans: c)

9. Consider the following processes, with the arrival time and the length of the CPU burst given in milliseconds. The scheduling algorithm used is preemptive shortest remaining-time first.

[](https://media.geeksforgeeks.org/wp-content/cdn-uploads/gq/2016/02/z5.png)  
The average turn around time of these processes is \_\_\_\_\_\_\_\_\_\_\_ milliseconds.  
   
Note : This question was asked as Numerical Answer Type.  
**(A)** 8.25  
**(B)** 10.25  
**(C)** 6.35  
**(D)** 4.25  
  
  
**Answer:** **(A)**

10. In which of the following scheduling criteria, context switching will never take place ?

1. ROUND ROBIN
2. Preemptive SJF
3. Non-Preemptive SJF
4. Preemptive priority

Ans: c)

**CRITICAL SECTION**

1. A situation where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which access takes place is called \_\_\_\_\_\_\_\_\_\_\_\_  
a) data consistency  
b) race condition  
c) aging  
d) starvation

Ans: b)

2. Mutual exclusion implies that \_\_\_\_\_\_\_\_\_\_\_\_  
a) if a process is executing in its critical section, then no other process must be executing in their critical sections  
b) if a process is executing in its critical section, then other processes must be executing in their critical sections  
c) if a process is executing in its critical section, then all the resources of the system must be blocked until it finishes execution  
d) none of the mentioned

Ans: a)

3.  Bounded waiting implies that there exists a bound on the number of times a process is allowed to enter its critical section \_\_\_\_\_\_\_\_\_\_\_\_  
a) after a process has made a request to enter its critical section and before the request is granted  
b) when another process is in its critical section  
c) before a process has made a request to enter its critical section  
d) none of the mentioned

Ans: a)

4. What are Spinlocks?  
a) CPU cycles wasting locks over critical sections of programs  
b) Locks that avoid time wastage in context switches  
c) Locks that work better on multiprocessor systems  
d) All of the mentioned

Ans: d)

5. The following program consists of 3 concurrent processes and 3 binary semaphores. The semaphores are initialized as S0 = 1, S1 = 0, S2 = 0.

Process P0

while(true)

{

wait(S0);

print '0';

release(S1);

release(S2);

}

Process P1

wait(S1);

release(S0);

Process P2

wait(S2);

release(S0);

How many times will P0 print ‘0’?  
a) At least twice  
b) Exactly twice  
c) Exactly thrice  
d) Exactly once

Ans: a)

6. Two processes, P1 and P2, need to access a critical section of code. Consider the following synchronization construct used by the processes.

Process P1 :

while(true)

{

w1 = true;

while(w2 == true);

Critical section

w1 = false;

}

Remainder Section

Process P2 :

while(true)

{

w2 = true;

while(w1 == true);

Critical section

w2 = false;

}

Remainder Section

Here, w1 and w2 have shared variables, which are initialized to false. Which one of the following statements is TRUE about the above construct?  
a) It does not ensure mutual exclusion  
b) It does not ensure bounded waiting  
c) It requires that processes enter the critical section in strict alternation  
d) It does not prevent deadlocks but ensures mutual exclusion

Ans: d)

7. What is a mutex?  
a) is a binary mutex  
b) must be accessed from only one process  
c) can be accessed from multiple processes  
d) none of the mentioned

Ans: b)

8. The program follows to use a shared binary semaphore T.

Process A

int Y;

A1: Y = X\*2;

A2: X = Y;

signal(T);

Process B

int Z;

B1: wait(T);

B2: Z = X+1;

X = Z;

T is set to 0 before either process begins execution and, as before, X is set to 5.  
Now, how many different values of X are possible after both processes finish executing?  
a) one  
b) two  
c) three  
d) four

Ans: a)

9.  Semaphores are mostly used to implement \_\_\_\_\_\_\_\_\_\_\_\_  
a) System calls  
b) IPC mechanisms  
c) System protection  
d) None of the mentioned

Ans: b)

10. Spinlocks are intended to provide \_\_\_\_\_\_\_\_\_\_ only.  
a) Mutual Exclusion  
b) Bounded Waiting  
c) Aging  
d) Progress

Ans: b)

**DEADLOCKS**

1. Which of the following condition is required for deadlock to be possible?

1. mutual exclusion
2. a process may hold allocated resources while awaiting assignment of other resources
3. no resource can be forcibly removed from a process holding it
4. all of the mentioned

Ans: d)

2. A system is in the safe state if:

1. the system can allocate resources to each process in some order and still avoid a deadlock
2. there exist a safe sequence
3. both (a) and (b)
4. none of the mentioned

Ans: c)

3. To avoid deadlock:

1. there must be a fixed number of resources to allocate
2. resource allocation must be done only once
3. all deadlocked processes must be aborted
4. inversion technique can be used

Ans: a)

4. To ensure no preemption, if a process is holding some resources and requests another resource that cannot be immediately allocated to it :

1. then the process waits for the resources be allocated to it
2. the process keeps sending requests until the resource is allocated to it
3. the process resumes execution without the resource being allocated to it
4. then all resources currently being held are preempted

Ans: d)

5.  What is the drawback of banker’s algorithm?  
a) in advance processes rarely know how much resource they will need  
b) the number of processes changes as time progresses  
c) resource once available can disappear  
d) all of the mentioned

Ans: d)

6. The number of resources requested by a process \_\_\_\_\_\_\_\_\_\_\_\_  
a) must always be less than the total number of resources available in the system  
b) must always be equal to the total number of resources available in the system  
c) must not exceed the total number of resources available in the system  
d) must exceed the total number of resources available in the system

Ans: c)

7.  Which of the following is/are the restrictions on deadlock avoidance.  
i) The maximum resource requirement for each process must be stated in advance  
ii) There must be fixed number of resources to allocate  
iii) No process may exit while holding resources.

1. i and ii only
2. ii and iii only
3. i and iii only
4. All i, ii and iii

Ans: d)

8. If we preempt a resource from a process, the process cannot continue with its normal execution and it must be \_\_\_\_\_\_\_\_\_\_\_\_  
a) aborted  
b) rolled back  
c) terminated  
d) queued

Ans: b)

**SWAPPING**

1. The run time mapping from virtual to physical addresses is done by a hardware device called the \_\_\_\_\_\_\_\_\_\_\_\_  
a) Virtual to physical mapper  
b) Memory management unit  
c) Memory mapping unit  
d) None of the mentioned

Ans: b)

2. The size of a process is limited to the size of \_\_\_\_\_\_\_\_\_\_\_\_  
a) physical memory  
b) external storage  
c) secondary storage  
d) none of the mentioned

Ans: a)

3. The backing store is generally a \_\_\_\_\_\_\_\_\_\_\_\_  
a) fast disk  
b) disk large enough to accommodate copies of all memory images for all users  
c) disk to provide direct access to the memory images  
d) all of the mentioned

Ans: d)

4. The \_\_\_\_\_\_\_\_\_ time in a swap out of a running process and swap in of a new process into the memory is very high.  
a) context – switch  
b) waiting  
c) execution  
d) all of the mentioned

Ans: a)

5. Paging \_\_\_\_\_\_\_\_\_

1. solves the memory fragmentation problem
2. allows modular programming
3. avoids deadlock
4. allows structured programming

Ans: a)

6. Thrashing occurs \_\_\_\_\_\_\_\_.

1. when excessive swapping takes place
2. when you thrash your computer
3. whenever deadlock occurs
4. when no swapping takes place

Ans: a)

7. Which scheduler performs the "swapping out" or "swapping in"?

1. Long-term scheduling
2. Medium-term scheduling
3. Short-term scheduling
4. None of the above

Ans: b)

8. State true of false.  
  
i) With paging, each process is divided into relatively small, fixed-size pages.  
ii) Segmentation provides for the use of pieces of varying size.

1. True, False
2. True, True
3. False,True
4. False, False

Ans: b)

9. In a system that does not support swapping,

1. the compiler normally binds symbolic addresses (variables) to relocatable addresses
2. the compiler normally binds symbolic addresses to physical addresses
3. the loader binds relocatable addresses to physical addresses
4. binding of symbolic addresses to physical addresses normally takes place during execution

Ans: a)

**DISK MANAGEMENT & DISK SCHEDULING**

1.The time taken to move the disk arm to the desired cylinder is called the \_\_\_\_\_\_\_\_\_\_\_\_  
a) positioning time  
b) random access time  
c) seek time  
d) rotational latency

Ans: c)

2. What is the host controller?  
a) controller built at the end of each disk  
b) controller at the computer end of the bus  
c) all of the mentioned  
d) none of the mentioned

Ans: b)

3. In the \_\_\_\_\_\_ algorithm, the disk arm starts at one end of the disk and moves toward the other end, servicing requests till the other end of the disk. At the other end, the direction is reversed and servicing continues.  
a) LOOK  
b) SCAN  
c) C-SCAN  
d) C-LOOK

Ans: b)

4.  The \_\_\_\_\_\_\_ program initializes all aspects of the system, from CPU registers to device controllers and the contents of main memory, and then starts the operating system.  
a) main  
b) bootloader  
c) bootstrap  
d) rom

Ans: b)

5. \_\_ program initializes all aspects of the system, from CPU registers to device controllers and the contents of main memory, and then starts the operating system.

1. Main
2. Bootloader
3. Bootstrap
4. rom

Ans: c)

**PAGING**

1. **Consider Peterson’s algorithm for mutual exclusion between two concurrent processes i and j. The program executed by process is shown below.**

**repeat**

**flag [i] = true;**

**turn = j;**

**while ( P ) do no-op;**

**Enter critical section, perform actions, then exit critical**

**section**

**flag [ i ] = false;**

**Perform other non-critical section actions.**

**until false;**

**For the program to guarantee mutual exclusion, the predicate P in the while loop should be (GATE 2001)**  
a) flag [j] = true and turn = i  
b) flag [j] = true and turn = j  
c) flag [i] = true and turn = j  
d) flag [i] = true and turn = i

**Answer:** (b)