BYJU'S EXAM PREP

GATE CS & IT

Operating system

PYQ Questions and Detailed Solution

1. In the context of operating system, which of the following statements is/are correct with respect to paging?

A. Paging helps solve the issue of external fragmentation.

B. Page size has no impact on internal fragmentation.

C. Multi-level paging is necessary to support pages of different sizes.

D. Paging incurs memory overheads

Ans. A, D

Sol. Pages are divided into fixed size slots , so no external fragmentation

But applications smaller than page size cause internal fragmentation

Page tables take extra pages in memory. Therefore incur extra cost

therefore A, D answer.

2. Which of the following standard C library functions will always invoke a system call when executed from a single-threaded process in a UNIX/Linux operating system?

A. strlen

B. sleep

C. malloc

D. exit

Ans.

Sol. Sleep (), exit () always invokes the system calls.

Strlen () and malloc () does not always invokes the system calls.

3. There are 6 jobs with distinct difficulty levels, and 3 computers with distinct processing speeds. Each job is assigned to a computer such that :

- The fastest computer gets the toughest job and the slowest computer gets the easiest job.
- Every computer gets at least one job

The number of ways in which this can be done is _____ .

Ans.

Sol. Let C1 be the fastest and C3 be the slowest computers.

These two are assigned two jobs. Now out of the remaining 4 jobs we need to ensure C2 gets at least 1. Without this constraint we can assign 4 jobs to 3 computers in $3^4=81$ ways. Out of these 81 ways $2^4=16$will be having no jobs for C2.

So, number of possible ways so that C2 gets at least one job $=81-16=65$.

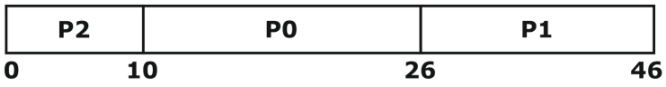
4. Three processes arrive at time zero with CPU bursts of 16, 20 and 10 milliseconds. If the scheduler has prior knowledge about the length of the CPU bursts, the minimum achievable average waiting time for these processes in a non-preemptive scheduler (rounded to nearest integer) is _____ milliseconds.

Ans.

Sol.

	AT	BT	WT
P0	0	16	10
P1	0	20	26
P2	0	10	0

Gantt chart :



Avg WT = {WT(P0) + WT(P1) + WT(P2)}/3 = {10 + 26 + 0}/3 = 36/3 = 12 ms

5. Consider the following pseudocode, where S is a semaphore initialized to 5 in line #2 and counter is a shared variable initialized to 0 in line#1 . Assume that the increment operation in line#7 is not atomic.

```
1) int counter = 0
2) Semaphore S = init(5);
3) void parop(void)
4) {
5) wait(S);
6) wait(S);
7) counter++;
8) signal(S);
9) signal(S);
10) }
```

If five threads execute the function parop concurrently, which of the following program behavior(s) is/are possible?

A. The value of counter is 5 after all the threads successfully complete the execution of parop.

B. The value of counter is 1 after all the threads successfully complete execution of parop.

C. The value of counter is 0 after all the threads successfully complete the execution of parop.

D. There is a deadlock involving all the threads.

Ans.

Sol. A. Assume the processes execute sequentially with no interleaving then after each process ends the counter value increments by 1, hence after 5 process 5 increments to zero hence a counter value of 5. Possible.

B. Let's assume that a process used 2 waits and reads the counter value and didn't update the value yet, all the other process let's say the other processes executed sequentially incremented and stored the value as 4 but since the value isn't written the first process yet the current value is overwritten by the first process as 1. Possible

C. There exists no pattern of execution in which the process increments the current value and completes while maintaining 0 as the counter value. Not possible

D. Assume that all the process use up the first wait operation, the semaphore value will now become zero and deadlock would've occurred. Possible

6. Which of the following statement(s) is/are correct in the context of CPU scheduling?

A. Turnaround time includes waiting time.

B. Round-robin policy can be used even when the CPU time required by each of the processes is not known apriori.

C. Implementing preemptive scheduling needs hardware support.

D. The goal is to only maximize CPU utilization and minimize throughput.

Ans.

Sol. option A is Correct, turnaround time = burst time + waiting

option B is correct

option C is correct, preemptive scheduling needs hardware supports such as timer.

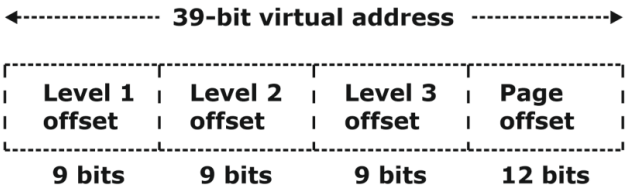
option D is incorrect, we have to maximize the throughput also.

7. Consider a three-level page table to translate a 39-bit virtual address to a physical address as shown below.

The page size is 4K8 (1KB = 2^{10} bytes) and page table entry size at every level is 8 bytes. A process P is currently using 2GB (1GB = 2^{30} bytes) virtual memory which is mapped to 2GB of physical memory. The minimum amount of memory required for the page table of P across all levels is _____ KB .

Ans.

Sol.



Since process P is using 2 GB physical memory and page size 2^{12} bytes.

So, Number of pages = $\frac{2^{31}}{2^{12}} = 2^{19}$

There are 2^9 entries per page table in 3rd level.

So, we need = $\frac{2^{19}}{2^9} = 2^{10}$ page tables in 3rd level

Now, it means 2^{10} entries in 2nd level so number of page tables in 2ndlevel

$= \frac{2^{10}}{2^9} = 2$ page table

Now we have 2 entries in 1st level hence we need only 1 page table in 1st level.

Total number of page table = 2^{10} + 2 + 1 = 1027 per table

Each page table has 2^9entries and each entry size is 8 bytes.

So, total size of page table in bytes = 1027 × 2^9× 8 bytes = 4108 KB

8. Which of the following statement(s) is/are correct in the context of CPU scheduling?

A. Turnaround time includes waiting time.

B. Round-robin policy can be used even when the CPU time required by each of the processes is not known apriori.

C. Implementing preemptive scheduling needs hardware support.

D. The goal is to only maximize CPU utilization and minimize throughput.

Ans.

Sol. option A is Correct, turnaround time = burst time + waiting

option B is correct

option C is correct, preemptive scheduling needs hardware supports such as timer.

option D is incorrect, we have to maximize the throughput also.

9. Consider the following multi-threaded code segment (in a mix of C and pseudocode), invoked by two processes P1 and P2, and each of the processes spawns two threads T1 and T2:

```
int x = 0;   // global
Lock L1;    // global
main() {
  create a thread to execute foo();  // Thread T1
  create a thread to execute foo();  // Thread T2
  wait for the two threads to finish execution;
  print (x);}

foo() {
  int y = 0;
  Acquire L1;
  x = x + 1;
  y = y + 1;
  Release L1;
  print (y);}
```

Which of the following statement(s) is / are correct?

A. At least one of P1 and P2 will print the value of x as 4.

B. At least one of the threads will print the value of y as 2.

C. Both T1 and T2, in both the processes, will print the value of y as 1.

D. Both P1 and P2 will print the value of x as 2.

Ans.

Sol. option C : correct

Threads don't share the stack. y is AUTO variable, stored in stack. So, each process spawns two threads, each locks and unlocks L1 as they increment x and y. Y is incremented from 0 to 1 by each thread of a process.

If P1 completely executes and then P2 completely executes without any preemption, then, x is incremented twice (once by each thread). Global variable x is stored in the data segment. Threads share data segments.

option D : Correct

If P1 and P2 executes in any order, x will be 2 printed by each process.

10. Consider a computer system with multiple shared resource types, with one instance per resource type. Each instance can be owned by only one process at a time. Owning and freeing of resources are done by holding a global lock (L). The following scheme is used to own a resource instance:

```
function OWNRESOURCE(Resource R)
        Acquire lock L // a global lock
        if R is available then
            Acquire R
            Release lock L
        else
            if R is owned by another process P then
                Terminate P, after releasing all resources owned by P
                Acquire R
                Restart P
                Release lock L
                end if
            end if
        end if
    end function
```

Which of the following choice(s) about the above scheme is/are correct?

A. The scheme violates the mutual exclusion property.

B. The scheme ensures that deadlocks will not occur.

C. The scheme may lead to starvation.

D. The scheme may lead to live-lock.

Ans.

Sol. The lock ensures that mutual exclusion is enforced

Since forceful preemption of resources by termination of the other processes is allowed, deadlock will never occur.

Say when process P is holding the single instance resource of type R1, after some time, process Q requests for resource type R1. Using the OWNRESOURCE algorithm, process Q will be able to terminate process P and thus releasing all the resources held by process P. Now Q acquires (holds) resource type R1 while process P gets restarted. And then process Q releases lock L.

It may lead to a live lock, where each process continues to preempt each other, thus none of them can make any progress but neither are in a blocked state.

Livelock occurs when two or more processes continually repeat the same interaction in response to changes in the other processes without doing any useful work. These processes are not in the waiting state, and they are running concurrently. This is different from a deadlock because in a deadlock all processes are in the waiting state.

11. Consider allocation of memory to a new process. Assume that none of the existing holes in the memory will exactly fit the process's memory requirement. Hence, a new hole of smaller size will be created if allocation is made of the existing holes. Which one of the following statements is TRUE?

A. The hole created by the best fit is never larger than the hole created by first fit

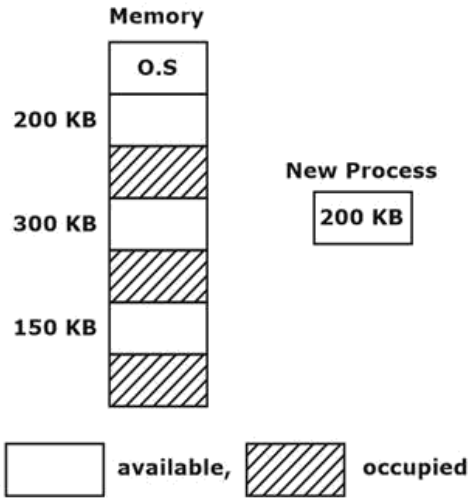
B. The hole created by the best fit is always larger than the hole created by next fit

C. The hole created by the next fit is never larger than the hole created by best fit

D. The hole created by worst fit is always larger than the hole created by first fit

Ans. A

Sol. Let us consider a new process required 120 kb memory and existing holes in the memory are 200, 300, 150 kb as shown in the diagram in the same order.



Now, when we allocate, this new process a memory using different algorithms, it would be like given below

Algorithm Allocated partition size of new tube

First Fit 200 KB 80 KB

Best Fit 150 KB 30 KB

Worst Fit 300 KB 180 KB

Next Fit 300 KB 180 KB

12. Consider the following statements about process state transitions for a system using preemptive scheduling.

I. A running process can move to ready state.

II. A ready process can move to running state.

III. A blocked process can move to running state.

IV. A blocked process can move to ready state.

Which of the above statements are TRUE?

A. I, II, III, and IV

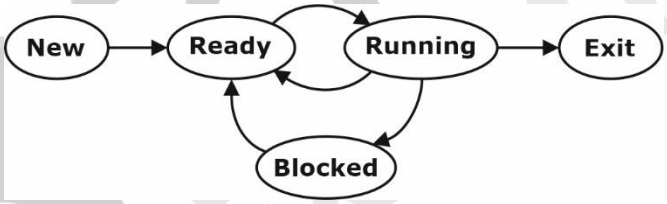
B. I, II, and IV only

C. I, II, and III only

D. II and III only

Ans. B

Sol.



1,2,4 are True

13. Consider the following set of processes, assumed to have arrived at time Consider the CPU scheduling algorithms Shortest Job First (SJF) and Round Robin (RR). For RR, assume that the processes are scheduled in the order P_1, P_2, P_3, P_4.

Processes	P_1	P_2	P_3	P_4
Burst time (in ms)	8	7	2	4

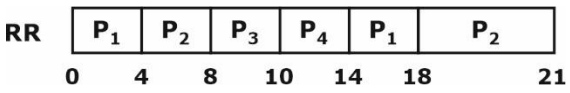
If the time quantum for RR is 4 ms, then the absolute value of the difference between the average turnaround times (in ms) of SJF and RR (round off t0 2 decimal places) is …….. .

Ans.

Sol. Average turnaround time

$P_1 = 21$, $P_2 = 13$, $P_3 = 2$, $P_4 = 6$

$$\text{Avg.} = \frac{21+13+2+6}{4} = \frac{42}{4} = 10.5$$



Average turnaround time

$P_1 = 18, P_2 = 21, P_3 = 10, P_4 = 14$

$$\text{Avg.} = \frac{18+21+10+14}{4} = \frac{64}{4} = 15.75$$

Difference = 15.75 – 10.5 = 5.25

14. Each of a set of n processes executes the following code using two semaphores a and b initialized to 1 and 0, respectively. Assume that *count* is a shared variable initialized to 0 and not used in CODE SECTION P.

```
CODE SECTION P
  wait (a); count=count+1;
  If (count==n) signal (b);
  signal (a); wait (b); signal (b);
CODE SECTION Q
```

What does the code achieve?

A. It ensures that all processes execute CODE SECTION P mutually exclusively.

B. It ensures that at most n–1 processes are in CODE SECTION P at any time.

C. It ensures that no process executes CODE SECTION Q before every process has finished CODE SECTION P.

D. It ensures that at most two processes are in CODE SECTION Q at any time.

Ans. C

Sol. Subject Operating Systems

a = 1 b = 0 count = 0

Code section P: For process 1

Wait (a) ⇒ a = 0

Count ++ ⇒ count = 1

If (count = = n) ⇒ false

signal (a) ⇒ a = 0

wait (b) process blocked

As shown above, all processes execute wait (b) before, signal (b) in if condition.

Because, 'if' condition becomes True only for process P_n(n^{th} process) [∵ if (count = = n)]

So, before Pn process finishes the code section, P, no other process executes code section Q.

Hence, no process executes code section Q before every process has finished code section P.

15. Consider the following five disk access requests of the form (request id, cylinder number) that are present in the disk scheduler queue at a given time.

(P, 155), (Q, 85), (R, 110), (S, 30), (T, 115)

Assume the head is positioned at cylinder 100. The scheduler follows Shortest Seek Time First scheduling to service the requests.

Which one of the following statements is FALSE?

A. T is serviced before P.

B. The head reverses its direction of movement between servicing of Q and P

C. R is serviced before P.

D. Q is serviced after S, but before T.

Ans. D

Sol. Given P : 155

Q : 85

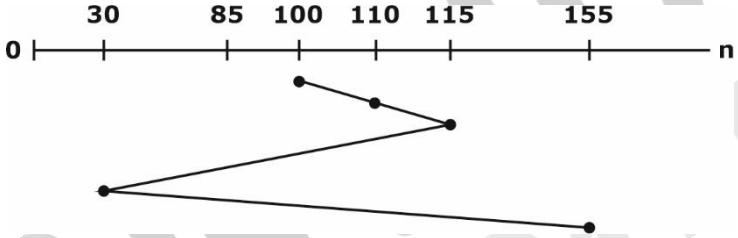
R : 110

S : 30

T : 115

Current head position = 100

SSTF = algorithm



From the above sequence, we can say that option D is correct.

16. Consider a paging system that uses a 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns. TLB lookup takes 20 ns. Each page transfer to/from the disk takes 5000 ns. Assume that the TLB hit ratio is 95%, page fault rate is 10%. Assume that for 20% of the total page faults, a dirty page has to be written back to disk before required page is read in from disk. TLB update time is negligible. The average memory access time is ns (round off to 1 decimal places) is _____.

Ans.

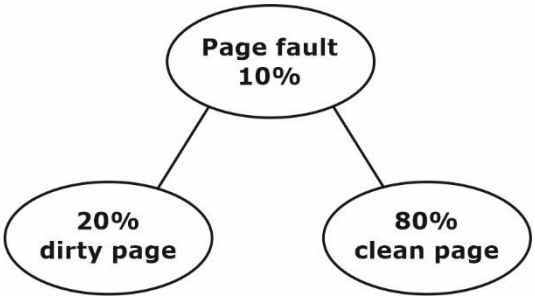
Sol. Main memory access time, T_m = 100 ns

TLB lookup, T_{TLB} = 20ns

Page transfer time, T_{PT} = 5000 ns

TLB hit ratio, x = 0.95 (95%)

page fault rate, p = 0.10 (10%)



We know,

EMAT for multilevel paging,

EMAT = x (T_c + T_m) + (1 − x) (T_c + (n + 1) T_m)

EMAT, when there is a page fault, S → is service time

EMAT = (1 − P) T_m + Ps

Here, we are using TLB, and page fault occurs whenever there is a miss in TLB, So the required EMAT is ,

EMAT = x(T_{tlb} + T_m) + (1 − x) [(1 − P) (T_{tlb} + T_m + T_m) + p(% dirty (T_{tlb} + T_m + 2T_{PT}) + % clean (T_{tlb} + T_m + T_{PT})

∴ EMAT = 0.95 (20 + 100) + 0.05 (0.9 (20 + 100 + 100) + 0.1 (0.2 (20 + 100 + 2(5000)) + 0.8 (20 + 100 + 5000))

= 154.5 ns

17. Consider three concurrent processes, P1, P2 and P3, as shown below, which access a shared variable D that has been initialised to 100

P1	P2	P3
:	:	:
:	:	:
D = D + 20	D = D − 50	D = D + 10
:	:	:
:	:	:

The processes are executed on a uniprocessor system running a time-shared operating system. If the minimum and maximum possible values of D after the three processes have completed execution are X and Y, respectively, then the value of Y − X is _____.

Ans.

Sol. Total possible execution sequences are P1 P2 P3, P1 P3 P2, P2 P1 P3, P2 P3 P1, P3 P1 P2 and P3 P2 P1.

Whatever the sequence we will use to execute, we will get

X= 50 and Y= 130

Hence, X-Y= 80

18. The following C program is executed on a Unix/Linux system:

```
#include
int main()
{
int i;
for (i=0; i<10; i++)
if (i%2 == 0) fork();
return 0;
}
```

The total number of child processes created is _____.

Ans.

Sol. Since the if condition will be validated for 6 times as, i=0, 2, 4, 6, 8.

So the number of child processes created is: 2^n-1.

So total processes created are $=2^5-1= 31$.

19. Assume that in a certain computer, the virtual addresses are 64 bits long and the physical addresses are 48 bits long. The memory is word addressable. The page size is 8 kB and the word size is 4 bytes. The Translation Look-aside Buffer (TLB) in the address translation path has 128 valid entries. At most how many distinct virtual addresses can be translated without any TLB miss?

A. 16×2^{10}

B. 256×2^{10}

C. 4×2^{20}

D. 8×2^{20}

Ans. B

Sol. 1 word = 4 bytes

Page size = 8 kB = 2^{13} B

Number of words in 1 page $= \frac{2^{13}}{2^2} = 2^{11}$

TLB can hold 128 valid entries so, at most 128×2^{11} memory address can be addressed without TLB miss.

$128 \times 2^{11} = 256 \times 2^{10}$

20. Consider the following four processes with arrival times (in milliseconds) and their length of CPU bursts (in milliseconds) as shown below:

Process	P1	P2	P3	P4
Arrival time	0	1	3	4
CPU burst time	3	1	3	Z

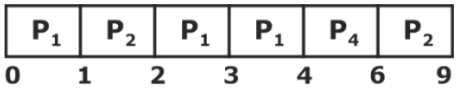
These processes are run on a single processor using a preemptive Shortest Remaining Time First scheduling algorithm. If the average waiting time of the processes is 1 millisecond, then the value of Z is_____.

Ans.

Sol. With hit and trial approach

At, Z = 1, the situation is not satisfied.

Let's assume, Z = 2



	Arrival time	CPU time	Completion time	Waiting time
P1	0	3	4	1
P2	1	1	2	0
P3	3	3	9	3
P4	4	Z = 2	0	0

$$\text{Average waiting time (WT)} = \frac{1+0+3+0}{4} = 1 \text{ ms}$$

Hence, Z = 2

So, the correct value Z would be 2.

21. The index node (inode) of a Unix-like file system has 12 direct, one single-indirect and one double-indirect pointers. The disk block size is 4 kB, and the disk block address is 32-bits long. The maximum possible file size is (rounded off to 1 decimal place) _____ GB.

Ans.

Sol. Given 12 direct, 1 single indirect, 1 double indirect pointers

Size of Disk block = 4kB

Disk Block Address = 32 bit = 4B

Number of addresses = Size of disk block/address size = 4KB/4B = 2^{10}

Maximum possible file size = 12 * 4kB + 2^{10} * 4kB + 2^{10} * 2^{10} * 4kB

= 4.00395 GB ≃ 4 GB

Hence 4GB is the correct answer

22. Consider the following snapshot of a system running n concurrent processes. Process i is holding Xi instances of a resource R, $1 \leq i \leq n$. Assume that all instances of R are currently in use. Further, for all i, process i can place a request for at most Yi additional instances of R while holding the Xi instances it already has. Of the n processes, there are exactly two processes p and q such that $Yp = Yq = 0$. Which one of the following conditions guarantees that no other process apart from p and q can complete execution?

A. $X_p + X_q <$ Min $\{Y_k \mid 1 \leq k \leq$,k ≠ p, k ≠ q$\}$

B. $X_p + X_q <$ Max $\{Y_k \mid 1 \le k \le$,k ≠ p, k ≠ q$\}$

C. Min (X_p, X_q) ≥ Min $\{Y_k \mid 1 \le k \le n$, k ≠ p, k ≠ q$\}$

D. Min (X_p, X_q) ≤ Max $\{Y_k \mid 1 \le k \le n$, k ≠ p, k ≠ q$\}$

Ans. A

23. Consider a process executing on an operating system that uses demand paging. The system's average time for memory access is M units if the corresponding memory page is available in memory and D units if the memory access causes a page fault. It has been experimentally measured that the average time taken for memory access in the process is X units. Which one of the following is the correct expression for the page fault rate experienced by the process?

A. (D – M)/(X – M)

B. (X – M)/(D – M)

C. (D – X)/(D – M)

D. (X – M)/(D – X)

Ans. B

Sol. Given,

Average time for a memory access = **M units** if page hits

Average time for a memory access = **D units** if page fault occurred.

Total/experimental average time taken for a memory access = **X units.**

Let page fault rate= **P**.

Therefore,

Effective memory access time = Page fault rate * Memory access time when page fault +(1 – page fault rate) * memory access time when no page fault

EMAT = P*D +(1-P)*M

X= P*D +(1-P)*M

X= P*D + M-P*M

X-M = P(D-M)

P= (X-M)/(D-M)

So, option B is the correct answer.

24. Consider a system with 3 processes that share 4 instances of the same resource type. Each process can request a maximum of K instances. Resource instances can be requested and released only one at a time. The largest value of K that will always avoid deadlock is _____.

Ans.

Sol.

P_1 P_2 P_3

| | | No deadlock

|

Maximum each process can request for 2 resources so, that there will not be any deadlock, because we, have only 4 resource available.

So, K value is '2'.

25. Consider the following solution to the producer-consumer synchronization problem. The shared buffer size is N. Three semaphores empty, full and mutex are defined with respective initial values of 0, N and 1. Semaphore empty denotes the number of available slots in the buffer, for the consumer to read from. Semaphore full denotes the number of available slots in the buffer, for the producer to write to. The placeholder variables, denoted by P, Q, R and S, in the code below can be assigned either empty or full. The valid semaphore operations are: wait ( ) and signal ( ).

Producer	Consumer

do{
wait (P);
wait (mutex);
//Add item to buffer
Signal (mutex);
Signal (Q);
} while (1);
	do{
	wait (R);
	wait (mutex);
	//Consume item from buffer
	Signal (mutex);
	Signal (S);
	} while (1);

Which one of the following assignments to P, Q, R and S will yield the correct solution?

A. P : full, Q: full, R: empty, S; empty

B. P: empty, Q: empty, R: full, S: full

C. P: full, Q: empty, R: empty, S: full

D. P: empty, Q: full, R: full, S: empty

Ans. C

Sol. Full = N, empty = 0, mutex = 1

Initially buffer will be empty, so consumer should not start first, so option b, D are eliminated.

With option A consumer will never consume the item, so it is wrong.

Option 'c' is correct answer which proper functionality of produce and consumer.

26. In a system, there are three types of resources: E, F and G. Four processes P_0, P_1, P_2 and P_3 execute concurrently. At the outset, the processes have declared their maximum resource requirements using a matrix named Max as given below. For example Max[P_2, F] is the maximum number of instances of F that P_2 would require. The number of instances of the resources allocated to the various processes at any given state is given by a matrix named allocation.

Consider a state of the system with the Allocation matrix as shown below and in which 3 instances of E and 3 instances of F are the only resources available.

Allocation			
	E	F	G
P_0	1	0	1
P_1	1	1	2
P_2	1	0	3
P_3	2	0	0

Max			
	E	F	G
P_0	4	3	1
P_1	2	1	4
P_2	1	3	3
P_3	5	4	1

From the perspective of deadlock avoidance, which one of the following is true?

A. The system is in safe state.

B. The system is not in safe state, but would be safe if one more instance of E were available.

C. The system is not in safe state, but would be safe if one more instance of F were available.

D. The system is not in safe state, but would be safe if one more instance of G were available.

Ans. A

Sol.

	Max need			**Current Allocation**			**Current available**			**Remaining need**		
	E	**F**	**G**	**E**	**F**	**G**	**E(3)**	**F(3)**	**G(0)**	**E**	**F**	**G**
P_0	4	3	1	1	0	1	4	3	1	3	3	0
P_1	2	1	4	1	1	2	5	3	4	1	0	2
P_2	1	3	3	1	0	3	6	4	6	0	3	0
P_3	5	4	1	2	0	0	8	4	6	3	4	1

Safe sequence : P_0, P_2, P_1, P_3

Safe state

27. Consider a storage disk with 4 platters (numbered as 0, 1, 2 and 3), 200 cylinders (numbered as 0, 1, …., 199) and 256 sectors per track (numbered as 0,1, ….., 255). The following 6 disk requests of the form [sector number, cylinder number, platter number] are received by the disk controller at the same time:

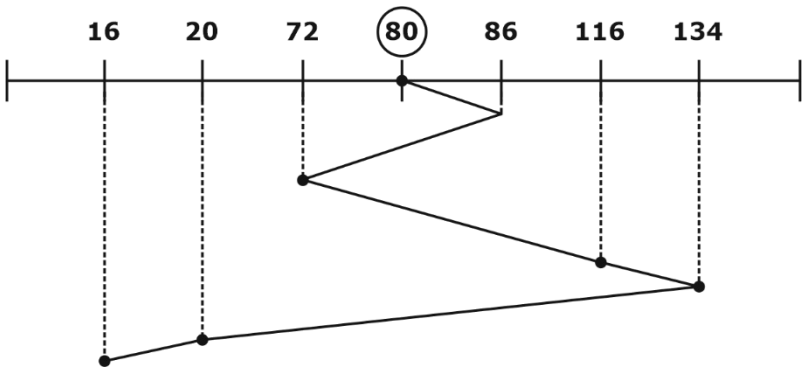
[120, 72, 2], [180, 134, 1], [60, 20, 0], [60, 20, 0], [212, 86, 3], [56, 116, 2], [118, 16, 1]

Currently the head is positioned at sector number 100 of cylinder 80 and is moving towards higher cylinder numbers. The average power dissipation in moving the head over 100 cylinders is 20 milliwatts and for reversing the direction of the head movement once is 15 milliwatts. Power dissipation associated with rotational latency and switching of head between different platters is negligible.

The total power consumption in milliwatts to satisfy all of the above disk requests using the Shortest Seek Time First disk scheduling algorithm is _____.

Ans.

Sol.



$(86-80)+(86-72)+(134-72)+$
$(134-16)+62+118+14=200$
$100 \rightarrow 20$
$200 \rightarrow ?$

$$\frac{200}{100}\times 20=40$$

3 direction changes $3\times 15=45$

$$40+45=85$$

28. Threads of a process share

A. global variable but not heap

B. heap but not global variables

C. neither global variables nor heap

D. Both heap and global variables

Ans. D

Sol. Threads of a process can share all resources except stack and register set.

29. Consider the following CPU processes with arrival times (in milliseconds) and length of CPU bursts (in milliseconds) as given below:

Process	Arrival Time	Burst Time
P1	0	7
P2	3	3
P3	5	5
P4	6	2

If the pre-emptive shortest remaining time first scheduling algorithm is used to schedule the processes, then the average waiting time across all processes is _____ milliseconds.

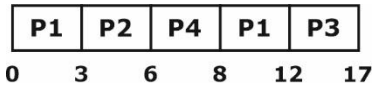
A. 0 B. 1

C. 2 D. 3

Ans. D

Sol.

Process	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	0	7	12	12	5
P2	3	3	6	3	0
P3	5	5	17	12	7
P4	6	2	8	2	0
					Total: 12



TAT = CT – AT

WT = TAT – BT

Total waiting time = 12

Average Waiting time = 12/4 = 3

30. A multithreaded program P executes with x number of threads and uses y number of locks for ensuring mutual exclusion while operating on shared memory locations. All locks in the program are non-re-entrant, i.e., if a thread holds a lock l, then it cannot re-acquire lock l without releasing it. If a thread is unable to acquire a lock, it blocks until the lock becomes available. The minimum value of x and the minimum value of y together for which execution of P can result in a deadlock are:

A. x = 1, y = 2

B. x = 2, y = 1

C. x = 2, y = 2

D. x = 1, y = 1

Ans. D

Sol. Reentrant property is provided, so that a process who owns a lock, can acquire same lock multiple times. Here it is non-reentrant as given, process cant own same lock multiple times. So if a thread tries to acquire already owned lock, will get blocked, and this is a deadlock.

Hence correct answer is x=1, y=1.

****

Our Outstanding GATE Results

100+ Students under AIR-100 2021


Rank 02
Poojasree (EC)


Rank 03
Manoj (EC)


Rank 03
Munish (ME)


Rank 06
Parag (EC)


Rank 07
Vatsal (ME)


Rank 08
Rahul (CE)

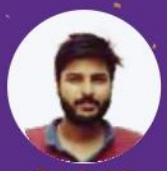

Rank 08
Hemant (EE)


Rank 08
Rajat (ME)


Rank 09
Vamsi (EC)


Rank 13
Shashwat (CE)

50+ Students under AIR-100 2020


Rank 06
Ghanendra (EC)


Rank 09
Avinash (ME)


Rank 09
Himanshu (EE)


Rank 26
Kartikay (CE)


Rank 30
Raja (EC)


Rank 39
Akash Singh (CS)


Rank 39
Navneet (CS)


Rank 54
Rishi (EE)


Rank 56
Apurv Mittal (ME)


Rank 56
Nikhil (ME)

Our Success Stories

Our Outstanding GATE Results

80+
Students under
AIR-100
2022


Rank 01
Ram (EC)


Rank 02
Amit (CE)

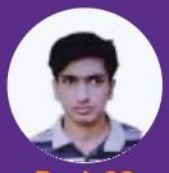

Rank 02
Vandit (EE)


Rank 03
Parvinder (ME)


Rank 06
Abhishek (ME)


Rank 06
Vivek (CE)


Rank 08
Kiran (CS)


Rank 09
Shivam (EC)


Rank 09
Tushar (EE)


Rank 10
Mitesh (CS)


Rank 11
Kaustav (ME)


Rank 11
Souvik (ME)


Rank 12
Himanshu (CE)


Rank 15
Surya (CS)


Rank 18
Swastik (CE)


Rank 20
Lakshay (ME)


Rank 21
Tathagata (CS)


Rank 22
Ayush (CE)


Rank 29
Piyush (EE)


Rank 38
Rajat (EE)