5 Operating System (305)

System calls, Processes, Threads, Inter-process communication, Concurrency and synchronization. Deadlock. CPU scheduling. Memory management and Virtual memory. File systems. Disks is also under this

Mark Distribution in Previous GATE

Year	2021-1	2021-2	2020	2019	2018	2017-1	2017-2	2016-1	2016-2	Minimum	Average	Maximum
1 Mark Count	4	2	2	2	3	2	2	1	1	1	2.1	4
2 Marks Count	1	3	4	4	3	2	2	4	3	1	2.8	4
Total Marks	6	8	10	10	9	6	6	9	7	**6**	**7.8**	**10**

5.1 Context Switch (3) top

5.1.1 Context Switch: GATE CSE 1999 | Question: 2.12 top

https://gateoverflow.in/1490

Which of the following actions is/are typically not performed by the operating system when switching context from process A to process B?

A. Saving current register values and restoring saved register values for process B.
B. Changing address translation tables.
C. Swapping out the memory image of process A to the disk.
D. Invalidating the translation look-aside buffer.

gate1999 operating-system context-switch normal

Answer

5.1.2 Context Switch: GATE CSE 2000 | Question: 1.20, ISRO2008-47 top

https://gateoverflow.in/644

Which of the following need not necessarily be saved on a context switch between processes?

A. General purpose registers
B. Translation look-aside buffer
C. Program counter
D. All of the above

gate2000-cse operating-system easy isro2008 context-switch

Answer

5.1.3 Context Switch: GATE CSE 2011 | Question: 6, UGCNET-June2013-III: 62 top

https://gateoverflow.in/2108

Let the time taken to switch from user mode to kernel mode of execution be $T1$ while time taken to switch between two user processes be $T2$. Which of the following is correct?

A. $T1 > T2$
B. $T1 = T2$
C. $T1 < T2$
D. Nothing can be said about the relation between $T1$ and $T2$

gate2011-cse operating-system context-switch easy ugcnetjune2013iii

Answer

Answers: Context Switch

5.1.1 Context Switch: GATE CSE 1999 | Question: 2.12 top

https://gateoverflow.in/1490

✔ Processes are generally swapped out from memory to Disk (secondary memory) when they are suspended. So. Processes are not swapped during context switching.

TLB : Whenever any page table entry is referred for the first time it is temporarily saved in TLB. Every element of this memory has a tag. And whenever anything is searched it is compared against TLB and we can get that entry/data with less memory access.

And Invalidation of TLB means resetting TLB which is necessary because a TLB entry may belong to any page table of any process thus resetting ensures that the entry corresponds to the process that we are searching for.

Hence, option (**C**) is correct.

90 votes

-- Manish Joshi (20.5k points)

5.1.2 Context Switch: GATE CSE 2000 | Question: 1.20, ISRO2008-47 top

https://gateoverflow.in/644

Answer: (**B**)

We don't need to save TLB or cache to ensure correct program resumption. They are just bonus for ensuring better performance. But PC, stack and registers must be saved as otherwise program cannot resume.

52 votes

-- Rajarshi Sarkar (27.9k points)

5.1.3 Context Switch: GATE CSE 2011 | Question: 6, UGCNET-June2013-III: 62 top

https://gateoverflow.in/2108

Time taken to switch two processes is very large as compared to time taken to switch between kernel and user mode of execution because :

When you switch processes, you have to do a context switch, save the PCB of previous process (note that the PCB of a process in Linux has over 95 entries), then save registers and then load the PCB of new process and load its registers etc.

When you switch between kernel and user mode of execution, OS has to just ***change a single bit*** at hardware level which is very fast operation.

So, answer is: (**C**).

116 votes

-- Mojo Jojo (2.8k points)

Context switches can occur only in kernel mode. So, to do context switch first switch from user mode to kernel mode and then do context switch (save the PCB of the previous process and load the PCB of new process)

Context switch = user - kernel switch + save/load PCB + kernel-user switch

C is answer.

79 votes

-- Sachin Mittal (15.8k points)

5.2 Deadlock Prevention Avoidance Detection (4) top

5.2.1 Deadlock Prevention Avoidance Detection: GATE CSE 2018 | Question: 24 top

https://gateoverflow.in/204098

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

gate2018-cse operating-system deadlock-prevention-avoidance-detection easy numerical-answers

Answer

5.2.2 Deadlock Prevention Avoidance Detection: GATE CSE 2018 | Question: 39 top

https://gateoverflow.in/204113

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 only resources available.

Allocation				Max			
	E	F	G		E	F	G
P_0	1	0	1	P_0	4	3	1
P_1	1	1	2	P_1	2	1	4
P_2	1	0	3	P_2	1	3	3
P_3	2	0	0	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

gate2018-cse operating-system deadlock-prevention-avoidance-detection normal

Answer

5.2.3 Deadlock Prevention Avoidance Detection: GATE CSE 2021 Set 2 | Question: 43 top

https://gateoverflow.in/357497

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 function
```

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

A. The scheme ensures that deadlocks will not occur
B. The scheme may lead to live-lock
C. The scheme may lead to starvation
D. The scheme violates the mutual exclusion property

gate2021-cse-set2 multiple-selects operating-system deadlock-prevention-avoidance-detection

Answer

5.2.4 Deadlock Prevention Avoidance Detection: GATE IT 2004 | Question: 63 top

https://gateoverflow.in/3706

In a certain operating system, deadlock prevention is attempted using the following scheme. Each process is assigned a unique timestamp, and is restarted with the same timestamp if killed. Let P_h be the process holding a resource R, P_r be a process requesting for the same resource R, and $T(P_h)$ and $T(P_r)$ be their timestamps respectively. The decision to wait or preempt one of the processes is based on the following algorithm.

```
if T(Pr) < T(Ph) then
    kill Pr
else wait
```

Which one of the following is TRUE?

A. The scheme is deadlock-free, but not starvation-free
B. The scheme is not deadlock-free, but starvation-free
C. The scheme is neither deadlock-free nor starvation-free
D. The scheme is both deadlock-free and starvation-free

gate2004-it operating-system normal deadlock-prevention-avoidance-detection

Answer

Answers: Deadlock Prevention Avoidance Detection

5.2.1 Deadlock Prevention Avoidance Detection: GATE CSE 2018 | Question: 24

https://gateoverflow.in/204098

✔ Number of processes $= 3$
Number of Resources $= 4$

Let's distribute each process one less than maximum demand $(K-1)$ resources. i.e. $3(K-1)$
Provide an additional resource to any of three processes for deadlock avoidance.

Total resources = $3(K-1)+1 = 3K-2$

Now, this 3K-2 should be less than or equal to the number of resources we have right now.
$3K-2 \leq 4$
$3K \leq 6$
$K \leq 2$
So, largest value of $K=2$

68 votes

-- Digvijay (44.9k points)

5.2.2 Deadlock Prevention Avoidance Detection: GATE CSE 2018 | Question: 39

https://gateoverflow.in/204113

✔

Allocation				Max				Need=Max-Allocation			
Process	E	F	G	Process	E	F	G	Process	E	F	G
P_0	1	0	1	P_0	4	3	1	P_0	3	3	0
P_1	1	1	2	P_1	2	1	4	P_1	1	0	2
P_2	1	0	3	P_2	1	3	3	P_2	0	3	0
P_3	2	0	0	P_3	5	4	1	P_3	3	4	1

Available Resource $(3,3,0)$
With $(3,3,0)$ we can satisfy the request of either P_0 or P_2.

Let's assume request of P_0 satisfied.
After execution, it will release resources.
Available Resource $= (3,3,0)+(1,0,1) = (4,3,1)$

Give $(0,3,0)$ out of $(4,3,1)$ unit of resources to P_2 and P_2 will completes its execution.
After execution, it will release resources.
Available Resource $= (4,3,1)+(1,0,3) = (5,3,4)$

Allocate $(1,0,2)$ out of $(5,3,4)$ unit of resources to P_1 and P_1 will completes its execution.
After execution, it will release resources.
Available Resource $= (5,3,4)+(1,1,2) = (6,4,6)$
And finally, allocate resources to P_3.

So, we have one of the possible safe sequence: $P_0 \longrightarrow P_2 \longrightarrow P_1 \longrightarrow P_3$

Correct Answer: A

18 votes

-- Digvijay (44.9k points)

5.2.3 Deadlock Prevention Avoidance Detection: GATE CSE 2021 Set 2 | Question: 43

https://gateoverflow.in/357497

✔ A system is in [Deadlock](#) when all the processes are in **Waiting** state. This is similar to a traffic jam where no vehicle moves.

A system is in Livelock when the processes do **repeated** work without any progress for the system (still no useful work). This is similar to a traffic jam where some vehicles reverses and then move forward hitting the same block again.

Now, both deadlock and livelock are **mutually exclusive** – at any point of time only one can happen in a system. But both of them imply no progress for system and hence starvation for the processes involved.

Now, coming to the given question, any process can kick out another process and then acquire the nedeed resource and this can go in a **cyclic fashion ensuring a livelock**. There is no possibility of a deadlock as at any time a process is free to kick out another process. Since there is a possibility of livelock, starvation possibility is also there. So, options A, B and C are TRUE.

A process is acquiring the resource owned by another process only after terminating the other process. Hence there is no violation of mutual exclusion property here.

Correct Answer: A;B;C.

References

👍 1 votes

-- Arjun Suresh (332k points)

5.2.4 Deadlock Prevention Avoidance Detection: GATE IT 2004 | Question: 63 top

https://gateoverflow.in/3706

✔ **Answer is (A).**

When the process wakes up again after it has been killed once or twice IT WILL HAVE SAME TIME-STAMP as it had WHEN IT WAS KILLED FIRST TIME. And that time stamp can never be greater than a process that was killed after that or a NEW process that may have arrived.

So every time when the killed process wakes up it MIGHT ALWAYS find a new process that will say "your time stamp is less than me and I take this resource", which of course is as we know, and that process will again be killed.

This may happen indefinitely if processes keep coming and killing that "INNOCENT" process every time it try to access.

So, **STARVATION is possible. Deadlock is not possible.**

👍 70 votes

-- Sandeep_Uniyal (6.5k points)

5.3 Disk Scheduling (13) top

5.3.1 Disk Scheduling: GATE CSE 1989 | Question: 4-xii top

https://gateoverflow.in/88222

Disk requests come to disk driver for cylinders $10, 22, 20, 2, 40, 6$ and 38, in that order at a time when the disk drive is reading from cylinder 20. The seek time is 6 msec per cylinder. Compute the total seek time if the disk arm scheduling algorithm is.

A. First come first served.
B. Closest cylinder next.

gate1989 descriptive operating-system disk-scheduling

Answer

5.3.2 Disk Scheduling: GATE CSE 1990 | Question: 9b top

https://gateoverflow.in/85678

Assuming the current disk cylinder to be 50 and the sequence for the cylinders to be $1, 36, 49, 65, 53, 12, 3, 20, 55, 16, 65$ and 78 find the sequence of servicing using

1. Shortest seek time first (SSTF) and
2. Elevator disk scheduling policies.

gate1990 descriptive operating-system disk-scheduling

Answer

5.3.3 Disk Scheduling: GATE CSE 1995 | Question: 20 top

https://gateoverflow.in/2658

The head of a moving head disk with 100 tracks numbered 0 to 99 is currently serving a request at track 55. If the queue of requests kept in FIFO order is

$$10, 70, 75, 23, 65$$

which of the two disk scheduling algorithms FCFS (First Come First Served) and SSTF (Shortest Seek Time First) will require less head movement? Find the head movement for each of the algorithms.

gate1995 operating-system disk-scheduling normal descriptive

Answer

5.3.4 Disk Scheduling: GATE CSE 1999 | Question: 1.10 top

https://gateoverflow.in/1463

Which of the following disk scheduling strategies is likely to give the best throughput?

A. Farthest cylinder next
B. Nearest cylinder next
C. First come first served
D. Elevator algorithm

gate1999 operating-system disk-scheduling normal

Answer

5.3.5 Disk Scheduling: GATE CSE 2004 | Question: 12 top

https://gateoverflow.in/1009

Consider an operating system capable of loading and executing a single sequential user process at a time. The disk head scheduling algorithm used is First Come First Served (FCFS). If FCFS is replaced by Shortest Seek Time First (SSTF), claimed by the vendor to give 50% better benchmark results, what is the expected improvement in the I/O performance of user programs?

A. 50%
B. 40%
C. 25%
D. 0%

gate2004-cse operating-system disk-scheduling normal

Answer

5.3.6 Disk Scheduling: GATE CSE 2009 | Question: 31 top

https://gateoverflow.in/1317

Consider a disk system with 100 cylinders. The requests to access the cylinders occur in following sequence:

$4, 34, 10, 7, 19, 73, 2, 15, 6, 20$

Assuming that the head is currently at cylinder 50, what is the time taken to satisfy all requests if it takes $1ms$ to move from one cylinder to adjacent one and shortest seek time first policy is used?

A. $95\ ms$
B. $119\ ms$
C. $233\ ms$
D. $276\ ms$

gate2009-cse operating-system disk-scheduling normal

Answer

5.3.7 Disk Scheduling: GATE CSE 2014 Set 1 | Question: 19 top

https://gateoverflow.in/1786

Suppose a disk has 201 cylinders, numbered from 0 to 200. At some time the disk arm is at cylinder 100, and there is a queue of disk access requests for cylinders $30, 85, 90, 100, 105, 110, 135$ and 145. If Shortest-Seek Time First (SSTF) is being used for scheduling the disk access, the request for cylinder 90 is serviced after servicing ____________ number of requests.

gate2014-cse-set1 operating-system disk-scheduling numerical-answers normal

Answer

5.3.8 Disk Scheduling: GATE CSE 2015 Set 1 | Question: 30 top

https://gateoverflow.in/8227

Suppose the following disk request sequence (track numbers) for a disk with 100 tracks is given:

$45, 20, 90, 10, 50, 60, 80, 25, 70.$

Assume that the initial position of the R/W head is on track 50. The additional distance that will be traversed by the R/W head when the Shortest Seek Time First (SSTF) algorithm is used compared to the SCAN (Elevator) algorithm (assuming that SCAN algorithm moves towards 100 when it starts execution) is________________tracks.

gate2015-cse-set1 operating-system disk-scheduling normal numerical-answers

Answer

5.3.9 Disk Scheduling: GATE CSE 2016 Set 1 | Question: 48 top

https://gateoverflow.in/39716

Cylinder a disk queue with requests for I/O to blocks on cylinders $47, 38, 121, 191, 87, 11, 92, 10.$ The C-LOOK scheduling algorithm is used. The head is initially at cylinder number 63, moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is__________.

gate2016-cse-set1 operating-system disk-scheduling normal numerical-answers

Answer

5.3.10 Disk Scheduling: GATE CSE 2020 | Question: 35 top

https://gateoverflow.in/333196

Consider the following five disk 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. Q is serviced after S,but before T.
C. The head reverses its direction of movement between servicing of Q and P.
D. R is serviced before P.

gate2020-cse operating-system disk-scheduling

Answer

5.3.11 Disk Scheduling: GATE IT 2004 | Question: 62 top

https://gateoverflow.in/3705

A disk has 200 tracks (numbered 0 through 199). At a given time, it was servicing the request of reading data from track 120, and at the previous request, service was for track 90. The pending requests (in order of their arrival) are for track numbers.

30 70 115 130 110 80 20 25.

How many times will the head change its direction for the disk scheduling policies SSTF(Shortest Seek Time First) and FCFS (First Come First Serve)?

A. 2 and 3
B. 3 and 3
C. 3 and 4
D. 4 and 4

gate2004-it operating-system disk-scheduling normal

Answer

5.3.12 Disk Scheduling: GATE IT 2007 | Question: 82 top

https://gateoverflow.in/3534

The head of a hard disk serves requests following the shortest seek time first (SSTF) policy. The head is initially positioned at track number 180.

Which of the request sets will cause the head to change its direction after servicing every request assuming that the head does not change direction if there is a tie in SSTF and all the requests arrive before the servicing starts?

A. $11, 139, 170, 178, 181, 184, 201, 265$
B. $10, 138, 170, 178, 181, 185, 201, 265$
C. $10, 139, 169, 178, 181, 184, 201, 265$
D. $10, 138, 170, 178, 181, 185, 200, 265$

gate2007-it operating-system disk-scheduling normal

Answer

5.3.13 Disk Scheduling: GATE IT 2007 | Question: 83 top

https://gateoverflow.in/3535

The head of a hard disk serves requests following the shortest seek time first (SSTF) policy.

What is the maximum cardinality of the request set, so that the head changes its direction after servicing every request if the total number of tracks are 2048 and the head can start from any track?

A. 9
B. 10
C. 11
D. 12

gate2007-it operating-system disk-scheduling normal

Answer

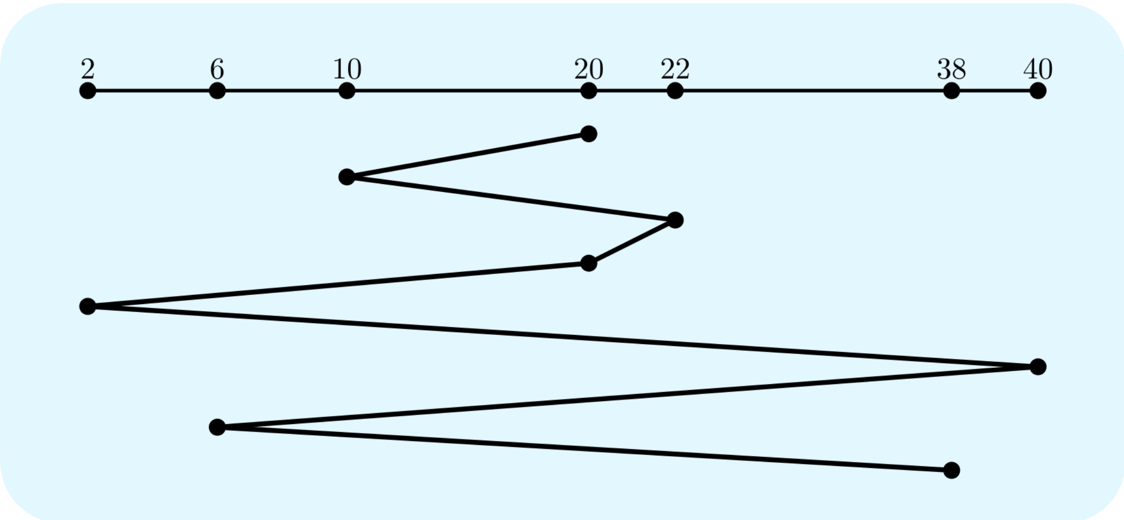
Answers: Disk Scheduling

5.3.1 Disk Scheduling: GATE CSE 1989 | Question: 4-xii top

https://gateoverflow.in/88222

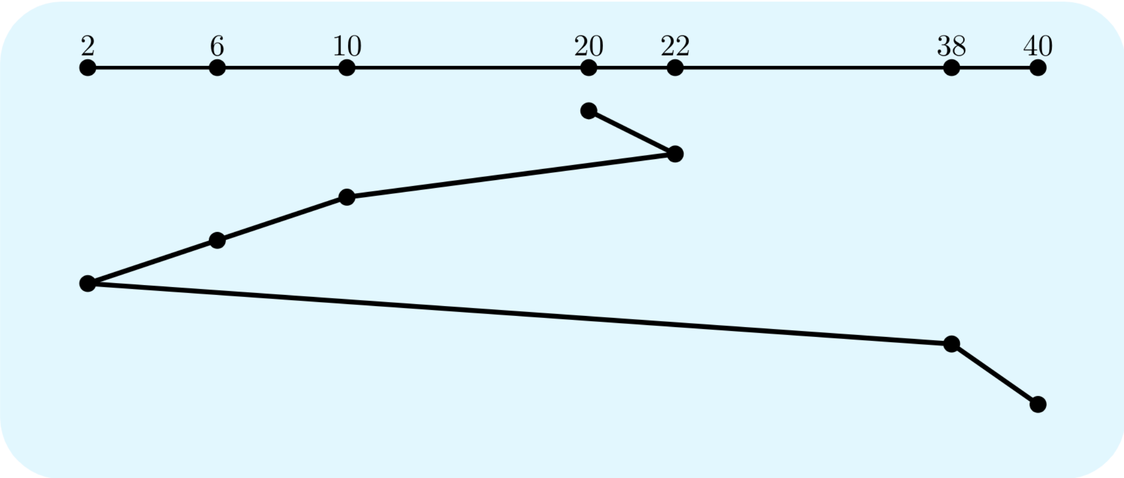
✔

A. In **FCFS** sequence will be $\Rightarrow 20, 10, 22, 20, 2, 40, 6, 38$
Total movement: $|20 - 10| + |10 - 22| + |22 - 20| + |20 - 2| + |2 - 40| + |40 - 6| + |6 - 38| = 146$
So total seek time $= 146 \times 6 = \mathbf{876msec}$



FCFS

B. In **Closest cylinder next** sequence will be $\Rightarrow 20, 22, 10, 6, 2, 38, 40$
Total movement: $|20 - 22| + |22 - 2| + |2 - 40| = 60$
So total seek time $= 60 \times 6 = \mathbf{360msec}$



Closest cylinder next

👍 27 votes

-- Lokesh Dafale (8.2k points)

5.3.2 Disk Scheduling: GATE CSE 1990 | Question: 9b top

https://gateoverflow.in/85678

✔

1. **SSTF**
   Sequence will be $\Rightarrow$ $50, 49, 53, 55, 65, 65, 78, 36, 20, 16, 12, 3, 1$

2. **Elevator disk scheduling (SCAN)**
   Here, I assume
   78 is the extreme point
   Sequence will be $\Rightarrow$ $50, 53, 55, 65, 65, 78, 49, 36, 20, 16, 12, 3, 1$

> SCAN(Elevator) It scans down towards the nearest end first

👍 15 votes

-- Lokesh Dafale (8.2k points)

5.3.3 Disk Scheduling: GATE CSE 1995 | Question: 20 top

https://gateoverflow.in/2658

✔

FCFS : $55 \to 10 \to 70 \to 75 \to 23 \to 65 \Rightarrow 45 + 60 + 5 + 52 + 42 = 204.$

SSTF : $55 \to 65 \to 70 \to 75 \to 23 \to 10 \Rightarrow 10 + 5 + 5 + 52 + 13 = 85$

Hence, SSTF.

👍 32 votes

-- kireeti (1k points)

5.3.4 Disk Scheduling: GATE CSE 1999 | Question: 1.10 top

https://gateoverflow.in/1463

✔

A. Farthest cylinder next $\to$ This might be candidate for worst algorithm . This is false.
B. Nearest cylinder next $\to$ This is output.
C. First come first served $\to$ This will not give best throughput. It is random .
D. Elevator algorithm $\to$ This is good but issue is that once direction is fixed we don't come back, until we go all the other way. So it does not give best throughput.

Correct Answer: B

👍 36 votes

-- Akash Kanase (36k points)

5.3.5 Disk Scheduling: GATE CSE 2004 | Question: 12 top

https://gateoverflow.in/1009

✔ Question says "single sequential user process". So, all the requests to disk scheduler will be in sequence and each one will be blocking the execution and hence there is no use of any disk scheduling algorithm. Any disk scheduling algorithm gives the same input sequence and hence the improvement will be 0% for SSTF over FCFS.

Correct Answer: D

74 votes

-- Arjun Suresh (332k points)

5.3.6 Disk Scheduling: GATE CSE 2009 | Question: 31 top

https://gateoverflow.in/1317

✔ Answer is **(B).**

$= (50-34)+(34-20)+(20-19)+(19-15)+(15-10)+(10-7)+(7-6)+(6-4)+(4-2)+(73-2)$

$= 16+14+1+4+5+3+1+2+2+71$
$= 119\ ms$

25 votes

-- Sona Praneeth Akula (3.4k points)

5.3.7 Disk Scheduling: GATE CSE 2014 Set 1 | Question: 19 top

https://gateoverflow.in/1786

✔ Requests are serviced in following order

100 105 110 90 85 135 145 30

So, request of 90 is serviced after 3 requests.

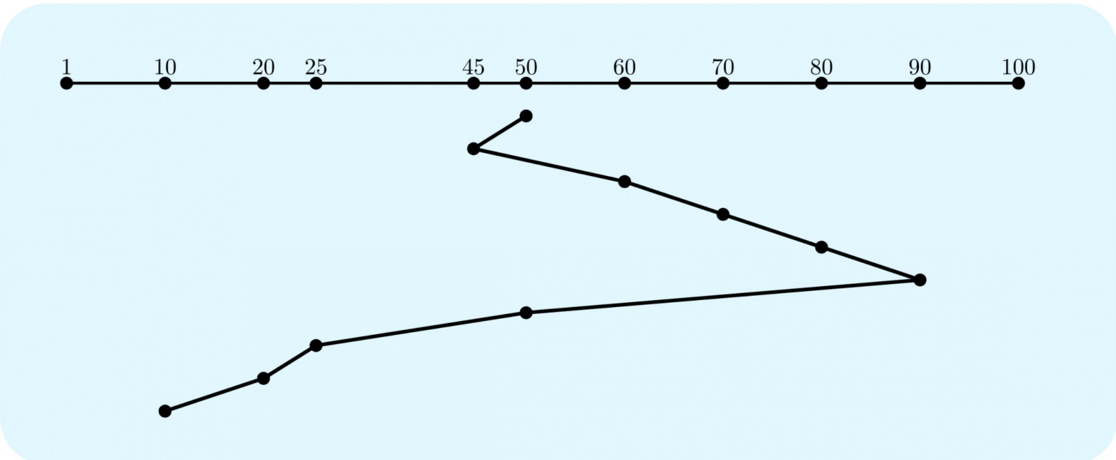
36 votes

-- Pooja Palod (24.1k points)

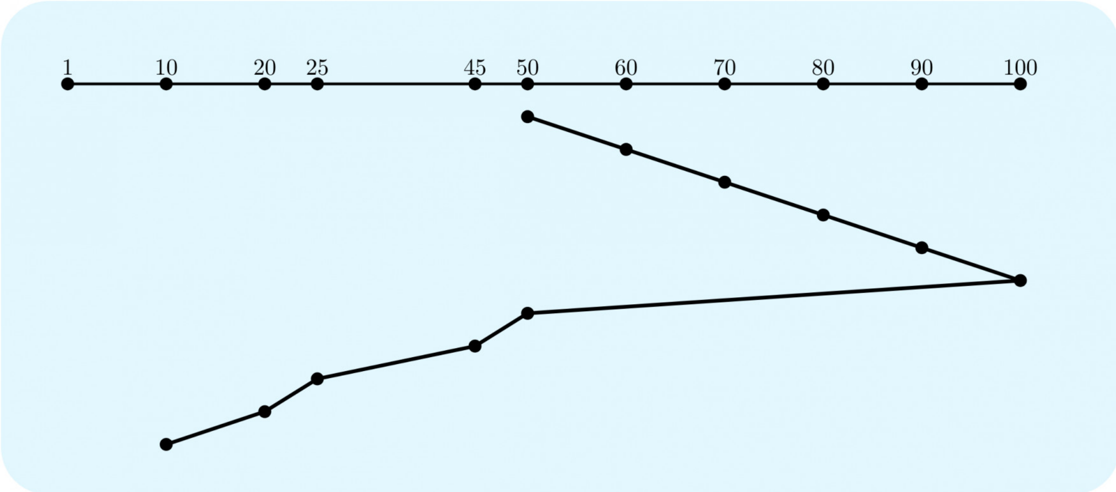
5.3.8 Disk Scheduling: GATE CSE 2015 Set 1 | Question: 30 top

https://gateoverflow.in/8227

✔ Refer : http://www.cs.iit.edu/~cs561/cs450/disksched/disksched.html



Shortest seek time first(SSTF)



SCAN(Elevator)

So, for SSTF it takes 130 head movements and for SCAN it takes 140 head movements.

Hence, not additional but $140 - 130 = 10$ less head movements SSTF takes.

References

65 votes -- Amar Vashishth (25.2k points)

5.3.9 Disk Scheduling: GATE CSE 2016 Set 1 | Question: 48 top

https://gateoverflow.in/39716

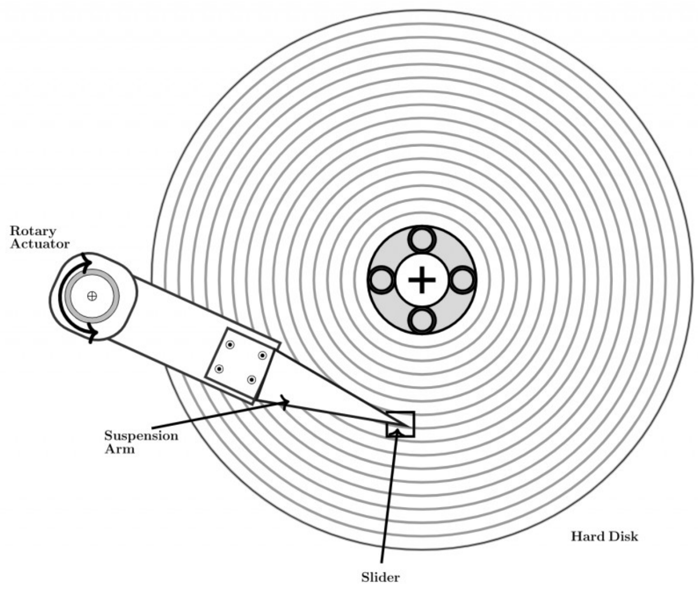
✔ $63 \rightarrow 191 = 128$
$191 \rightarrow 10 = 181$
$10 \rightarrow 47 = 37$
$\text{Total} = 346$

67 votes -- Abhilash Panicker (7.6k points)

Answer is 346 as already calculated in answers here. Those having some doubt regarding long jump can check this image.

In the question Total Head Movements are asked. When Head reaches any End, **there is no mechanism for head to jump directly to some arbitrary track. It has to Move. So it has to move along the tracks to reach Track Request on other side. Therefore head will move and we must count it.**

Since the purpose of disk scheduling algorithms is to reduce such Head movements by finding an Optimal algorithm. If we ignore the move which is actually happening in disk, that doesn't serve the purpose of analyzing the algorithms.



32 votes

-- Anurag Semwal (6.7k points)

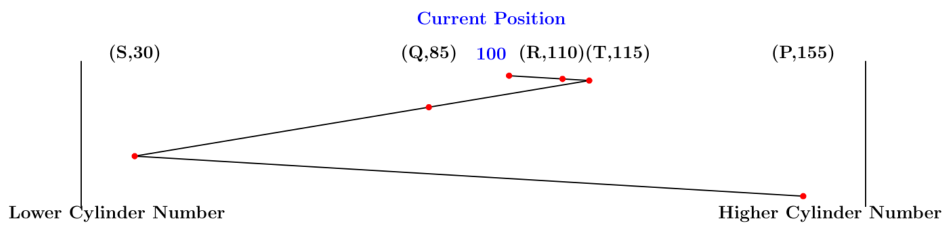
5.3.10 Disk Scheduling: GATE CSE 2020 | Question: 35 top

https://gateoverflow.in/333196

Shortest Seek Time First (SSTF), selects the request with minimum to seek time first from the current head position. In the given question disk requests are given in the form of $\langle request\ id,\ cylinder\ number\rangle$

Cylinder Queue: $(P, 155), (Q, 85), (R, 110), (S, 30), (T, 115)$

Head starts at: 100



- It is clear that R and T are serviced before P.
- Q is serviced when head is moving towards lower cylinders and P is serviced when head is moving towards higher cylinders thus reverses it direction at S.

Option B) is the correct answer

7 votes

-- Ashwani Kumar (13k points)

5.3.11 Disk Scheduling: GATE IT 2004 | Question: 62 top

https://gateoverflow.in/3705

Answer is (C)

SSTF: (90) 120 115 110 130 80 70 30 25 20

Direction changes at $120, 110, 130$

FCFS: (90) 120 30 70 115 130 110 80 20 25

direction changes at $120, 30, 130, 20$

42 votes

-- Sandeep_Uniyal (6.5k points)

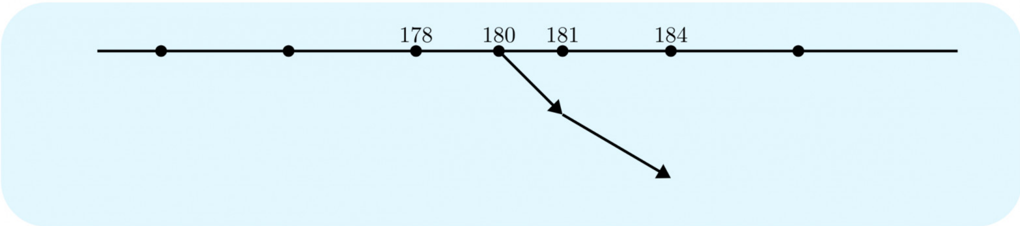
5.3.12 Disk Scheduling: GATE IT 2007 | Question: 82 top

https://gateoverflow.in/3534

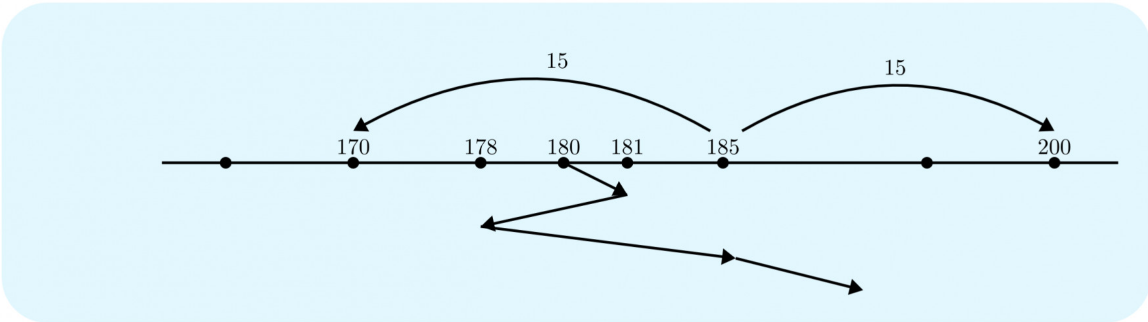
It should be (**B**).

When the head starts from 180. It seeks the nearest track which is 181. Then, from 181 it seeks the nearest one which is 178 and 184. But the difference in both from 181 is same and as given in the question. If there is a tie then the head wont change its direction, and therefore to change the direction we need to consider 178. and thus we can eliminate option **(A)** and **(C)**.

We need head direction change after every request service



Options A and C



Option D

Coming next to option **(B)** and **(D)**.

Following the above procedure you'll see that option **(D)** is eliminated on similar ground. And thus you can say option **(B)** is correct.

25 votes

-- Gate Keeda (15.9k points)

5.3.13 Disk Scheduling: GATE IT 2007 | Question: 83

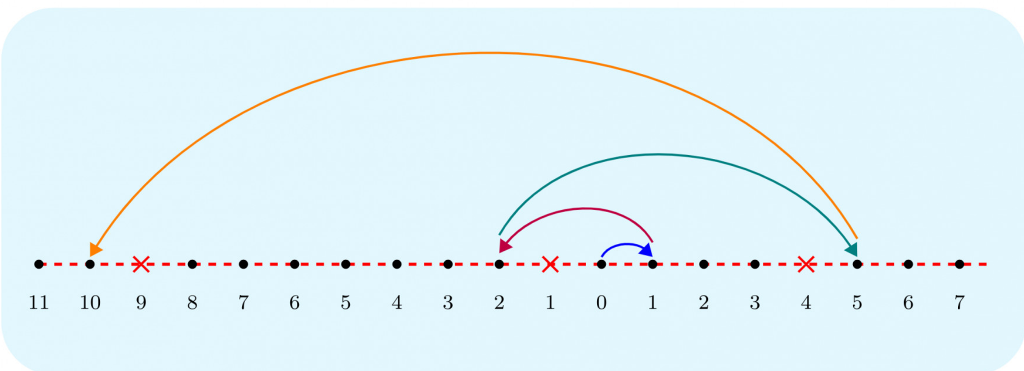
https://gateoverflow.in/3535

✔ We need **two** conditions to satisfy:

1. The **alternating direction** with **shortest seeks time first policy**.
2. Maximize the no. of requests.

The first condition can be satisfied by not having two requests in the equal distance from the current location. As shown below, we must not have request located int he red marked positions.

Now to maximize the no of request we need the requests to be located as **compact** as possible. Which can be done by just placing the request in the next position after the red marked position in a particular direction (the direction in which the head need to move now to satisfy the 1^{st} criteria).



Seek length sequences for maximum cardinality and alternating head movements:

- $1, 3, 7, 15, \ldots$

- Or, $2^1 - 1, 2^2 - 1, 2^3 - 1, 2^4 - 1, \ldots$
- We have 2048 tracks so, maximum swing (seek length) can be 2047
- Which corresponds to a seek length of $2^{11} - 1$ in the 11^{th} service.

Correct Answer: C

74 votes

-- Debashish Deka (40.8k points)

5.4 Disks (31) top

5.4.1 Disks: GATE CSE 1990 | Question: 7-c top

https://gateoverflow.in/85406

A certain moving arm disk-storage device has the following specifications:

- Number of tracks per surface $= 404$
- Track storage capacity $= 130030$ bytes.
- Disk speed $= 3600$ rpm
- Average seek time $= 30$ m secs.

Estimate the average latency, the disk storage capacity, and the data transfer rate.

gate1990 operating-system disks descriptive

Answer

5.4.2 Disks: GATE CSE 1993 | Question: 6.7 top

https://gateoverflow.in/2289

A certain moving arm disk storage, with one head, has the following specifications:

- Number of tracks/recording surface $= 200$
- Disk rotation speed $= 2400$ rpm
- Track storage capacity $= 62,500$ bits

The average latency of this device is P ms and the data transfer rate is Q bits/sec. Write the values of P and Q.

gate1993 operating-system disks normal descriptive

Answer

5.4.3 Disks: GATE CSE 1993 | Question: 7.8 top

https://gateoverflow.in/2296

The root directory of a disk should be placed

A. at a fixed address in main memory
B. at a fixed location on the disk
C. anywhere on the disk
D. at a fixed location on the system disk
E. anywhere on the system disk

gate1993 operating-system disks normal

Answer

5.4.4 Disks: GATE CSE 1995 | Question: 14 top

https://gateoverflow.in/2650

If the overhead for formatting a disk is 96 bytes for a 4000 byte sector,

A. Compute the unformatted capacity of the disk for the following parameters:
   - Number of surfaces: 8
   - Outer diameter of the disk: 12 cm
   - Inner diameter of the disk: 4 cm
   - Inner track space: 0.1 mm
   - Number of sectors per track: 20
B. If the disk in (A) is rotating at 360 rpm, determine the effective data transfer rate which is defined as the number of bytes transferred per second between disk and memory.

gate1995 operating-system disks normal descriptive

Answer

5.4.5 Disks: GATE CSE 1996 | Question: 23 top

https://gateoverflow.in/2775

A file system with a one-level directory structure is implemented on a disk with disk block size of $4K$ bytes. The disk is used as follows:

Disk-block 0	File Allocation Table, consisting of one 8-bit entry per data block, representing the data block address of the next data block in the file
Disk-block 1	Directory, with one 32 bit entry per file:
Disk-block 2	Data-block 1;
Disk-block 3	Data-block 2; etc.

a. What is the maximum possible number of files?
b. What is the maximum possible file size in blocks

gate1996 operating-system disks normal file-system descriptive

Answer

5.4.6 Disks: GATE CSE 1997 | Question: 74 top

https://gateoverflow.in/19704

A program P reads and processes 1000 consecutive records from a sequential file F stored on device D without using any file system facilities. Given the following

- Size of each record $= 3200$ bytes
- Access time of $D = 10$ msecs
- Data transfer rate of $D = 800 \times 10^3$ bytes/second
- CPU time to process each record $= 3$ msecs

What is the elapsed time of P if

A. F contains unblocked records and P does not use buffering?

B. F contains unblocked records and P uses one buffer (i.e., it always reads ahead into the buffer)?

C. records of F are organized using a blocking factor of 2 (i.e., each block on D contains two records of F) and P uses one buffer?

gate1997 operating-system disks

Answer

5.4.7 Disks: GATE CSE 1998 | Question: 2-9 top

https://gateoverflow.in/1681

Formatting for a floppy disk refers to

A. arranging the data on the disk in contiguous fashion

B. writing the directory

C. erasing the system data

D. writing identification information on all tracks and sectors

gate1998 operating-system disks normal

Answer

5.4.8 Disks: GATE CSE 1998 | Question: 25-a top

https://gateoverflow.in/1740

Free disk space can be used to keep track of using a free list or a bit map. Disk addresses require d bits. For a disk with B blocks, F of which are free, state the condition under which the free list uses less space than the bit map.

gate1998 operating-system disks descriptive

Answer

5.4.9 Disks: GATE CSE 1998 | Question: 25b top https://gateoverflow.in/41055

Consider a disk with c cylinders, t tracks per cylinder, s sectors per track and a sector length s_l. A logical file d_l with fixed record length r_l is stored continuously on this disk starting at location (c_L, t_L, s_L), where c_L, t_L and S_L are the cylinder, track and sector numbers, respectively. Derive the formula to calculate the disk address (i.e. cylinder, track and sector) of a logical record n assuming that $r_l = s_l$.

gate1998 operating-system disks descriptive

Answer

5.4.10 Disks: GATE CSE 1999 | Question: 2-18, ISRO2008-46 top https://gateoverflow.in/1496

Raid configurations of the disks are used to provide

A. Fault-tolerance

B. High speed

C. High data density

D. (A) & (B)

gate1999 operating-system disks easy isro2008

Answer

5.4.11 Disks: GATE CSE 2001 | Question: 1.22 top https://gateoverflow.in/715

Which of the following requires a device driver?

A. Register
B. Cache
C. Main memory
D. Disk

gate2001-cse operating-system disks easy

Answer

5.4.12 Disks: GATE CSE 2001 | Question: 20 top https://gateoverflow.in/761

Consider a disk with the 100 tracks numbered from 0 to 99 rotating at 3000 rpm. The number of sectors per track is 100 and the time to move the head between two successive tracks is 0.2 millisecond.

A. Consider a set of disk requests to read data from tracks $32, 7, 45, 5$ and 10. Assuming that the elevator algorithm is used to schedule disk requests, and the head is initially at track 25 moving up (towards larger track numbers), what is the total seek time for servicing the requests?
B. Consider an initial set of 100 arbitrary disk requests and assume that no new disk requests arrive while servicing these requests. If the head is initially at track 0 and the elevator algorithm is used to schedule disk requests, what is the worse case time to complete all the requests?

gate2001-cse operating-system disks normal descriptive

Answer

5.4.13 Disks: GATE CSE 2001 | Question: 8 top https://gateoverflow.in/749

Consider a disk with the following specifications: 20 surfaces, 1000 tracks/surface, 16 sectors/track, data density 1 KB/sector, rotation speed 3000 rpm. The operating system initiates the transfer between the disk and the memory sector-wise. Once the head has been placed on the right track, the disk reads a sector in a single scan. It reads bits from the sector while the head is passing over the sector. The read bits are formed into bytes in a serial-in-parallel-out buffer and each byte is then transferred to memory. The disk writing is exactly a complementary process.

For parts (C) and (D) below, assume memory read-write time = 0.1 microseconds/byte, interrupt driven transfer has an interrupt overhead = 0.4 microseconds, the DMA initialization, and termination overhead is negligible compared to the total sector transfer time. DMA requests are always granted.

A. What is the total capacity of the desk?
B. What is the data transfer rate?
C. What is the percentage of time the CPU is required for this disk I/O for byte-wise interrupts driven transfer?
D. What is the maximum percentage of time the CPU is held up for this disk I/O for cycle-stealing DMA transfer?

gate2001-cse operating-system disks normal descriptive

Answer

5.4.14 Disks: GATE CSE 2003 | Question: 25, ISRO2009-12 top

https://gateoverflow.in/915

Using a larger block size in a fixed block size file system leads to

A. better disk throughput but poorer disk space utilization
B. better disk throughput and better disk space utilization
C. poorer disk throughput but better disk space utilization
D. poorer disk throughput and poorer disk space utilization

gate2003-cse operating-system disks normal isro2009

Answer

5.4.15 Disks: GATE CSE 2004 | Question: 49 top

https://gateoverflow.in/1045

A unix-style I-nodes has 10 direct pointers and one single, one double and one triple indirect pointers. Disk block size is 1 Kbyte, disk block address is 32 bits, and 48-bit integers are used. What is the maximum possible file size?

A. 2^{24} bytes
B. 2^{32} bytes
C. 2^{34} bytes
D. 2^{48} bytes

gate2004-cse operating-system disks normal

Answer

5.4.16 Disks: GATE CSE 2005 | Question: 21 top

https://gateoverflow.in/1357

What is the swap space in the disk used for?

A. Saving temporary html pages
B. Saving process data
C. Storing the super-block
D. Storing device drivers

gate2005-cse operating-system disks easy

Answer

5.4.17 Disks: GATE CSE 2007 | Question: 11, ISRO2009-36, ISRO2016-21 top

https://gateoverflow.in/1209

Consider a disk pack with 16 surfaces, 128 tracks per surface and 256 sectors per track. 512 bytes of data are stored in a bit serial manner in a sector. The capacity of the disk pack and the number of bits required to specify a particular sector in the disk are respectively:

A. 256 Mbyte, 19 bits
B. 256 Mbyte, 28 bits
C. 512 Mbyte, 20 bits

D. 64 Gbyte, 28 bits

gate2007-cse operating-system disks normal isro2016

Answer

5.4.18 Disks: GATE CSE 2008 | Question: 32 top

https://gateoverflow.in/443

For a magnetic disk with concentric circular tracks, the seek latency is not linearly proportional to the seek distance due to

A. non-uniform distribution of requests

B. arm starting and stopping inertia

C. higher capacity of tracks on the periphery of the platter

D. use of unfair arm scheduling policies

gate2008-cse operating-system disks normal

Answer

5.4.19 Disks: GATE CSE 2009 | Question: 51 top

https://gateoverflow.in/1337

A hard disk has 63 sectors per track, 10 platters each with 2 recording surfaces and 1000 cylinders. The address of a sector is given as a triple $\langle c, h, s \rangle$, where c is the cylinder number, h is the surface number and s is the sector number. Thus, the 0^{th} sector is addresses as $\langle 0, 0, 0 \rangle$, the 1^{st} sector as $\langle 0, 0, 1 \rangle$, and so on

The address $\langle 400, 16, 29 \rangle$ corresponds to sector number:

A. 505035
B. 505036
C. 505037
D. 505038

gate2009-cse operating-system disks normal

Answer

5.4.20 Disks: GATE CSE 2009 | Question: 52 top

https://gateoverflow.in/43477

A hard disk has 63 sectors per track, 10 platters each with 2 recording surfaces and 1000 cylinders. The address of a sector is given as a triple $\langle c, h, s \rangle$, where c is the cylinder number, h is the surface number and s is the sector number. Thus, the 0^{th} sector is addresses as $\langle 0, 0, 0 \rangle$, the 1^{st} sector as $\langle 0, 0, 1 \rangle$, and so on

The address of the 1039^{th} sector is

A. $\langle 0, 15, 31 \rangle$
B. $\langle 0, 16, 30 \rangle$
C. $\langle 0, 16, 31 \rangle$
D. $\langle 0, 17, 31 \rangle$

gate2009-cse operating-system disks normal

Answer

5.4.21 Disks: GATE CSE 2011 | Question: 44 top

https://gateoverflow.in/2146

An application loads 100 libraries at startup. Loading each library requires exactly one disk access. The seek time of the disk to a random location is given as 10 ms. Rotational speed of disk is 6000 rpm. If all 100 libraries are loaded from random locations on the disk, how long does it take to load all libraries? (The time to transfer data from the disk block once the head has been positioned at the start of the block may be neglected.)

A. 0.50 s
B. 1.50 s
C. 1.25 s
D. 1.00 s

gate2011-cse operating-system disks normal

Answer

5.4.22 Disks: GATE CSE 2012 | Question: 41

https://gateoverflow.in/2149

A file system with 300 GByte disk uses a file descriptor with 8 direct block addresses, 1 indirect block address and 1 doubly indirect block address. The size of each disk block is 128 Bytes and the size of each disk block address is 8 Bytes. The maximum possible file size in this file system is

A. 3 KBytes
B. 35 KBytes
C. 280 KBytes
D. dependent on the size of the disk

gate2012-cse operating-system disks normal

Answer

5.4.23 Disks: GATE CSE 2013 | Question: 29

https://gateoverflow.in/1540

Consider a hard disk with 16 recording surfaces $(0-15)$ having 16384 cylinders $(0-16383)$ and each cylinder contains 64 sectors $(0-63)$. Data storage capacity in each sector is 512 bytes. Data are organized cylinder-wise and the addressing format is $\langle \text{cylinder no., surface no., sector no.} \rangle$. A file of size 42797 KB is stored in the disk and the starting disk location of the file is $\langle 1200, 9, 40 \rangle$. What is the cylinder number of the last sector of the file, if it is stored in a contiguous manner?

A. 1281
B. 1282
C. 1283
D. 1284

gate2013-cse operating-system disks normal

Answer

5.4.24 Disks: GATE CSE 2014 Set 2 | Question: 20

https://gateoverflow.in/1977

A FAT (file allocation table) based file system is being used and the total overhead of each entry in the FAT is 4 bytes in size. Given a 100×10^6 bytes disk on which the file system is stored and data block size is 10^3 bytes, the maximum size of a file that can be stored on this disk in units of 10^6 bytes is _________.

gate2014-cse-set2 operating-system disks numerical-answers normal file-system

Answer

5.4.25 Disks: GATE CSE 2015 Set 1 | Question: 48

https://gateoverflow.in/8354

Consider a disk pack with a seek time of 4 milliseconds and rotational speed of 10000 rotations per minute (RPM). It has 600 sectors per track and each sector can store 512 bytes of data. Consider a file stored in the disk. The file contains 2000 sectors. Assume that every sector access necessitates a seek, and the average rotational latency for accessing each sector is half of the time for one complete rotation. The total time (in milliseconds) needed to read the entire file is__________________

gate2015-cse-set1 operating-system disks normal numerical-answers

Answer

5.4.26 Disks: GATE CSE 2015 Set 2 | Question: 49

https://gateoverflow.in/8251

Consider a typical disk that rotates at 15000 rotations per minute (RPM) and has a transfer rate of 50×10^6 bytes/sec. If the average seek time of the disk is twice the average rotational delay and the controller's transfer time is 10 times the disk transfer time, the average time (in milliseconds) to read or write a 512-byte sector of the disk is _____

gate2015-cse-set2 operating-system disks normal numerical-answers

Answer

5.4.27 Disks: GATE CSE 2018 | Question: 53 top

https://gateoverflow.in/204128

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

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

Currently 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 _____

gate2018-cse operating-system disks numerical-answers

Answer

5.4.28 Disks: GATE IT 2005 | Question: 63 top

https://gateoverflow.in/3824

In a computer system, four files of size 11050 bytes, 4990 bytes, 5170 bytes and 12640 bytes need to be stored. For storing these files on disk, we can use either 100 byte disk blocks or 200 byte disk blocks (but can't mix block sizes). For each block used to store a file, 4 bytes of bookkeeping information also needs to be stored on the disk. Thus, the total space used to store a file is the sum of the space taken to store the file and the space taken to store the book keeping information for the blocks allocated for storing the file. A disk block can store either bookkeeping information for a file or data from a file, but not both.
What is the total space required for storing the files using 100 byte disk blocks and 200 byte disk blocks respectively?

A. 35400 and 35800 bytes
B. 35800 and 35400 bytes
C. 35600 and 35400 bytes
D. 35400 and 35600 bytes

gate2005-it operating-system disks normal

Answer

5.4.29 Disks: GATE IT 2005 | Question: 81-a top

https://gateoverflow.in/3845

A disk has 8 equidistant tracks. The diameters of the innermost and outermost tracks are 1 cm and 8 cm respectively. The innermost track has a storage capacity of 10 MB.

What is the total amount of data that can be stored on the disk if it is used with a drive that rotates it with

I. Constant Linear Velocity
II. Constant Angular Velocity?

A. I. 80 MB; II. 2040 MB
B. I. 2040 MB; II 80 MB
C. I. 80 MB; II. 360 MB
D. I. 360 MB; II. 80 MB

gate2005-it operating-system disks normal

Answer

5.4.30 Disks: GATE IT 2005 | Question: 81-b top

https://gateoverflow.in/3846

A disk has 8 equidistant tracks. The diameters of the innermost and outermost tracks are 1 cm and 8 cm respectively. The innermost track has a storage capacity of 10 MB.

If the disk has 20 sectors per track and is currently at the end of the 5^{th} sector of the inner-most track and the head can move at a speed of 10 meters/sec and it is rotating at constant angular velocity of 6000 RPM, how much time will it take to read 1 MB contiguous data starting from the sector 4 of the outer-most track?

A. $13.5 \; ms$
B. $10 \; ms$
C. $9.5 \; ms$
D. $20 \; ms$

gate2005-it operating-system disks normal

Answer

5.4.31 Disks: GATE IT 2007 | Question: 44, ISRO2015-34 top

https://gateoverflow.in/3479

A hard disk system has the following parameters :

- Number of tracks $= 500$
- Number of sectors/track $= 100$
- Number of bytes /sector $= 500$
- Time taken by the head to move from one track to adjacent track $= 1\ ms$
- Rotation speed $= 600\ rpm$.

What is the average time taken for transferring 250 bytes from the disk ?

A. $300.5\ ms$
B. $255.5\ ms$
C. $255\ ms$
D. $300\ ms$

gate2007-it operating-system disks normal isro2015

Answer

Answers: Disks

5.4.1 Disks: GATE CSE 1990 | Question: 7-c top

https://gateoverflow.in/85406

✔

1. Avg Latency $= \frac{1}{2} \times \frac{60}{R} = \frac{1}{2} \times \frac{60}{3600} = 8.33$ ms
2. Disk Storage Capacity = *(We need a number of surface to calculate it)* 404×130030 Bytes $\simeq 50$ MB per surface (approx)
3. Data transfer rate = Track capacity $\times \frac{R}{60} = 130030 \times \frac{3600}{60} = 7801.8$ kBps

16 votes

-- Lokesh Dafale (8.2k points)

5.4.2 Disks: GATE CSE 1993 | Question: 6.7 top

https://gateoverflow.in/2289

✔ RPM $= 2400$

So, in 60 s, the disk rotates 2400 times.

Average latency is the time for half a rotation $= 0.5 \times 60/2400\ s = 3/240\ s = 12.5$ ms.

In one full rotation, entire data in a track can be transferred. Track storage capacity $= 62500$ bits.

So, disk transfer rate $= 62500 \times 2400/60$ s $= 2.5 \times 10^6$ bps.

62 votes

-- Arjun Suresh (332k points)

5.4.3 Disks: GATE CSE 1993 | Question: 7.8 top

https://gateoverflow.in/2296

✔ File system uses directories which are the files containing the name and location of other file in the file system. Unlike other file,directory does not store the user data. Directories are the file that can point to other directories. Root directory point to various user directory. So they will be stored in such a way that user cannot easily modify them. They should be placed at fixed location on the disk.

Correct Answer: B

39 votes

-- neha pawar (3.3k points)

5.4.4 Disks: GATE CSE 1995 | Question: 14 top

https://gateoverflow.in/2650

✔ **For (A) part :**

No of track $=$ Recording width/ inner space between track

Recording width $=$ (Outer Diameter $-$ Inner Diameter $)/2 = (12-4)/2 = 4\ cm$

Therefore no. of track $= 4$ cm$/0.1$ mm $= 400$ track

Since they have ask capacity of unformatted disk , so no 96 bytes in 4000 bytes would be wasted for non data purpose

Whole 4000 is used

So, total capacity $= 400 \times 8 \times 20 \times 4000 = 256 \times 10^6$ Bytes $=$ **256 MB**

For (B) part :

Its is given 360 rotations in 60 seconds

That is 360 rotations $= 60$ sec

Therefore, 1 rotations will take $(1/6)$ sec

In $(1/6)$ sec - we can read one track $= 20 \times (4000 - 96)\ B = 20 \times 3904\ B$

Then, in 1 sec it will be $= 20 \times 3904 \times 6$ bytes $=$ Data transfer rate $=$ **468.480 KBps** (when we consider 1 Read/Write Head **for all** surface) .

If we consider **1 Read/Write Heads per surface (which is default approach),** then n**umber of surfaces = 8**

Data transfer rate $= (468.480 \times 8)\ KBps = 3747.84\ KBps$

But for our convenience we consider only 1 surface, it reads from one surface at a time. As data transfer rate is measured wrt a single surface only .

Hence, for part B, the correct answer is **468.480 KBps.**

40 votes

-- spriti1991 (1.5k points)

5.4.5 Disks: GATE CSE 1996 | Question: 23 top

https://gateoverflow.in/2775

✔

a. Maximum possible number of files:
   As per question, 32 bits (or 4 Bytes) are required per file. And there is only one block to store this, ie the Disk block 1, which is of size 4KB. So number of files possible is 4 KB$/$4 Bytes= 1 K files possible.
b. Max file size:
   As per question the Disk Block Address (FAT entry gives DBA) is of 8 bits. So, ideally the max file size should be $2^8 = 256$ Block size.. But question makes it clear that two blocks, DB0 and DB1, stores control information. So. effectively we have $256 - 2 = 254$ blocks with us and the max file size shoud be $= 254\times$ size of one block $= 254 \times 4$ KB $=$ 1016 KB.

32 votes

-- Hunaif (575 points)

5.4.6 Disks: GATE CSE 1997 | Question: 74 top

https://gateoverflow.in/19704

✔

- 1000 consecutive records
- Size of 1 record $=$ 3200 Bytes
- Access Time of device $D = 10$ ms
- Data Transfer Rate of device $D = 800 * 10^3$ Bytes per second.
- CPU time to process Each record $=$ 3ms.
- Time to transfer 1 record (3200 Bytes) $= \frac{3200 \text{ Bytes}}{800*10^3} = 4$ ms

(A) Unblocked records with No buffer. Hence, each time only when a record is fetched in its full entirety it will be processed.

Time to fetch $=$ Access Time for D(Every time you'll access the device. This is also known as device latency)$+$(Data transfer time)

$=$ 10ms $+$ 4ms $=$ 14ms

Total time taken by CPU for each record $=$ fetch $+$ execute $=$ 14ms $+$ 3ms $=$ 17ms

Total time for program $p = 1000 * 17\text{ms} = 17\text{sec}$

(B) Unblocked records and
1 buffer. Records will be accessed one by one and for each record fetched into the buffer, the device delay has to be taken into account.

Time to bring one record into buffer $= 10 + 4 = 14$ ms.

Now let us see how the program goes.

- At $t = 0\text{ms}$, the program starts and the buffer is empty.
- At $t = 14\text{ms}$, R_1 fetched into the buffer and CPU starts processing it.
- At $t = 17\text{ms}$, cpu has processed R_1 and waiting for more records.
- At $t = 28\text{ms}$, buffer gets filled with R_2 and CPU starts processing it.

To get the Total time of the program we think in terms of the last record because when it is processed, all others would already have been processed too!.

Last record R_{1000} would be fetched at $t = 0 + 14 * 1000 = 14000$ ms and 3ms will be taken by CPU to process this.

So, total elapsed time of program $P = 14000 + 3 = 14003\text{ms} = 14.003\text{sec}$

(C) Each disk block contains
2 records and Assuming buffer can hold
1 disk block at a time.

So, 1 Block Size $= 2 * 3200 = 6400$ Bytes

Time to read a block $= \frac{6400}{800*10^3} = 8\text{ms}$.

Each block read you have to incur the device access cost.

So, the total time to fetch one block and bring it into buffer $= 10 + 8 = 18$ ms.

We have 1000 files and so we need to read in 500 blocks.

Each block has two records and therefore CPU time per block $= 6\text{ms}$.

Again to count the program time P, we think in terms of the last Block.

Last block would be fetched at $t = 0 + (18 * 500) = 9000$ ms.

After this 6 ms more to process 2 records present in the 500^{th} block.

So, program time $P = 9000 + 6 = 9006\text{ms} = 9.006\text{sec}$.

35 votes

-- Ayush Upadhyaya (28.4k points)

5.4.7 Disks: GATE CSE 1998 | Question: 2-9 top

https://gateoverflow.in/1681

Answer is (**D**) .

> The formatted disk capacity is always less than the "raw" unformatted capacity specified by the disk's manufacturer, because some portion of each track is used for sector identification and for gaps (empty spaces) between sectors and at the end of the track.

Reference : https://en.wikipedia.org/wiki/Floppy_disk_format

References

33 votes

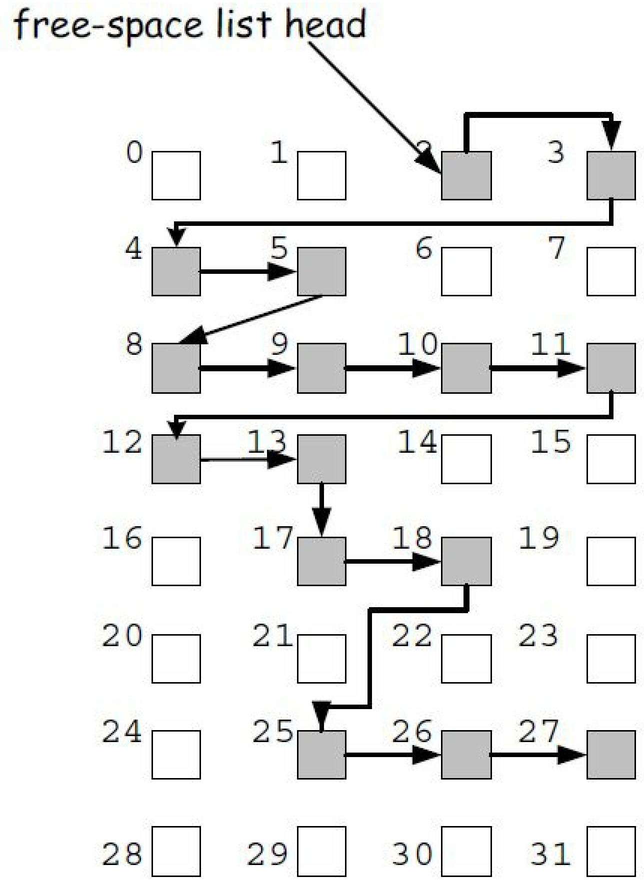
-- Akash Kanase (36k points)

5.4.8 Disks: GATE CSE 1998 | Question: 25-a top

https://gateoverflow.in/1740

Bit map maintains one bit for each block, If it is free then bit will be "0" if occupied then bit will be "1".
For space purpose, it doesn't matter what bit we are using, only matters that how many blocks are there.
For B blocks, Bit map takes space of "B" bits.

Free list is a list that maintains addresses of free blocks only. If we have 3 free blocks then it maintains 3 addresses in a list, if 4 free blocks then 4 address in a list and like that.



Given that we have F free blocks, therefore F addresses in a list, and each address size is d bits therefore Free list takes space of "Fd".

condition under which the free list uses less space than the bit map: $Fd < B$

👍 45 votes

-- Sachin Mittal (15.8k points)

5.4.9 Disks: GATE CSE 1998 | Question: 25b top

https://gateoverflow.in/41055

GIVEN: Consider a disk with c cylinders, t tracks per cylinder, s sectors per track

from this, we can conclude that 1 cylinder contains = t*s sectors

and one track contains =s sectors

now we have to drive the formula of logical address n

so the cylinder no is $\lfloor \left(\frac{n}{ts}\right) \rfloor$

and track number will be floor of ((n%ts)/s)

and sector no will be n%s

👍 8 votes

-- Gurdeep (6.8k points)

5.4.10 Disks: GATE CSE 1999 | Question: 2-18, ISRO2008-46 top

https://gateoverflow.in/1496

✔

A. Fault tolerance and
B. High Speed

👍 21 votes

-- GateMaster Prime (1.2k points)

5.4.11 Disks: GATE CSE 2001 | Question: 1.22 top https://gateoverflow.in/715

✔ A disk driver is a device driver that allows a specific disk drive to communicate with the remainder of the computer. A good example of this driver is a floppy disk driver.

👍 32 votes

-- Bhagirathi Nayak (11.7k points)

5.4.12 Disks: GATE CSE 2001 | Question: 20 top https://gateoverflow.in/761

✔ **Answer for (A):**

We are using SCAN - Elevator algorithms.

We will need to go from $25 \rightarrow 99 \rightarrow 5$. (As we will move up all the way to 99,servicing all request, then come back to 5.)

So, total seeks $= 74 + 94 = 168$

Total time $= 168 \times 0.2 = 33.60000$

Answer for (B):

We need to consider rotational latency too $\rightarrow$

3000 rpm

I.e. 50 rps

$1\ r = 1000/50\ msec = 20\ msec$

So, rotational latency $= 20/2 = 10\ msec$ per access.

In worst case we need to go from tracks $0 - 99$. I.e. 99 seeks

Total time $= 99 \times 0.2 + 10 \times 100 = 1019.8\ msec = 1.019\ sec$

👍 36 votes

-- Akash Kanase (36k points)

5.4.13 Disks: GATE CSE 2001 | Question: 8 top https://gateoverflow.in/749

✔ (a) $20 \times 1000 \times 16 \times 1KB = 3,20,000KB$

(b)

$$\begin{aligned}
\text{3000 rotations} &= \text{60 seconds} \\
\text{1 rotation} &= \frac{60}{3000} \text{ seconds} \\
\text{1 rotation} = \text{1 track} &= \frac{1}{50} \text{ seconds} \\
\text{1 track} = 16 \times 1KB &= \frac{1}{50} \text{ seconds} \\
800KB &= \text{1 second}
\end{aligned}$$

Hence, transfer rate $= 800KB/s$

(c) Data is transferred byte-wise; given in the question.
CPU read/write time for a byte $= 0.1\mu s$
Interrupt overhead (counted in CPU utilization time only) $= 0.4\mu s$
Transfer time for 1 byte data which took place at the rate of 800 $KB/s = 1.25\mu s$
Percentage of CPU time required for this job $= \dfrac{0.1 + 0.4}{0.4 + 0.1 + 1.25} \times 100 = 28.57\%$

(d) Percentage of CPU time held up for disk I/O for cycle stealing DMA transfer $= \dfrac{0.1 + 0}{1.25} \times 100 = 8.00\%$

👍 30 votes

-- Amar Vashishth (25.2k points)

5.4.14 Disks: GATE CSE 2003 | Question: 25, ISRO2009-12 top https://gateoverflow.in/915

✔ Answer is (**A**). Larger block size means less number of blocks to fetch and hence better throughput. But larger block size also means space is wasted when only small size is required.

63 votes -- Arjun Suresh (332k points)

5.4.15 Disks: GATE CSE 2004 | Question: 49 top https://gateoverflow.in/1045

✔ Size of Disk Block $= 1024$ Byte

Disk Blocks address $= 4B$

No. of addresses per block $1024/4 = 256 = 2^8$ addresses

We have:

10 Direct

$1\ SI = 2^8 \text{Indirect} \times 2^{10} = 2^{18}\text{Byte}$

$1\ DI = 2^8\ SI = (2^8)^2 \text{Direct} = 2^{16}\text{Direct} * *2^{10} = 2^{26}\text{Byte}$

$1\ TI = 2^8\ DI = (2^8)^2\ SI = (2^8)^3 = 2^{24}\text{Direct} = 2^{24} \times 2^{10} = 2^{34}$ Byte.

So, total size $= 2^{18} + 2^{26} + 2^{34}$ Byte $+ 10240$Byte . Which is nearly 2^{34} Bytes. (We don't have exact option available. Choose approximate one)

Answer $\rightarrow$ (**C**)

43 votes -- Akash Kanase (36k points)

5.4.16 Disks: GATE CSE 2005 | Question: 21 top https://gateoverflow.in/1357

✔ Swap space(on the disk) is used by Operating System to store the pages that are swapped out of the memory due to less memory available on the disk. Interestingly the Android Operating System, which is Linux kernel under the hood has the swapping disabled and has its own way of handling "low memory" situations.

Pages are basically Process data, hence the answer is (**B**).

30 votes -- Sandeep Kumar (संदीप कुमार) (2.2k points)

5.4.17 Disks: GATE CSE 2007 | Question: 11, ISRO2009-36, ISRO2016-21 top https://gateoverflow.in/1209

✔ Answer is (**A**).

16 surfaces $= 4$ bits, 128 tracks $= 7$ bits, 256 sectors $= 8$ bits, sector size 512 bytes $= 9$ bits

Capacity of disk $= 2^{4+7+8+9} = 2^{28} = 256\ MB$

To specify a particular sector we do not need sector size, so bits required $= 4 + 7 + 8 = 19$

38 votes -- jayendra (6.7k points)

5.4.18 Disks: GATE CSE 2008 | Question: 32 top https://gateoverflow.in/443

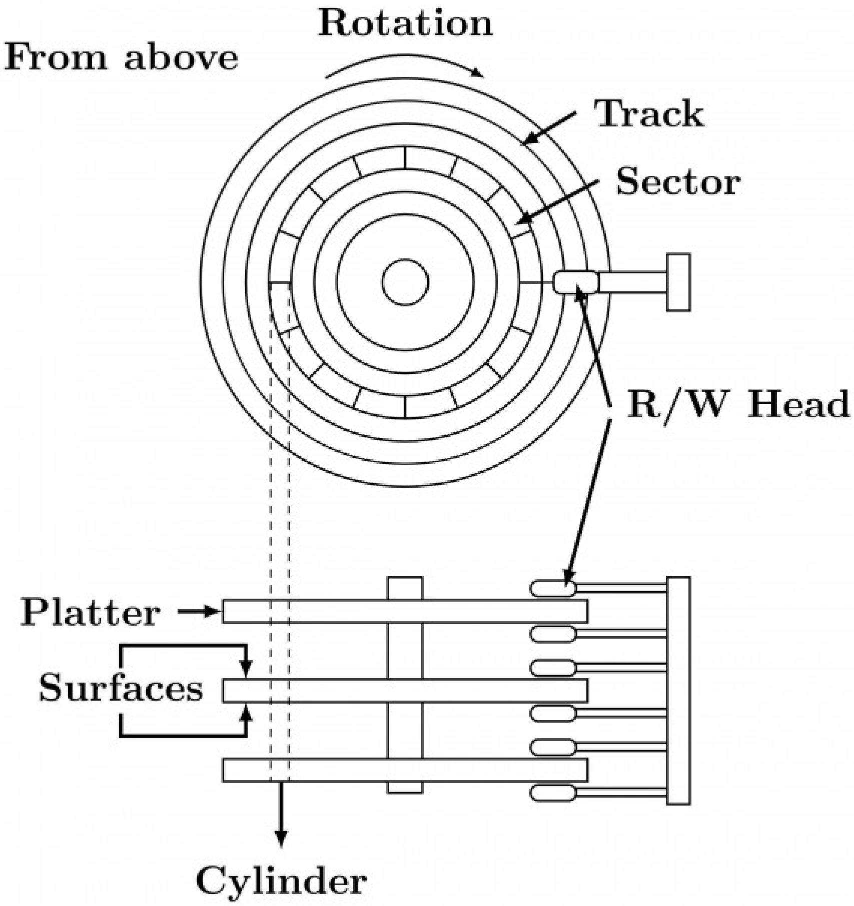
✔ The answer is B, because due to Inertia

Whenever your read-write head moves from 1 track to another track, it has to face resistance due to change in state of motion including speed and direction, which is nothing but inertia. Hence the answer is B

31 votes -- spriti1991 (1.5k points)

5.4.19 Disks: GATE CSE 2009 | Question: 51 top https://gateoverflow.in/1337

✔ The data on a disk is ordered in the following way. It is first stored on the first sector of the first surface of the first cylinder. Then in the next sector, and next, until all the sectors on the first track are exhausted. Then it moves on to the first sector of the second surface (remains at the same cylinder), then next sector and so on. It exhausts all available surfaces for the first cylinder in this way. After that, it moves on to repeat the process for the next cylinder.



So, to reach to the cylinder numbered $400 (401^{th} cylinder)$ we need to skip $400 \times (10 \times 2) \times 63 = 504,000$ sectors.

Then, to skip to the 16^{th} surface of the cylinder numbered 400, we need to skip another $16 \times 63 = 1,008$ sectors.

Finally, to find the 29 sector, we need to move another 29 sectors.

In total, we moved $504,000 + 1,008 + 29 = 505,037$ sectors.

Hence, the answer to 51 is option (**C**).

95 votes -- Pragy Agarwal (18.3k points)

5.4.20 Disks: GATE CSE 2009 | Question: 52 top https://gateoverflow.in/43477

✔ 1039^{th} sector will be stored in track number $(1039 + 1)/63 = 16.5$ (as counting starts from 0 as given in question) and each track has 63 sectors. So, we need to go to 17^{th} track which will be numbered 16 and each cylinder has 20 tracks (10 platters $\times 2$ recording surface each) . Number of extra sectors needed $= 1040 - 16 \times 63 = 32$ and hence the sector number will be 31. So, option (**C**).

47 votes -- Pragy Agarwal (18.3k points)

5.4.21 Disks: GATE CSE 2011 | Question: 44 top https://gateoverflow.in/2146

✔ Disk access time $=$ Seek time $+$ Rotational latency $+$ Transfer time (given that transfer time is neglected)
Seek time $= 10$ ms
Rotational speed $= 6000$ rpm

- 60 $s \rightarrow$ 6000 rotations
- 1 rotation $\rightarrow 60/6000\ s$
- Rotational latency $= 1/2 \times 60/6000\ s = 5$ ms

Total time to transfer one library $= 10 + 5 = 15$ ms
$\therefore$ Total time to transfer 100 libraries $= 100 \times 15$ ms $= 1.5s$

Correct Answer: B

65 votes -- neha pawar (3.3k points)

5.4.22 Disks: GATE CSE 2012 | Question: 41 top

https://gateoverflow.in/2149

✔ Direct block addressing will point to 8 disk blocks $= 8 \times 128\ B = 1\ KB$

Singly Indirect block addressing will point to 1 disk block which has $128/8$ disc block addresses $= (128/8) \times 128\ B = 2\ KB$

Doubly indirect block addressing will point to 1 disk block which has $128/8$ addresses to disk blocks which in turn has $128/8$ addresses to disk blocks $= 16 \times 16 \times 128\ B = 32\ KB$

Total $= 35\ KB$

Answer is (**B**).

61 votes -- Vikrant Singh (11.2k points)

5.4.23 Disks: GATE CSE 2013 | Question: 29 top

https://gateoverflow.in/1540

✔ First convert $\langle 1200, 9, 40 \rangle$ into sector address.

$(1200 \times \mathbf{16} \times \mathbf{64}) + (9 \times \mathbf{64}) + 40 = 1229416$

Number of sectors to store file $= (42797\ KB)/512 = 85594$

Last sector to store file $= 1229416 + 85594 = 1315010$

Now, do reverse engineering,

$1315010/(\mathbf{16} \times \mathbf{64}) = 1284.189453$ (1284 will be cylinder number and remaining sectors $= 194$)

$194/\mathbf{64} = 3.03125$ (3 is surface number and remaining sectors are 2)

$\therefore \langle 1284, 3, 1 \rangle$ is last sector address.

Correct Answer: D

210 votes -- Laxmi (793 points)

$42797\ KB = 42797 \times 1024$ bytes require $42797 \times 1024/512$ sectors $= 85594$ sectors.

$\langle 1200, 9, 40 \rangle$ is the starting address. So, we can have 24 sectors in this recording surface. Remaining 85570 sectors.

85570 sectors require $\lceil \frac{85570}{64} \rceil = 1338$ recording surfaces. We start with recording surface 9, so we can have 7 more in the given cylinder. So, we have $1338 - 7 = 1331$ recording surfaces left.

In a cylinder, we have 16 recording surfaces. So, 1331 recording surfaces require $\lceil \frac{1331}{16} \rceil = 84$ different cylinders.

The first cylinder (after the current one) starts at 1201. So, the last one should be 1284.

$\langle 1284, 3, 1 \rangle$ will be the end address. $(1331 - 16 \times 83 + 1 - 1 = 3$ (3 surfaces full and 1 partial and -1 since address starts from 0), and $85570 - 1337 \times 64 - 1 = 1)$

37 votes -- Arjun Suresh (332k points)

5.4.24 Disks: GATE CSE 2014 Set 2 | Question: 20 top

https://gateoverflow.in/1977

✔ Each datablock will have its entry.

So, Total Number of entries in the FAT $= \dfrac{\text{Disk Capacity}}{\text{Block size}} = \dfrac{100MB}{1KB} = 100K$

Each entry takes up $4B$ as overhead

So, space occupied by overhead $= 100K \times 4B = 400KB = 0.4MB$

We have to give space to Overheads on the same file system and at the rest available space we can store data.

So, assuming that we use all available storage space to store a single file = Maximum file size =
$\text{Total File System size} - \text{Overhead} = 100MB - 0.4MB = 99.6MB$

👍 88 votes

-- Kalpish Singhal (1.6k points)

5.4.25 Disks: GATE CSE 2015 Set 1 | Question: 48 top

https://gateoverflow.in/8354

✔ Since each sector requires a seek,

Total time $= 2000$ (seek time + avg. rotational latency + data transfer time)

Since data transfer rate is not given, we can take that in 1 rotation, all data in a track is read. i.e., in $60/10000 = 6$ ms, 600×512 bytes are read. So, time to read 512 bytes $= 6/600$ ms $= 0.01$ ms

$= 2000 \times (4\ ms + 60 \times 1000/2 \times 10000 + 0.01)$

$= 2000 \times (7.01\ ms)$

$= 14020\ ms.$

http://www.csee.umbc.edu/~olano/611s06/storage-io.pdf

References

👍 69 votes

-- Arjun Suresh (332k points)

5.4.26 Disks: GATE CSE 2015 Set 2 | Question: 49 top

https://gateoverflow.in/8251

✔ Average time to read/write = Avg. seek time + Avg. rotational delay + Effective transfer time

Rotational delay $= \frac{60}{15} = 4$ ms

Avg. rotational delay $= \frac{1}{2} \times 4 = 2$ ms

Avg. seek time $= 2 \times 2 = 4$ ms

Disk transfer time $= \dfrac{512 \text{ Bytes}}{50*10^6 \text{ Bytes/sec}} = 0.0102$ ms

Effective transfer time $= 10\times$ disk transfer time $= 0.102$ ms

So, avg. time to read/write $= 4 + 2 + 0.0102 + 0.102 = 6.11$ ms $\approx$ **6.1 ms**

Reference: http://www.csc.villanova.edu/~japaridz/8400/sld012.htm

References

👍 74 votes

-- Arjun Suresh (332k points)

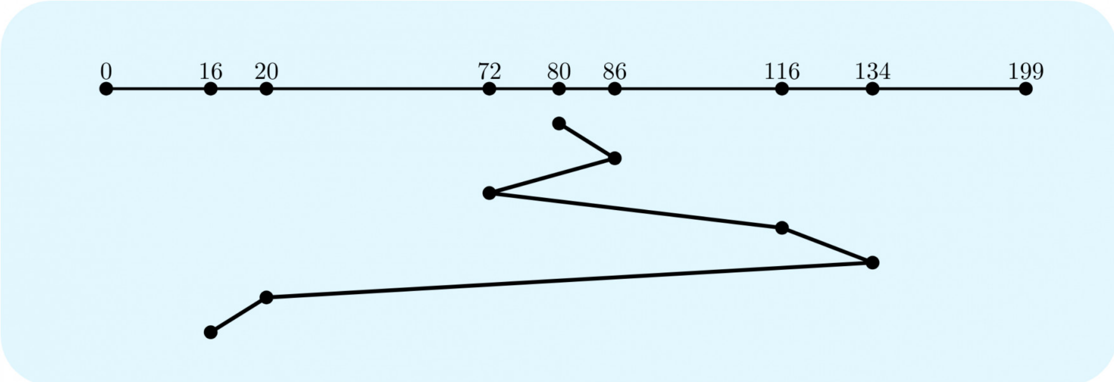
5.4.27 Disks: GATE CSE 2018 | Question: 53 top https://gateoverflow.in/204128

Shortest Seek Time First (SSTF), selects the request with minimum to seek time first from the current head position.

In the given question disk requests are given in the form of $\langle$sectorNo, cylinderNo, platterNo$\rangle$.

Cylinder Queue : $72, 134, 20, 86, 116, 16$

Head starts at : 80



Shortest seek time first(SSTF)

Total head movements in SSTF $= (86-80)+(86-72)+(134-72)+(134-16) = 200$

- Power dissipated in moving 100 cylinder $= 20$ mW
- Power dissipated by 200 movements (say $P1$) $= 0.2 * 200 = 40$ mW
- Power dissipated in reversing head direction once $= 15$ mW
- Number of times head changes its direction $= 3$
- Power dissipated in reversing head direction (say $P2$) $= 3 * 15 = 45$ mW

Total Power Consumption is $P1 + P2 = 85$ mW

Hence, 85 mW is the correct answer.

48 votes -- Ashwani Kumar (13k points)

5.4.28 Disks: GATE IT 2005 | Question: 63 top https://gateoverflow.in/3824

for 100 bytes block:

$11050 = 111$ blocks requiring $111 \times 4 = 444$ bytes of bookkeeping info which requires another 5 disk blocks. So, totally $111 + 5 = 116$ disk blocks. Similarly,
$4990 = 50 + (50 \times 4)/100 = 52$
$5170 = 52 + (52 \times 4)/100 = 55$
$12640 = 127 + (127 \times 4/100) = 133$

$356 \times 100 = 35600$ bytes

For 200 bytes block:

$56 + (56 \times 4/200) = 58$
$25 + (25 \times 4/200) = 26$
$26 + (26 \times 4/200) = 27$
$64 + (64 \times 4/200) = 66$

$177 \times 200 = 35400$

So, (**C**) option.

48 votes -- Viral Kapoor (1.9k points)

5.4.29 Disks: GATE IT 2005 | Question: 81-a top https://gateoverflow.in/3845

- With ***Constant Linear Velocity, CLV,*** the density of bits is uniform from cylinder to cylinder. Because there are more sectors in outer cylinders, the disk spins slower when reading those cylinders, causing the rate of bits passing under the read-write head to remain constant. This is the approach used by modern CDs and DVDs.
- With ***Constant Angular Velocity, CAV,*** the disk rotates at a constant angular speed, with the bit density decreasing on outer cylinders. (These disks would have a constant number of sectors per track on all cylinders.)
- CLV$= 10 + 20 + 30 + 40 + ..80 = 360$
- CAV$= 10 \times 8 = 80$ so answer should be (D)

Edit:- for CLV disk capacity

let track diameters like 1cm, 2cm... 8cm.

As described that density is uniform.

So all tracks has equal storage density.

Track capacity=storage density $\times$ circumference$(2 \times pi \times r)$

For 1st track. 10MB $=$ density $\times 2 \times pi \times 1$

Density $= 10/pi.$ MB/cm

For 2nd track capacity = density $\times$ circumference

$= (10/pi) \times (pi \times 2)$MB $=$ 20MB

Now each track capacity can be calculated and added for disk capacity

72 votes

-- spriti1991 (1.5k points)

5.4.30 Disks: GATE IT 2005 | Question: 81-b top

https://gateoverflow.in/3846

Total Time $=$ Seek $+$ Rotation $+$ Transfer.

Seek Time :

Current Track 1

Destination Track 8

Distance Required to travel $= 4 - 0.5 = 3.5\ Cm$

Time required $= 10\ m/s == 1\ Cm/ms ==$ **3.5 ms** **[Time= Distance / Speed]**

Rotation Time:

6000 RPM in 60 sec

100 RPS in 1 sec

1 Revolution in 10 ms

1 Revolution $=$ Covering entire Track

1 Track $=$ 20 sector

1 sector required $= 10/20 = 0.5\ ms$

Disk is constantly Rotating so when head moved from inner most track to outer most track total movement of disk $= (3.5/0.5) = 7$ sectors

Which means that when disk reached outer most track head was at end of 12^{th} sector

Total Rotational Delay $=$ Time required to go from end of 12 to end of 3 $=$ 11 sectors

1 sector $= 0.5\ ms$ so 11 sector $=$ **5.5ms**

Transfer Time

Total Data in Outer most track $= 10\ MB$

Data in single Sector $= 10\ MB/20 = 0.5\ MB$

Data required to read $= 1\ MB = 2$ sector

Time required to read data $= 2 \times 0.5 =$ **1ms**

Total Time = Seek + Rotation + Transfer $=$ 3.5ms $+$ 5.5ms $+$ 1ms $=$ 10 ms

Correct Answer: B

95 votes

-- Keval Malde (13.3k points)

5.4.31 Disks: GATE IT 2007 | Question: 44, ISRO2015-34 top

https://gateoverflow.in/3479

✔ **option (D)**

Explanation

Avg. time to transfer $=$ Avg. seek time $+$ Avg. rotational delay $+$ Data transfer time

Avg Seek Time

given that : time to move between successive tracks is 1 ms

time to move from track 1 to track 1 : $0ms$
time to move from track 1 to track 2 : $1ms$
time to move from track 1 to track 3 : $2ms$
..
..
time to move from track 1 to track 500 : $499ms$

Avg Seek time $= \frac{\sum 0+1+2+3+...+499}{500}$
$= \mathbf{249.5\ ms}$

Avg Rotational Delay
RMP : 600

600 rotations in 60 sec
one Rotation takes $60/600$ sec $= 0.1$ sec

Avg Rotational Delay $= \frac{0.1}{2}$ { usually $\frac{\text{Rotation time}}{2}$ is taken as Avg Roational Delay }

$= .05sec$

$= \mathbf{50\ ms}$

Data Transfer Time

One 1 Roatation we can read data on one complete track .

$= 100 \times 500 = 50,000$ B data is read in one complete rotation

one complete rotation takes 0.1 s (we seen above)

$0.1 \rightarrow 50,000$ bytes.
250 bytes $\rightarrow 0.1 \times 250/50,000 = \mathbf{0.5\ ms}$

Avg. time to transfer
$=$ **Avg. seek time**
$+$ **Avg. rotational delay**
$+$ **Data transfer time**

$= \mathbf{249.5 + 50 + 0.5}$

$= \mathbf{300\ ms}$

161 votes

-- Akhil Nadh PC (16.5k points)

5.5 File System (6) top

5.5.1 File System: GATE CSE 2002 | Question: 2.22 top

https://gateoverflow.in/852

In the index allocation scheme of blocks to a file, the maximum possible size of the file depends on

A. the size of the blocks, and the size of the address of the blocks.
B. the number of blocks used for the index, and the size of the blocks.
C. the size of the blocks, the number of blocks used for the index, and the size of the address of the blocks.
D. None of the above

gate2002-cse operating-system normal file-system

Answer

5.5.2 File System: GATE CSE 2008 | Question: 20 top

https://gateoverflow.in/418

The data blocks of a very large file in the Unix file system are allocated using

A. continuous allocation
B. linked allocation
C. indexed allocation
D. an extension of indexed allocation

gate2008-cse file-system operating-system normal

Answer

5.5.3 File System: GATE CSE 2017 Set 2 | Question: 08 top

https://gateoverflow.in/118437

In a file allocation system, which of the following allocation scheme(s) can be used if no external fragmentation is allowed ?

1. Contiguous
2. Linked
3. Indexed

A. 1 and 3 only
B. 2 only
C. 3 only
D. 2 and 3 only

gate2017-cse-set2 operating-system file-system normal

Answer

5.5.4 File System: GATE CSE 2019 | Question: 42 top

https://gateoverflow.in/302806

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

gate2019-cse numerical-answers operating-system file-system

Answer

5.5.5 File System: GATE CSE 2021 Set 1 | Question: 15 top

https://gateoverflow.in/357437

Consider a linear list based directory implementation in a file system. Each directory is a list of nodes, where each node contains the file name along with the file metadata, such as the list of pointers to the data blocks. Consider a given directory foo.

Which of the following operations will necessarily require a full scan of foo for successful completion?

A. Creation of a new file in foo
B. Deletion of an existing file from foo
C. Renaming of an existing file in foo
D. Opening of an existing file in foo

gate2021-cse-set1 multiple-selects operating-system file-system

Answer

5.5.6 File System: GATE IT 2004 | Question: 67 top

https://gateoverflow.in/3710

In a particular Unix OS, each data block is of size 1024 bytes, each node has 10 direct data block addresses and three additional addresses: one for single indirect block, one for double indirect block and one for triple indirect block. Also, each block can contain addresses for 128 blocks. Which one of the following is approximately the maximum size of a file in the file system?

A. 512 MB
B. 2 GB
C. 8 GB
D. 16 GB

gate2004-it operating-system file-system normal

Answer

Answers: File System

5.5.1 File System: GATE CSE 2002 | Question: 2.22 top

https://gateoverflow.in/852

✔ In Index allocation size of maximum file can be derived like following:

No of addressable blocks using one Index block (A) = Size of block / Size of block address

No of block addresses available for addressing one file (B) =

No of Maximum blocks we can use for the Index * No of addressable blocks using one Index block (A)

Size of File = B * Size of Block

So, it is clear that;

Answer is **(C).**

A & B are incomplete.

49 votes

-- Akash Kanase (36k points)

5.5.2 File System: GATE CSE 2008 | Question: 20 top

https://gateoverflow.in/418

✔ The data blocks of a very large file in the unix file system are allocated using an extension of indexed allocation or EXT2 file system. Hence, option (**D**) is the right answer.

44 votes

-- Kalpna Bhargav (2.5k points)

5.5.3 File System: GATE CSE 2017 Set 2 | Question: 08 top

https://gateoverflow.in/118437

✔ Both Linked and Indexed allocation free from external fragmentation

Refer:galvin

Reference: https://webservices.ignou.ac.in/virtualcampus/adit/course/cst101/block4/unit4/cst101-bl4-u4-06.htm

References

39 votes

-- Aboveallplayer (12.5k points)

5.5.4 File System: GATE CSE 2019 | Question: 42 top

https://gateoverflow.in/302806

✔ 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 $= \frac{4kB}{4B} = 2^{10}$

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

$= 4.00395\ GB \simeq 4\ GB$

Hence $4GB$ is the correct answer

20 votes -- Ashwani Kumar (13k points)

5.5.5 File System: GATE CSE 2021 Set 1 | Question: 15 top

https://gateoverflow.in/357437

✔ **Correct Options: A, C**

> (**Note:** In the question it's given "which of the following options require a full scan of foo for successful completion" . Meaning the best algorithm scans the list entirely for each type of input to verify the correctness of the procedure and ,can't partially scan and complete for any particular instance...)

Each File in Directory is uniquely referenced by its **name**. So **different files** must have **different names!**

So,

A. **Creation of a New File:** For creating new file, we've to check whether the new name is same as the existing files. Hence, the linked list must be scanned in its entirety.

B. **Deletion of an Existing File:** Deletion of a file doesn't give rise to name conflicts, hence if the node representing the files is found earlier, it can be deleted without a through scan.

C. **Renaming a File:** Can give rise to name conflicts, same reason can be given as option A.

D. **Opening of existing file**: same reason as option B.

4 votes -- NIKHIL SHARMA (605 points)

5.5.6 File System: GATE IT 2004 | Question: 67 top

https://gateoverflow.in/3710

✔ Answer: (**B**)

Maximum file size $= 10 \times 1024$ Bytes $+1 \times 128 \times 1024$ Bytes $+1 \times 128 \times 128 \times 1024$ Bytes $+1 \times 128 \times 128 \times 128 \times 1024$ Bytes $=$ approx $2\ GB$.

35 votes -- Rajarshi Sarkar (27.9k points)

5.6 Fork (5) top

5.6.1 Fork: GATE CSE 2005 | Question: 72 top

https://gateoverflow.in/765


```
Consider the following code fragment:
if (fork() == 0)
{
   a = a + 5;
   printf("%d, %p n", a, &a);
}
else
{
   a = a - 5;
   printf ("%d, %p n", a,& a);
}
```

Let u, v be the values printed by the parent process and x, y be the values printed by the child process. Which one of the following is **TRUE**?

A. $u = x + 10$ and $v = y$
B. $u = x + 10$ and $v! = y$
C. $u + 10 = x$ and $v = y$
D. $u + 10 = x$ and $v! = y$

gate2005-cse operating-system fork normal

Answer

5.6.2 Fork: GATE CSE 2008 | Question: 66 top https://gateoverflow.in/489

A process executes the following code

```
for(i=0; i<n; i++) fork();
```

The total number of child processes created is

A. n
B. $2^n - 1$
C. 2^n
D. $2^{n+1} - 1$

gate2008-cse operating-system fork normal

Answer

5.6.3 Fork: GATE CSE 2012 | Question: 8 top https://gateoverflow.in/40

A process executes the code

```
fork();
fork();
fork();
```

The total number of **child** processes created is

A. 3
B. 4
C. 7
D. 8

gate2012-cse operating-system easy fork

Answer

5.6.4 Fork: GATE CSE 2019 | Question: 17 top https://gateoverflow.in/302831

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

```
#include<unistd.h>
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 ________________ .

gate2019-cse numerical-answers operating-system fork

Answer

5.6.5 Fork: GATE IT 2004 | Question: 64 top https://gateoverflow.in/3707

A process executes the following segment of code :

```
for(i = 1; i <= n; i++)
    fork ();
```

The number of new processes created is

A. n
B. $((n(n+1))/2)$
C. $2^n - 1$
D. $3^n - 1$

gate2004-it operating-system fork easy

Answer

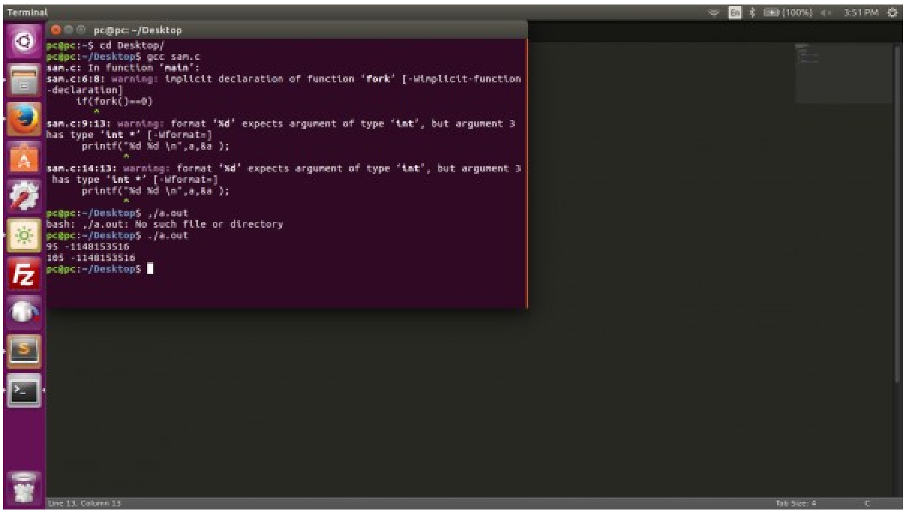
5.6.1 Fork: GATE CSE 2005 | Question: 72 top

https://gateoverflow.in/765

It should be **Option C.**

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int a =100;
    if(fork()==0)
    {
     a=a+5;
     printf("%d %d \n",a,&a );
    }
   else
    {
     a=a-5;
     printf("%d %d \n",a,&a );
    }
}
```

Output:



Fork returns 0 when it is a child process.

```
if ( fork == 0)
```

Is true when it is child . Child increment valule of a .

In the above output:

- 95 is printed by parent : $\mathbf{u}$
- 105 is printed by child : $\mathbf{x}$
  $\Rightarrow \mathbf{u + 10 = x}$

The logical addresses remains the same between the parent and child processes.

Hence, answer should be:

$\mathbf{u + 10 = x}$ **and** $\mathbf{v = y}$

61 votes

-- Akhil Nadh PC (16.5k points)

(c) is the answer. Child is incrementing a by 5 and parent is decrementing a by 5. So, x = u + 10.

During fork(), address space of parent is copied for the child. So, any modifications to child variable won't affect the parent variable or vice-verse. But this copy is for physical pages of memory. The logical addresses remains the same between the parent and child processes.

27 votes

-- gatecse (63.3k points)

5.6.2 Fork: GATE CSE 2008 | Question: 66 top

https://gateoverflow.in/489

Each fork() creates a child which start executing from that point onward. So, number of child processes created will be

$2^n - 1$.

At each fork, the number of processes doubles like from $1 - 2 - 4 - 8 \dots 2^n$. Of these except 1, all are child processes.

Reference: https://gateoverflow.in/3707/gate2004-it_64

References

34 votes

-- Arjun Suresh (332k points)

5.6.3 Fork: GATE CSE 2012 | Question: 8 top

https://gateoverflow.in/40

At each fork() the no. of processes becomes doubled. So, after 3 fork calls, the total no. of processes will be 8. Out of this 1 is the parent process and 7 are child processes. So, total number of child processes created is 7.

42 votes

-- Arjun Suresh (332k points)

5.6.4 Fork: GATE CSE 2019 | Question: 17 top

https://gateoverflow.in/302831

Answer is 31

Fork is called whenever i is even, so we can re-write the code as

```
for(i=0; i<10; i=i+2)
        fork();
```

fork() will be called 5 times($i = 0, 2, 4, 6, 8$)

$\therefore$ Total number of process $2^5 = 32$

Total number of child process would be $2^5 - 1 = 31$

17 votes

-- Abhishek Shaw (1.1k points)

5.6.5 Fork: GATE IT 2004 | Question: 64 top

https://gateoverflow.in/3707

Option (**C**).

At each fork, the number of processes doubles like from $1 - 2 - 4 - 8 \dots 2^n$. Of these except 1, all are child processes.

35 votes

-- prakash (237 points)

5.7 Inter Process Communication (1) top

5.7.1 Inter Process Communication: GATE CSE 1997 | Question: 3.7 top

https://gateoverflow.in/2238

I/O redirection

A. implies changing the name of a file
B. can be employed to use an existing file as input file for a program
C. implies connecting 2 programs through a pipe
D. None of the above

gate1997 operating-system normal inter-process-communication

Answer

Answers: Inter Process Communication

5.7.1 Inter Process Communication: GATE CSE 1997 | Question: 3.7 top https://gateoverflow.in/2238

✔ Answer: (**B**)

Typically, the syntax of these characters is as follows, using < to redirect input, and > to redirect output.

```
command1 > file1
```

executes command1, placing the output in file1, as opposed to displaying it at the terminal, which is the usual destination for standard output. This will clobber any existing data in file1.

Using,

```
command1 < file1
```

executes command1, with file1 as the source of input, as opposed to the keyboard, which is the usual source for standard input.

```
command1 < infile > outfile
```

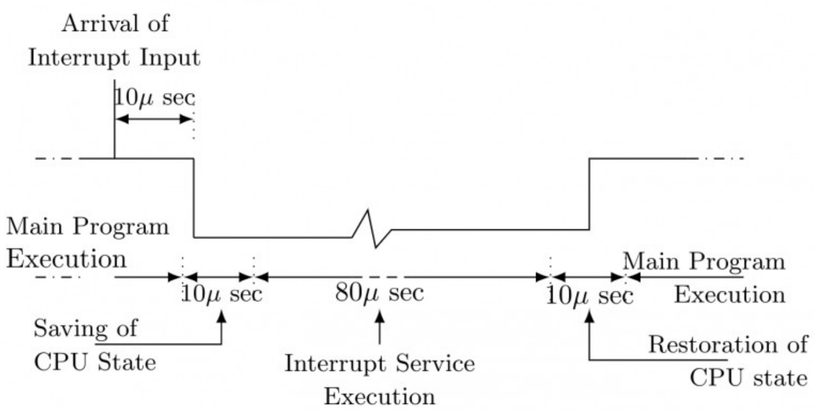
combines the two capabilities: command1 reads from infile and writes to outfile.

👍 35 votes -- Rajarshi Sarkar (27.9k points)

5.8 Interrupts (8) top

5.8.1 Interrupts: GATE CSE 1993 | Question: 6.8 top https://gateoverflow.in/2290

The details of an interrupt cycle are shown in figure.



Given that an interrupt input arrives every 1 msec, what is the percentage of the total time that the CPU devotes for the main program execution.

gate1993 operating-system interrupts normal descriptive

Answer

5.8.2 Interrupts: GATE CSE 1997 | Question: 3.6 top https://gateoverflow.in/2237

The correct matching for the following pairs is:

(A)	Disk Scheduling	(1)	Round robin
(B)	Batch Processing	(2)	SCAN
(C)	Time-sharing	(3)	LIFO
(D)	Interrupt processing	(4)	FIFO

A. A-3 B-4 C-2 D-1
B. A-4 B-3 C-2 D-1
C. A-2 B-4 C-1 D-3
D. A-3 B-4 C-3 D-2

gate1997 operating-system normal disk-scheduling interrupts

Answer

5.8.3 Interrupts: GATE CSE 1997 | Question: 3.8 top

https://gateoverflow.in/2239

When an interrupt occurs, an operating system

A. ignores the interrupt

B. always changes state of interrupted process after processing the interrupt

C. always resumes execution of interrupted process after processing the interrupt

D. may change state of interrupted process to 'blocked' and schedule another process.

gate1997 operating-system interrupts normal

Answer

5.8.4 Interrupts: GATE CSE 1998 | Question: 1.18 top

https://gateoverflow.in/1655

Which of the following devices should get higher priority in assigning interrupts?

A. Hard disk
B. Printer
C. Keyboard
D. Floppy disk

gate1998 operating-system interrupts normal

Answer

5.8.5 Interrupts: GATE CSE 1999 | Question: 1.9 top

https://gateoverflow.in/1462

Listed below are some operating system abstractions (in the left column) and the hardware components (in the right column)

(A)	Thread	1.	Interrupt
(B)	Virtual address space	2.	Memory
(C)	File system	3.	CPU
(D)	Signal	4.	Disk

A. (A) – 2 (B) – 4 (C) – 3 (D) – 1
B. (A) – 1 (B) – 2 (C) – 3 (D) – 4
C. (A) – 3 (B) – 2 (C) – 4 (D) – 1
D. (A) – 4 (B) – 1 (C) – 2 (D) – 3

gate1999 operating-system easy interrupts virtual-memory disks

Answer

5.8.6 Interrupts: GATE CSE 2001 | Question: 1.12 top

https://gateoverflow.in/705

A processor needs software interrupt to

A. test the interrupt system of the processor
B. implement co-routines
C. obtain system services which need execution of privileged instructions
D. return from subroutine

gate2001-cse operating-system interrupts easy

Answer

5.8.7 Interrupts: GATE CSE 2011 | Question: 11 top https://gateoverflow.in/2113

A computer handles several interrupt sources of which of the following are relevant for this question.

- Interrupt from CPU temperature sensor (raises interrupt if CPU temperature is too high)
- Interrupt from Mouse (raises Interrupt if the mouse is moved or a button is pressed)
- Interrupt from Keyboard (raises Interrupt if a key is pressed or released)
- Interrupt from Hard Disk (raises Interrupt when a disk read is completed)

Which one of these will be handled at the **HIGHEST** priority?

A. Interrupt from Hard Disk
B. Interrupt from Mouse
C. Interrupt from Keyboard
D. Interrupt from CPU temperature sensor

gate2011-cse operating-system interrupts normal

Answer

5.8.8 Interrupts: GATE CSE 2018 | Question: 9 top https://gateoverflow.in/204083

The following are some events that occur after a device controller issues an interrupt while process L is under execution.

- P. The processor pushes the process status of L onto the control stack
- Q. The processor finishes the execution of the current instruction
- R. The processor executes the interrupt service routine
- S. The processor pops the process status of L from the control stack
- T. The processor loads the new PC value based on the interrupt

Which of the following is the correct order in which the events above occur?

A. QPTRS
B. PTRSQ
C. TRPQS
D. QTPRS

gate2018-cse operating-system interrupts normal

Answer

Answers: Interrupts

5.8.1 Interrupts: GATE CSE 1993 | Question: 6.8 top https://gateoverflow.in/2290

✔ Time to service an interrupt $=$ saving of cpu state $+$ ISR execution $+$ restoring of CPU state
$= (80 + 10 + 10) \times 10^{-6} = \mathbf{100 \quad microseconds}$

For every 1 ms an interrupt occurs which is served for 100 microseconds
1 ms $\rightarrow$ 1000 microseconds.

After every 1000 microseconds of main code execution, 100 microseconds for interrupt overhead exists.

Thus, for every 1000 microseconds, $(1000 - 100) = 900$ microseconds of main program and 100 microseconds of interrupt overhead exists.

Thus, $900/1000$ is usage of CPU to execute main program

% of CPU time used to execute main program is $\mathbf{(900/1000) \times 100 = 90.00\%}$.

29 votes

-- Surabhi Kadur (819 points)

5.8.2 Interrupts: GATE CSE 1997 | Question: 3.6 top https://gateoverflow.in/2237

✔ (**C**) is answer. Interrupt processing is LIFO because when we are processing an interrupt, we disable the interrupts originating from lower priority devices so lower priority interrupts can not be raised. If an interrupt is detected then it means that it has higher priority than currently executing interrupt so this new interrupt will preempt the current interrupt so, LIFO. Other matches are easy

44 votes -- ashish gusai (523 points)

5.8.3 Interrupts: GATE CSE 1997 | Question: 3.8 top https://gateoverflow.in/2239

✔ Think about this:
When a process is running and after time slot is over, who schedules new process?
- Scheduler.

But to run "scheduler" itself, we have to first schedule scheduler.
This is catch here, We need hardware support to schedule scheduler. That is hardware timer. When timer expires, then hardware generates interrupt and scheduler gets schedule.
Now after servicing that interrupt, schedular may schedule another process.

This was about Hardware interrupt.

Now think if user invokes a system call, System call in effect leads to interrupt, and after this interrupt CPU resumes execution of current running process,

Conclusion: Its about type of interrupt being serviced.
Options with "always" are false.

Hence, option (**D**).

78 votes -- Sachin Mittal (15.8k points)

5.8.4 Interrupts: GATE CSE 1998 | Question: 1.18 top https://gateoverflow.in/1655

✔ It should be a Hard disk. I don't think there is a rule like that. But hard disk makes sense compared to others here.
http://www.ibm1130.net/functional/IOInterrupts.html

References

33 votes -- Arjun Suresh (332k points)

5.8.5 Interrupts: GATE CSE 1999 | Question: 1.9 top https://gateoverflow.in/1462

✔ Answer: (C) A - 3, B - 2, C - 4, D - 1

(A)	Thread	3.	CPU
(B)	Virtual address space	2.	Memory
(C)	File system	4.	Disk
(D)	Signal	1.	Interrupt

Why?

- *Thread* & Process are handled by CPU.
- *Virtual Address Space* is a type of memory address.
- *File System* is used for disk management.
- *Interrupt* is a type of signal from Hardware/Software source.

29 votes -- Siddharth Mahapatra (1.2k points)

5.8.6 Interrupts: GATE CSE 2001 | Question: 1.12 top https://gateoverflow.in/705

✔ Answer is (**C**).

(A) and **(B)** are obviously incorrect. In **(D)** no need to change mode while returning from any subroutine. therefore software interrupt is not needed for that. But in **(C)** to execute any privileged instruction processor needs software interrupt while changing mode.

37 votes

-- jayendra (6.7k points)

5.8.7 Interrupts: GATE CSE 2011 | Question: 11 top

https://gateoverflow.in/2113

✔ Answer should be (**D**) Higher priority interrupt levels are assigned to requests which, if delayed or interrupted,could have serious consequences. Devices with high speed transfer such as magnetic disks are given high priority, and slow devices such as keyboard receive low priority. We know that mouse pointer movements are more frequent than keyboard ticks. So its obvious that its data transfer rate is higher than keyboard. Delaying a CPU temperature sensor could have serious consequences, overheat can damage CPU circuitry. From the above information we can conclude that priorities are-

CPU temperature sensor > Hard Disk > Mouse > Keyboard

54 votes

-- Tejas Jaiswal (559 points)

5.8.8 Interrupts: GATE CSE 2018 | Question: 9 top

https://gateoverflow.in/204083

✔ Answer should be **A**.

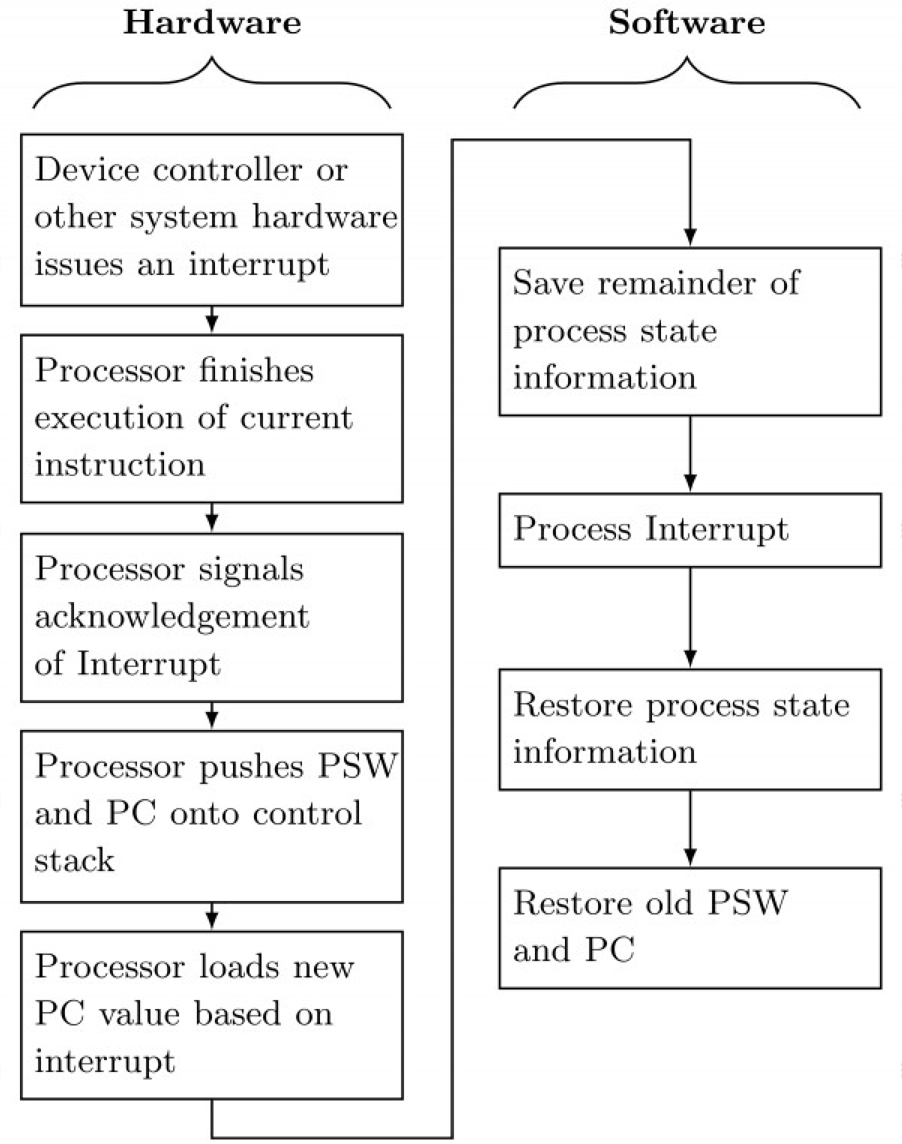


Fig: ***Simple Interrupt Processing***

37 votes

-- Ayush Upadhyaya (28.4k points)

5.9 Io Handling (6) top

5.9.1 Io Handling: GATE CSE 1996 | Question: 1.20, ISRO2008-56 top

https://gateoverflow.in/2724

Which of the following is an example of spooled device?

A. A line printer used to print the output of a number of jobs
B. A terminal used to enter input data to a running program
C. A secondary storage device in a virtual memory system
D. A graphic display device

gate1996 operating-system io-handling normal isro2008

Answer

5.9.2 Io Handling: GATE CSE 1998 | Question: 1.29 top https://gateoverflow.in/1666

Which of the following is an example of a spooled device?

A. The terminal used to enter the input data for the C program being executed
B. An output device used to print the output of a number of jobs
C. The secondary memory device in a virtual storage system
D. The swapping area on a disk used by the swapper

gate1998 operating-system io-handling easy

Answer

5.9.3 Io Handling: GATE CSE 2005 | Question: 19 top https://gateoverflow.in/1355

Which one of the following is true for a CPU having a single interrupt request line and a single interrupt grant line?

A. Neither vectored interrupt nor multiple interrupting devices are possible
B. Vectored interrupts are not possible but multiple interrupting devices are possible
C. Vectored interrupts and multiple interrupting devices are both possible
D. Vectored interrupts are possible but multiple interrupting devices are not possible

gate2005-cse operating-system io-handling normal

Answer

5.9.4 Io Handling: GATE CSE 2005 | Question: 20 top https://gateoverflow.in/1356

Normally user programs are prevented from handling I/O directly by I/O instructions in them. For CPUs having explicit I/O instructions, such I/O protection is ensured by having the I/O instruction privileged. In a CPU with memory mapped I/O, there is no explicit I/O instruction. Which one of the following is true for a CPU with memory mapped I/O?

A. I/O protection is ensured by operating system routine(s)
B. I/O protection is ensured by a hardware trap
C. I/O protection is ensured during system configuration
D. I/O protection is not possible

gate2005-cse operating-system io-handling normal

Answer

5.9.5 Io Handling: GATE IT 2004 | Question: 11, ISRO2011-33 top https://gateoverflow.in/3652

What is the bit rate of a video terminal unit with 80 characters/line, 8 bits/character and horizontal sweep time of 100 μs (including 20 μs of retrace time)?

A. 8 Mbps
B. 6.4 Mbps
C. 0.8 Mbps
D. 0.64 Mbps

gate2004-it operating-system io-handling easy isro2011

Answer

5.9.6 Io Handling: GATE IT 2006 | Question: 8 top https://gateoverflow.in/3547

Which of the following DMA transfer modes and interrupt handling mechanisms will enable the highest I/O band-width?

A. Transparent DMA and Polling interrupts

B. Cycle-stealing and Vectored interrupts
C. Block transfer and Vectored interrupts
D. Block transfer and Polling interrupts

gate2006-it operating-system io-handling dma normal

Answer

Answers: Io Handling

5.9.1 Io Handling: GATE CSE 1996 | Question: 1.20, ISRO2008-56 top

https://gateoverflow.in/2724

✔ Answer is (**A**).

Spooling(simultaneous peripheral operations online) is a technique in which an intermediate device such as disk is interposed between process and low speed i/o device. For ex. in printer if a process attempt to print a document but printer is busy printing another document, the process, instead of waiting for printer to become available,write its output to disk. When the printer become available the data on disk is printed. Spooling allows process to request operation from peripheral device without requiring that the device be ready to service the request.

50 votes

-- neha pawar (3.3k points)

5.9.2 Io Handling: GATE CSE 1998 | Question: 1.29 top

https://gateoverflow.in/1666

✔ Answer : Option (B)

SPOOLing (Simultaneous Peripheral Operations OnLine) is a technique in which an intermediate device such as disk is interposed between process and low speed I/O device like a printer. If a process attempts to print a document but printer is busy printing another document, the process, instead of waiting for printer to become available, write its output to disk. When the printer become available the data on disk is printed. Spooling allows process to request operations from peripheral devices without requiring that the device be ready to service the request.

20 votes

-- Tilak D. Nanavati (2.9k points)

5.9.3 Io Handling: GATE CSE 2005 | Question: 19 top

https://gateoverflow.in/1355

✔ (**C**) is the correct answer. We can use one Interrupt line for all the devices connected and pass it through OR gate. On receiving by the CPU, it executes the corresponding ISR and after exec INTA is sent via one line. For Vectored Interrupts it is always possible if we implement in daisy chain mechanism.

Ref : [Click Here]

References

29 votes

-- confused_luck (741 points)

5.9.4 Io Handling: GATE CSE 2005 | Question: 20 top

https://gateoverflow.in/1356

✔ Option (**A**). User applications are not allowed to perform I/O in user mode - All I/O requests are handled through system calls that must be performed in kernel mode.

45 votes

-- Vikrant Singh (11.2k points)

5.9.5 Io Handling: GATE IT 2004 | Question: 11, ISRO2011-33 top https://gateoverflow.in/3652

✔ Answer: (**B**)

Bit rate of a video terminal unit $= 80 \times 8$ bits$/100\mu s = 6.4$ Mbps

22 votes -- Rajarshi Sarkar (27.9k points)

5.9.6 Io Handling: GATE IT 2006 | Question: 8 top https://gateoverflow.in/3547

✔ CPU get highest bandwidth in transparent DMA and polling. but it asked for I/O bandwidth not cpu bandwidth so option (A) is wrong.

In case of Cycle stealing, in each cycle time device send data then wait again after few CPU cycle it sends to memory . So option (B) is wrong.

In case of Polling CPU takes the initiative so I/O bandwidth can not be high so option (D) is wrong .

Consider Block transfer, in each single block device send data so bandwidth (means the amount of data) must be high . This makes option (**C**) correct.

38 votes -- Bikram (58.4k points)

5.10 Memory Management (9) top

5.10.1 Memory Management: GATE CSE 1992 | Question: 12-b top https://gateoverflow.in/43582

Let the page reference and the working set window be $c\ c\ d\ b\ c\ e\ c\ e\ a\ d$ and 4, respectively. The initial working set at time $t = 0$ contains the pages $\{a, d, e\}$, where a was referenced at time $t = 0$, d was referenced at time $t = -1$, and e was referenced at time $t = -2$. Determine the total number of page faults and the average number of page frames used by computing the working set at each reference.

gate1992 operating-system memory-management normal descriptive

Answer

5.10.2 Memory Management: GATE CSE 1995 | Question: 5 top https://gateoverflow.in/2641

A computer installation has $1000k$ of main memory. The jobs arrive and finish in the following sequences.


```
Job 1 requiring 200k arrives
Job 2 requiring 350k arrives
Job 3 requiring 300k arrives
Job 1 finishes
Job 4 requiring 120k arrives
Job 5 requiring 150k arrives
Job 6 requiring 80k arrives
```

A. Draw the memory allocation table using Best Fit and First Fit algorithms.

B. Which algorithm performs better for this sequence?

gate1995 operating-system memory-management normal descriptive

Answer

5.10.3 Memory Management: GATE CSE 1996 | Question: 2.18 top https://gateoverflow.in/2747

A 1000 Kbyte memory is managed using variable partitions but no compaction. It currently has two partitions of sizes 200 Kbyte and 260 Kbyte respectively. The smallest allocation request in Kbyte that could be denied is for

A. 151
B. 181
C. 231
D. 541

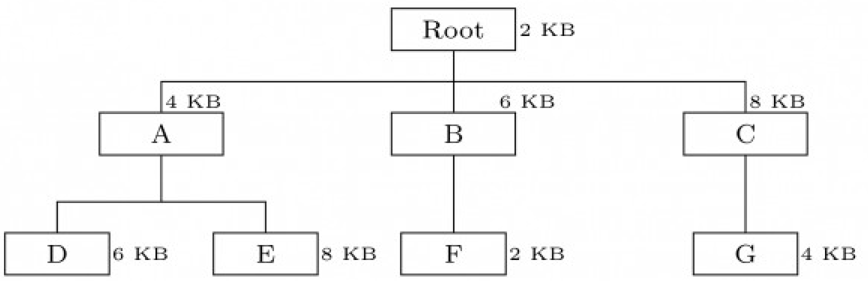
gate1996 operating-system memory-management normal

Answer

5.10.4 Memory Management: GATE CSE 1998 | Question: 2.16 top

https://gateoverflow.in/1689

The overlay tree for a program is as shown below:



What will be the size of the partition (in physical memory) required to load (and run) this program?

A. 12 KB
B. 14 KB
C. 10 KB
D. 8 KB

gate1998 operating-system normal memory-management

Answer

5.10.5 Memory Management: GATE CSE 2014 Set 2 | Question: 55 top

https://gateoverflow.in/2022

Consider the main memory system that consists of 8 memory modules attached to the system bus, which is one word wide. When a write request is made, the bus is occupied for 100 nanoseconds (ns) by the data, address, and control signals. During the same 100 ns, and for 500 ns thereafter, the addressed memory module executes one cycle accepting and storing the data. The (internal) operation of different memory modules may overlap in time, but only one request can be on the bus at any time. The maximum number of stores (of one word each) that can be initiated in 1 millisecond is ________

gate2014-cse-set2 operating-system memory-management numerical-answers normal

Answer

5.10.6 Memory Management: GATE CSE 2015 Set 2 | Question: 30 top

https://gateoverflow.in/8145

Consider 6 memory partitions of sizes 200 KB, 400 KB, 600 KB, 500 KB, 300 KB and 250 KB, where KB refers to kilobyte. These partitions need to be allotted to four processes of sizes 357 KB, 210 KB, 468 KB, 491 KB in that order. If the best-fit algorithm is used, which partitions are NOT allotted to any process?

A. 200 KB and 300 KB
B. 200 KB and 250 KB
C. 250 KB and 300 KB
D. 300 KB and 400 KB

gate2015-cse-set2 operating-system memory-management easy

Answer

5.10.7 Memory Management: GATE CSE 2020 | Question: 11 top

https://gateoverflow.in/333220

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 in any of the existing holes. Which one of the following statement is TRUE?

A. The hole created by first fit is always larger than the hole created by next fit.
B. The hole created by worst fit is always larger than the hole created by first fit.
C. The hole created by best fit is never larger than the hole created by first fit.
D. The hole created by next fit is never larger than the hole created by best fit.

gate2020-cse operating-system memory-management

Answer

5.10.8 Memory Management: GATE IT 2006 | Question: 56 top

https://gateoverflow.in/3600

For each of the four processes P_1, P_2, P_3, and P_4. The total size in kilobytes (KB) and the number of segments are given below.

Process	Total size (in KB)	Number of segments
P_1	195	4
P_2	254	5
P_3	45	3
P_4	364	8

The page size is 1 KB. The size of an entry in the page table is 4 bytes. The size of an entry in the segment table is 8 bytes. The maximum size of a segment is 256 KB. The paging method for memory management uses two-level paging, and its storage overhead is P. The storage overhead for the segmentation method is S. The storage overhead for the segmentation and paging method is T. What is the relation among the overheads for the different methods of memory management in the concurrent execution of the above four processes?

A. P < S < T
B. S < P < T
C. S < T < P
D. T < S < P

gate2006-it operating-system memory-management difficult

Answer

5.10.9 Memory Management: GATE IT 2007 | Question: 11 top

https://gateoverflow.in/3444

Let a memory have four free blocks of sizes $4k$, $8k$, $20k$, $2k$. These blocks are allocated following the best-fit strategy. The allocation requests are stored in a queue as shown below.

Request No	J1	J2	J3	J4	J5	J6	J7	J8
Request Sizes	2k	14k	3k	6k	6k	10k	7k	20k
Usage Time	4	10	2	8	4	1	8	6

The time at which the request for $J7$ will be completed will be

A. 16
B. 19
C. 20
D. 37

gate2007-it operating-system memory-management normal

Answer

Answers: Memory Management

5.10.1 Memory Management: GATE CSE 1992 | Question: 12-b top

https://gateoverflow.in/43582

✔ Window size of working set $= 4$

Initial pages in the working set window $= \{e, d, a\}$

Incoming page	Time	Working set window	Hit/ Miss	Current window size
c	1	$\{e, d, a, c\}$	**miss**	4
c	2	$\{d, a, c\}$	hit	3
d	3	$\{a, c, d\}$	hit	3
b	4	$\{c, d, b\}$	**miss**	3
c	5	$\{d, b, c\}$	hit	3
e	6	$\{d, b, c, e\}$	**miss**	4
c	7	$\{b, c, e\}$	hit	3
e	8	$\{c, e\}$	hit	2
a	9	$\{c, e, a\}$	**miss**	3
d	10	$\{c, e, a, d\}$	**miss**	4

Total number of page faults $= \mathbf{5}$.

Average no. of page frames used by window set $= (4+3+3+3+3+4+3+2+3+4)/10 = 32/10 = \mathbf{3.2}$

51 votes

-- Dhananjay Kumar Sharma (18.8k points)

5.10.2 Memory Management: GATE CSE 1995 | Question: 5 top

https://gateoverflow.in/2641

✔ Initial there is $1000k$ main memory available.

Then job 1 arrive and occupied $200k$, then job 2 arrive, occupy $350k$, after that job 3 arrive and occupy $300k$ (assume continuous allocation) now free memory is $1000 - 850(200 + 350 + 300) = 150k$ (till these jobs first fit and best fit are same)

Now, job 1 is finished. So, that space is also free. So, here $200k$ slot and $150k$ slots are free.

Now, job 4 arrives which is $120k$.

Case 1:

- First fit, so it will be in 200 k slot (free slot) and now free is $= 200 - 120 = 80k$,
- Now $150k$ arrive which will be in 150 k slot
- Then, $80k$ arrive which will occupy in $80k$ slot $(200 - 120)$ so, all jobs will be allocated successfully.

Case 2:

- Best fit : $120k$ job will occupy best fit free space which is $150k$ so, now remaining $150 - 120 = 30k$,
- Then $150k$ job arrive it will be occupied in $200k$ slot, which is best fit for this job. So, free space $= 200 - 150 = 50$,
- Now, job $80k$ arrive, but there is no continuous $80k$ memory free. So, it will not be allocated successfully.

So, first fit is better.

27 votes

-- minal (13.1k points)

5.10.3 Memory Management: GATE CSE 1996 | Question: 2.18 top

https://gateoverflow.in/2747

✔ The answer is (**B**). Since the total size of the memory is 1000 KB, let's assume that the partitioning for the current allocation is done in such a way that it will leave minimum free space.

Partitioning the 1000 KB as below will allow gaps of 180 KB each and hence a request of 181 KB will not be met.

$[180 \text{ KB} - 200 \text{ KB} - 180 \text{ KB} - 260 \text{ KB} - 180 \text{ KB}]$. The reasoning is more of an intuition rather than any formula.

70 votes

-- kireeti (1k points)

5.10.4 Memory Management: GATE CSE 1998 | Question: 2.16 top

https://gateoverflow.in/1689

✔ "To enable a process to be larger than the amount of memory allocated to it, we can use overlays. The idea of overlays is to keep in memory only those instructions and data that are needed at any given time. When other instructions are needed, they are loaded into space occupied previously by instructions that are no longer needed." For the above program, maximum memory will be required when running code portion present at leaves. Max requirement $=$ (max of requirements of D, E, F, and G.
$= MAX(12, 14, 10, 14) = 14$ **(Answer)**

👍 44 votes -- learncp (1.1k points)

5.10.5 Memory Management: GATE CSE 2014 Set 2 | Question: 55 top https://gateoverflow.in/2022

✔ When a write request is made, the bus is occupied for 100 ns. So, between 2 writes at least 100 ns interval must be there.

Now, after a write request, for $100 + 500 = 600$ ns, the corresponding memory module is busy storing the data. But, assuming the next stores are to a different memory module (we have totally 8 modules in question), we can have consecutive stores at intervals of 100 ns. So, maximum number of stores in 1 ms

$= 10^{-3} \times 1/(100 \times 10^{-9}) = 10,000$

👍 73 votes -- Arjun Suresh (332k points)

5.10.6 Memory Management: GATE CSE 2015 Set 2 | Question: 30 top https://gateoverflow.in/8145

✔ Option **(A)** is correct because we have 6 memory partitions of sizes 200 KB, 400 KB, 600 KB, 500 KB, 300 KB and 250 KB and the partition allotted to the process using best fit is given below:

357 KB process allotted at partition 400 KB.
210 KB process allotted at partition 250 KB
468 KB process allotted at partition 500 KB
491 KB process allotted at partition 600 KB

So, we have left only two partitions 200 KB and 300 KB

👍 30 votes -- Anoop Sonkar (4.1k points)

5.10.7 Memory Management: GATE CSE 2020 | Question: 11 top https://gateoverflow.in/333220

✔ Best fit will search for the smallest block which is able to accommodate the request. So, the hole created by the Best Fit is always **less than or equal** to the hole created using any other method.

Worst fit search for the biggest possible block which is able to accommodate the request. It might be the case that block biggest possible block may be in the first block and both worst and first fit select the same block.

So, we can't say that hole formed by worst fit is always greater than first. The size of the hole can be same too. (B) is false

Ans: (C) Hole created by the best fit is never larger than the hole created by first fit,

The hole created by the Best Fit is equal to the hole created by first fit when the first fit happens to select the smallest block which can accommodate the required size.

👍 13 votes -- Srinivas_Reddy_Kotla (775 points)

5.10.8 Memory Management: GATE IT 2006 | Question: 56 top https://gateoverflow.in/3600

For 2-level paging.

Page size is $1KB$. So, no. of pages required for $P_1 = 195$. An entry in page table is of size 4 bytes and assuming an inner level page table takes the size of a page (this information is not given in question), we can have up to 256 entries in a second level page table and we require only 195 for P_1. Thus only 1 second level page table is enough. So, memory overhead $= 1KB$ (for first level) (again assumed as page size as not explicitly told in question) $+1KB$ for second level $= 2KB$.

For P_2 and P_3 also, we get $2KB$ each and for P_4 we get $1 + 2 = 3KB$ as it requires 1 first level page table and 2 second level page tables $(364 > 256)$. So, total overhead for their concurrent execution $= 2 \times 3 + 3 = 9KB$.

Thus $P = 9KB$.

For Segmentation method

Ref: http://web.cs.wpi.edu/~cs3013/b02/week6-segmentation/week6-segmentation.html

P_1 uses 4 segments $\rightarrow$ 4 entries in segment table $= 4 \times 8 = 32$ bytes.

Similarly, for P_2, P_3 and P_4 we get 5×8, 3×8 and 8×8 bytes respectively and the total overhead will be $32 + 40 + 24 + 64 = 160$ bytes.

So, $S = 160B$.

For Segmentation with Paging

Here we segment first and then page. So, we need the page table size. We are given maximum size of a segment is 256 KB and page size is $1KB$ and thus we require 256 entries in the page table. So, total size of page table $= 256 \times 4 = 1024$ bytes (exactly 1 page size).

So, now for P_1 we require 1 segment table of size 32 bytes plus 4 page table of size $1KB$ for the 4 segments. Similarly,

$P_2 - 40$ bytes and 5 KB
$P_3 - 24$ bytes and 3 KB
$P_4 - 64$ bytes and 8 KB.

Thus total overhead $= 160$ bytes $4\ KB + 5\ KB + 3\ KB + 8\ KB = 20480 + 160 = 20640$ bytes.

So, $T = 20640B$.

So, answer will be (**B**)- $S < P < T$.

References

79 votes

-- Arjun Suresh (332k points)

5.10.9 Memory Management: GATE IT 2007 | Question: 11 top

https://gateoverflow.in/3444

✔ PS: Since the block sizes are given, we cannot assume further splitting of them.

Also, the question implies a multiprocessing environment and we can assume the execution of a process is not affecting other process' runtime.

At t=0

Memory Block	Size	Job
A	4k	J3 (finishes at t = 2)
B	8k	J4 (finishes at t = 8)
C	20k	J2 (finishes at t = 10)
D	2k	J1 (finishes at t=4)

At t=8

Memory Block	Size	Job
A	4k	
B	8k	J5 (finishes at t=12)
C	20k	J2 (finishes at t = 10)
D	2k	

At t=10

Memory Block	Size	Job
A	4k	
B	8k	J5 (finishes at t=12)
C	20k	J6 (finishes at t = 11)
D	2k	

At t=11

Memory Block	Size	Job
A	4k	
B	8k	J5 (finishes at t=12)
C	20k	J7 (finishes at t = 19)
D	2k	

So, $J7$ finishes at $t = 19$.

Reference: http://thumbsup2life.blogspot.fr/2011/02/best-fit-first-fit-and-worst-fit-memory.html

Correct Answer: B

References

57 votes

-- Arjun Suresh (332k points)

5.11 Os Protection (3) top

5.11.1 Os Protection: GATE CSE 1999 | Question: 1.11, UGCNET-Dec2015-II: 44 top

https://gateoverflow.in/1464

System calls are usually invoked by using

A. a software interrupt
B. polling
C. an indirect jump
D. a privileged instruction

gate1999 operating-system normal ugcnetdec2015ii os-protection

Answer

5.11.2 Os Protection: GATE CSE 2001 | Question: 1.13 top

https://gateoverflow.in/706

A CPU has two modes -- privileged and non-privileged. In order to change the mode from privileged to non-privileged

A. a hardware interrupt is needed
B. a software interrupt is needed
C. a privileged instruction (which does not generate an interrupt) is needed
D. a non-privileged instruction (which does not generate an interrupt) is needed

gate2001-cse operating-system normal os-protection

Answer

5.11.3 Os Protection: GATE IT 2005 | Question: 19, UGCNET-June2012-III: 57 top

https://gateoverflow.in/3764

A user level process in Unix traps the signal sent on a Ctrl-C input, and has a signal handling routine that saves appropriate files before terminating the process. When a Ctrl-C input is given to this process, what is the mode in which the signal handling routine executes?

A. User mode
B. Kernel mode
C. Superuser mode
D. Privileged mode

gate2005-it operating-system os-protection normal ugcnetjune2012iii

Answer

Answers: Os Protection

5.11.1 Os Protection: GATE CSE 1999 | Question: 1.11, UGCNET-Dec2015-II: 44 top

https://gateoverflow.in/1464

✔ Software interrupt is the answer.

Privileged instruction cannot be the answer as system call is done from user mode and privileged instruction cannot be done from user mode.

44 votes

-- Arjun Suresh (332k points)

5.11.2 Os Protection: GATE CSE 2001 | Question: 1.13 top https://gateoverflow.in/706

Answer should be (**D**). Changing from privileged to non-privileged doesn't require an interrupt unlike from non-privileged to privileged. Also, to loose a privilege we don't need a privileged instruction though a privileged instruction does no harm.

http://web.cse.ohio-state.edu/~teodores/download/teaching/cse675.au08/CSE675.02_MIPS-ISA_part3.pdf

References

64 votes -- Arjun Suresh (332k points)

5.11.3 Os Protection: GATE IT 2005 | Question: 19, UGCNET-June2012-III: 57 top https://gateoverflow.in/3764

- When an user send an input to the process it can not be in privileged mode as it is coming from an user so option D , Privileged mode can not be possible here ..

Now see , kernel mode = Privileged mode

- That means both option B and option D are equal. As option D can not be possible , option B also false.
- There is nothing called superuser mode so option C is clearly wrong .
- Only option A is left , when an user input come like ' ctrl+c' the signal handling routine **executes in user mode only** as **a user level** process in UNIX traps the signal.

Hence option A is correct answer.

37 votes -- Bikram (58.4k points)

5.12 Page Replacement (30) top

5.12.1 Page Replacement: GATE CSE 1993 | Question: 21 top https://gateoverflow.in/2318

The following page addresses, in the given sequence, were generated by a program:

1 2 3 4 1 3 5 2 1 5 4 3 2 3

This program is run on a demand paged virtual memory system, with main memory size equal to 4 pages. Indicate the page references for which page faults occur for the following page replacement algorithms.

A. LRU
B. FIFO

Assume that the main memory is initially empty.

gate1993 operating-system page-replacement normal descriptive

Answer

5.12.2 Page Replacement: GATE CSE 1994 | Question: 1.13 top https://gateoverflow.in/2454

A memory page containing a heavily used variable that was initialized very early and is in constant use is removed then

A. LRU page replacement algorithm is used
B. FIFO page replacement algorithm is used
C. LFU page replacement algorithm is used
D. None of the above

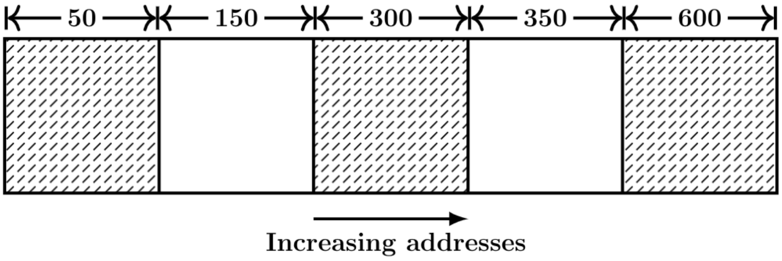
gate1994 operating-system page-replacement easy

Answer

5.12.3 Page Replacement: GATE CSE 1994 | Question: 1.24 top

https://gateoverflow.in/2467

Consider the following heap (figure) in which blank regions are not in use and hatched region are in use.



The sequence of requests for blocks of sizes $300, 25, 125, 50$ can be satisfied if we use

A. either first fit or best fit policy (any one)

B. first fit but not best fit policy

C. best fit but not first fit policy

D. None of the above

gate1994 operating-system page-replacement normal

Answer

5.12.4 Page Replacement: GATE CSE 1995 | Question: 1.8 top

https://gateoverflow.in/2595

Which of the following page replacement algorithms suffers from Belady's anamoly?

A. Optimal replacement

B. LRU

C. FIFO

D. Both (A) and (C)

gate1995 operating-system page-replacement normal

Answer

5.12.5 Page Replacement: GATE CSE 1995 | Question: 2.7 top

https://gateoverflow.in/2619

The address sequence generated by tracing a particular program executing in a pure demand based paging system with 100 records per page with 1 free main memory frame is recorded as follows. What is the number of page faults?

$0100, 0200, 0430, 0499, 0510, 0530, 0560, 0120, 0220, 0240, 0260, 0320, 0370$

A. 13
B. 8
C. 7
D. 10

gate1995 operating-system page-replacement normal

Answer

5.12.6 Page Replacement: GATE CSE 1997 | Question: 3.10, ISRO2008-57, ISRO2015-64 top

https://gateoverflow.in/2241

Dirty bit for a page in a page table

A. helps avoid unnecessary writes on a paging device
B. helps maintain LRU information
C. allows only read on a page
D. None of the above

gate1997 operating-system page-replacement easy isro2008 isro2015

Answer

5.12.7 Page Replacement: GATE CSE 1997 | Question: 3.5 top https://gateoverflow.in/2236

Locality of reference implies that the page reference being made by a process

A. will always be to the page used in the previous page reference
B. is likely to be to one of the pages used in the last few page references
C. will always be to one of the pages existing in memory
D. will always lead to a page fault

gate1997 operating-system page-replacement easy

Answer

5.12.8 Page Replacement: GATE CSE 1997 | Question: 3.9 top https://gateoverflow.in/2240

Thrashing

A. reduces page I/O
B. decreases the degree of multiprogramming
C. implies excessive page I/O
D. improve the system performance

gate1997 operating-system page-replacement easy

Answer

5.12.9 Page Replacement: GATE CSE 2001 | Question: 1.21 top https://gateoverflow.in/714

Consider a virtual memory system with FIFO page replacement policy. For an arbitrary page access pattern, increasing the number of page frames in main memory will

A. always decrease the number of page faults
B. always increase the number of page faults
C. sometimes increase the number of page faults
D. never affect the number of page faults

gate2001-cse operating-system page-replacement normal

Answer

5.12.10 Page Replacement: GATE CSE 2002 | Question: 1.23 top https://gateoverflow.in/828

The optimal page replacement algorithm will select the page that

A. Has not been used for the longest time in the past
B. Will not be used for the longest time in the future
C. Has been used least number of times
D. Has been used most number of times

gate2002-cse operating-system page-replacement easy

Answer

5.12.11 Page Replacement: GATE CSE 2004 | Question: 21, ISRO2007-44 top https://gateoverflow.in/1018

The minimum number of page frames that must be allocated to a running process in a virtual memory environment is determined by

A. the instruction set architecture
B. page size
C. number of processes in memory
D. physical memory size

gate2004-cse operating-system virtual-memory page-replacement normal isro2007

Answer

5.12.12 Page Replacement: GATE CSE 2005 | Question: 22, ISRO2015-36 top

https://gateoverflow.in/1358

Increasing the RAM of a computer typically improves performance because:

A. Virtual Memory increases
B. Larger RAMs are faster
C. Fewer page faults occur
D. Fewer segmentation faults occur

gate2005-cse operating-system page-replacement easy isro2015

Answer

5.12.13 Page Replacement: GATE CSE 2007 | Question: 56 top

https://gateoverflow.in/1254

A virtual memory system uses First In First Out (FIFO) page replacement policy and allocates a fixed number of frames to a process. Consider the following statements:

P: Increasing the number of page frames allocated to a process sometimes increases the page fault rate.

Q: Some programs do not exhibit locality of reference.

Which one of the following is TRUE?

A. Both P and Q are true, and Q is the reason for P
B. Both P and Q are true, but Q is not the reason for P.
C. P is false but Q is true
D. Both P and Q are false.

gate2007-cse operating-system page-replacement normal

Answer

5.12.14 Page Replacement: GATE CSE 2007 | Question: 82 top

https://gateoverflow.in/1274

A process has been allocated 3 page frames. Assume that none of the pages of the process are available in the memory initially. The process makes the following sequence of page references (reference string): $\mathbf{1, 2, 1, 3, 7, 4, 5, 6, 3, 1}$

If optimal page replacement policy is used, how many page faults occur for the above reference string?

A. 7
B. 8
C. 9
D. 10

gate2007-cse operating-system page-replacement normal

Answer

5.12.15 Page Replacement: GATE CSE 2007 | Question: 83 top

https://gateoverflow.in/43510

A process, has been allocated 3 page frames. Assume that none of the pages of the process are available in the memory initially. The process makes the following sequence of page references (reference string): $1, 2, 1, 3, 7, 4, 5, 6, 3, 1$

Least Recently Used (LRU) page replacement policy is a practical approximation to optimal page replacement. For the above reference string, how many more page faults occur with LRU than with the optimal page replacement policy?

A. 0
B. 1
C. 2
D. 3

gate2007-cse normal operating-system page-replacement

Answer

5.12.16 Page Replacement: GATE CSE 2009 | Question: 9, ISRO2016-52 top https://gateoverflow.in/1301

In which one of the following page replacement policies, Belady's anomaly may occur?

A. FIFO
B. Optimal
C. LRU
D. MRU

gate2009-cse operating-system page-replacement normal isro2016

Answer

5.12.17 Page Replacement: GATE CSE 2010 | Question: 24 top https://gateoverflow.in/2203

A system uses FIFO policy for system replacement. It has 4 page frames with no pages loaded to begin with. The system first accesses 100 distinct pages in some order and then accesses the same 100 pages but now in the reverse order. How many page faults will occur?

A. 196
B. 192
C. 197
D. 195

gate2010-cse operating-system page-replacement normal

Answer

5.12.18 Page Replacement: GATE CSE 2012 | Question: 42 top https://gateoverflow.in/2150

Consider the virtual page reference string

$$1, 2, 3, 2, 4, 1, 3, 2, 4, 1$$

on a demand paged virtual memory system running on a computer system that has main memory size of 3 page frames which are initially empty. Let LRU, FIFO and OPTIMAL denote the number of page faults under the corresponding page replacement policy. Then

A. $\text{OPTIMAL} < \text{LRU} < \text{FIFO}$
B. $\text{OPTIMAL} < \text{FIFO} < \text{LRU}$
C. $\text{OPTIMAL} = \text{LRU}$
D. $\text{OPTIMAL} = \text{FIFO}$

gate2012-cse operating-system page-replacement normal

Answer

5.12.19 Page Replacement: GATE CSE 2014 Set 1 | Question: 33 top https://gateoverflow.in/1805

Assume that there are 3 page frames which are initially empty. If the page reference string is $1, 2, 3, 4, 2, 1, 5, 3, 2, 4, 6$ the number of page faults using the optimal replacement policy is__________.

gate2014-cse-set1 operating-system page-replacement numerical-answers

Answer

5.12.20 Page Replacement: GATE CSE 2014 Set 2 | Question: 33 top https://gateoverflow.in/1992

A computer has twenty physical page frames which contain pages numbered 101 through 120. Now a program accesses the pages numbered 1, 2, …, 100 in that order, and repeats the access sequence **THRICE**. Which one of the following page replacement policies experiences the same number of page faults as the optimal page replacement policy for this program?

A. Least-recently-used
B. First-in-first-out
C. Last-in-first-out
D. Most-recently-used

gate2014-cse-set2 operating-system page-replacement ambiguous

Answer

5.12.21 Page Replacement: GATE CSE 2014 Set 3 | Question: 20 top https://gateoverflow.in/2054

A system uses 3 page frames for storing process pages in main memory. It uses the Least Recently Used (**LRU**) page replacement policy. Assume that all the page frames are initially empty. What is the total number of page faults that will occur while processing the page reference string given below?

4, 7, 6, 1, 7, 6, 1, 2, 7, 2

gate2014-cse-set3 operating-system page-replacement numerical-answers normal

Answer

5.12.22 Page Replacement: GATE CSE 2015 Set 1 | Question: 47 top https://gateoverflow.in/8353

Consider a main memory with five-page frames and the following sequence of page references: 3, 8, 2, 3, 9, 1, 6, 3, 8, 9, 3, 6, 2, 1, 3 . Which one of the following is true with respect to page replacement policies First In First Out (FIFO) and Least Recently Used (LRU)?

A. Both incur the same number of page faults
B. FIFO incurs 2 more page faults than LRU
C. LRU incurs 2 more page faults than FIFO
D. FIFO incurs 1 more page faults than LRU

gate2015-cse-set1 operating-system page-replacement normal

Answer

5.12.23 Page Replacement: GATE CSE 2016 Set 1 | Question: 49 top https://gateoverflow.in/39711

Consider a computer system with ten physical page frames. The system is provided with an access sequence $(a_1, a_2, \dots, a_{20}, a_1, a_2, \dots a_{20})$, where each a_i is a distinct virtual page number. The difference in the number of page faults between the last-in-first-out page replacement policy and the optimal page replacement policy is_________.

gate2016-cse-set1 operating-system page-replacement normal numerical-answers

Answer

5.12.24 Page Replacement: GATE CSE 2016 Set 2 | Question: 20 top https://gateoverflow.in/39559

In which one of the following page replacement algorithms it is possible for the page fault rate to increase even when the number of allocated frames increases?

A. LRU (Least Recently Used)
B. OPT (Optimal Page Replacement)
C. MRU (Most Recently Used)
D. FIFO (First In First Out)

gate2016-cse-set2 operating-system page-replacement easy

Answer

5.12.25 Page Replacement: GATE CSE 2017 Set 1 | Question: 40 top

https://gateoverflow.in/118323

Recall that Belady's anomaly is that the page-fault rate may *increase* as the number of allocated frames increases. Now, consider the following statements:

- S_1: *Random page replacement* algorithm (where a page chosen at random is replaced) suffers from Belady's anomaly.
- S_2: *LRU page replacement* algorithm suffers from Belady's anomaly.

Which of the following is CORRECT?

A. S_1 is true, S_2 is true
B. S_1 is true, S_2 is false
C. S_1 is false, S_2 is true
D. S_1 is false, S_2 is false

gate2017-cse-set1 page-replacement operating-system normal

Answer

5.12.26 Page Replacement: GATE CSE 2021 Set 1 | Question: 11 top

https://gateoverflow.in/357441

In the context of operating systems, 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. Paging incurs memory overheads
D. Multi-level paging is necessary to support pages of different sizes

gate2021-cse-set1 multiple-selects operating-system page-replacement

Answer

5.12.27 Page Replacement: GATE CSE 2021 Set 2 | Question: 48 top

https://gateoverflow.in/357489

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

$\longleftarrow$ 39 – bit virtual address $\longrightarrow$

Level 1 offset	Level 2 offset	Level 3 offset	Page offset
9 bits	9 bits	9 bits	12 bits

The page size is 4 KB $(1\text{KB} = 2^{10}$ bytes$)$ and page table entry size at every level is 8 bytes. A process P is currently using 2GB $(1\text{GB} = 2^{30}$ 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.

gate2021-cse-set2 numerical-answers operating-system memory-management page-replacement

Answer

5.12.28 Page Replacement: GATE IT 2007 | Question: 12 top

https://gateoverflow.in/3445

The address sequence generated by tracing a particular program executing in a pure demand paging system with 100 bytes per page is

$$0100, 0200, 0430, 0499, 0510, 0530, 0560, 0120, 0220, 0240, 0260, 0320, 0410.$$

Suppose that the memory can store only one page and if x is the address which causes a page fault then the bytes from addresses x to $x + 99$ are loaded on to the memory.
How many page faults will occur?

A. 0
B. 4
C. 7

D. 8

gate2007-it operating-system virtual-memory page-replacement normal

Answer

5.12.29 Page Replacement: GATE IT 2007 | Question: 58 top

https://gateoverflow.in/3500

A demand paging system takes 100 time units to service a page fault and 300 time units to replace a dirty page. Memory access time is 1 time unit. The probability of a page fault is p. In case of a page fault, the probability of page being dirty is also p. It is observed that the average access time is 3 time units. Then the value of p is

A. 0.194
B. 0.233
C. 0.514
D. 0.981

gate2007-it operating-system page-replacement probability normal

Answer

5.12.30 Page Replacement: GATE IT 2008 | Question: 41 top

https://gateoverflow.in/3351

Assume that a main memory with only 4 pages, each of 16 bytes, is initially empty. The CPU generates the following sequence of virtual addresses and uses the Least Recently Used (LRU) page replacement policy.

0, 4, 8, 20, 24, 36, 44, 12, 68, 72, 80, 84, 28, 32, 88, 92

How many page faults does this sequence cause? What are the page numbers of the pages present in the main memory at the end of the sequence?

A. 6 and $1, 2, 3, 4$
B. 7 and $1, 2, 4, 5$
C. 8 and $1, 2, 4, 5$
D. 9 and $1, 2, 3, 5$

gate2008-it operating-system page-replacement normal

Answer

Answers: Page Replacement

5.12.1 Page Replacement: GATE CSE 1993 | Question: 21 top

https://gateoverflow.in/2318

✔ LRU : $1, 2, 3, 4, 5, 2, 4, 3, 2$
FIFO : $1, 2, 3, 4, 5, 1, 2, 3$

17 votes

-- Digvijay (44.9k points)

5.12.2 Page Replacement: GATE CSE 1994 | Question: 1.13 top

https://gateoverflow.in/2454

✔ FIFO replaces a page which was brought into memory first will be removed first so since the variable was initialized very early. it is in the set of first in pages. so it will be removed answer: (B) if you use LRU - since it is used constantly it is a recently used item always. so cannot be removed. If you use LFU - the frequency of the page is more since it is in constant use. So cannot be replaced.

34 votes

-- Sankaranarayanan P.N (8.5k points)

5.12.3 Page Replacement: GATE CSE 1994 | Question: 1.24 top

https://gateoverflow.in/2467

✔ In the first fit, block requests will be satisfied from the first free block that fits it.

- The request for 300 will be satisfied by a 350 size block reducing the free size to 50.
- Request for 25, satisfied by 150 size block, reducing it to 125.
- Request for 125 satisfied by 125 size block.

- And request for 50 satisfied by the 50 size block.

So, all requests can be satisfied.

In the best fit strategy, a block request is satisfied by the smallest block that can fit it.

- The request for 300 will be satisfied by a 350 size block reducing the free size to 50.
- Request for 25, satisfied by 50 size block as its the smallest size that fits 25, reducing it to 25.
- Request for 125, satisfied by 150 size block, reducing it to 25.

Now, the request for 50 cannot be satisfied as the two 25 size blocks are not contiguous.

So, answer **(B).**

31 votes -- Arjun Suresh (332k points)

5.12.4 Page Replacement: GATE CSE 1995 | Question: 1.8 top https://gateoverflow.in/2595

✔ Answer is (**C**).

FIFO sufferes from Belady's anomaly. Optimal replacement never suffers from Belady's anomaly.

17 votes -- jayendra (6.7k points)

5.12.5 Page Replacement: GATE CSE 1995 | Question: 2.7 top https://gateoverflow.in/2619

✔

- $0100 - 1$ page fault. Records $0100 - 0199$ in memory
- $0200 - 2$ page faults. Records $0200 - 0299$ in memory
- $0430 - 3$ page faults. Records $0400 - 0499$ in memory
- $0499 - 3$ page faults. Records $0400 - 0499$ in memory
- $0510 - 4$ page faults. Records $0500 - 0599$ in memory
- $0530 - 4$ page faults. Records $0500 - 0599$ in memory
- $0560 - 4$ page faults. Records $0500 - 0599$ in memory
- $0120 - 5$ page faults. Records $0100 - 0199$ in memory
- $0220 - 6$ page faults. Records $0200 - 0299$ in memory
- $0240 - 6$ page faults. Records $0200 - 0299$ in memory
- $0260 - 6$ page faults. Records $0200 - 0299$ in memory
- $0320 - 7$ page faults. Records $0300 - 0399$ in memory
- $0370 - 7$ page faults. Records $0300 - 0399$ in memory

So, (C) - 7 page faults.

55 votes -- Arjun Suresh (332k points)

5.12.6 Page Replacement: GATE CSE 1997 | Question: 3.10, ISRO2008-57, ISRO2015-64 top https://gateoverflow.in/2241

✔ The dirty bit allows for a performance optimization. A page on disk that is paged in to physical memory, then read from, and subsequently paged out again does not need to be written back to disk, since the page hasn't changed. However, if the page was written to after it's paged in, its dirty bit will be set, indicating that the page must be written back to the backing store answer: (**A**)

52 votes -- Sankaranarayanan P.N (8.5k points)

5.12.7 Page Replacement: GATE CSE 1997 | Question: 3.5 top https://gateoverflow.in/2236

✔ Answer is (**B**)

Locality of reference is also called as principle of locality. It means that same data values or related storage locations are frequently accessed. This in turn saves time. There are mainly three types of principle of locality:

1. temporal locality
2. spatial locality
3. sequential locality

This is required because in programs related data are stored in consecutive locations and in loops same locations are referred again and again

28 votes

-- Neeraj7375 (1.1k points)

5.12.8 Page Replacement: GATE CSE 1997 | Question: 3.9 top

https://gateoverflow.in/2240

✔ (**C**)- implies excessive page I/O

http://en.wikipedia.org/wiki/Thrashing_%28computer_science%29

References

28 votes

-- Sankaranarayanan P.N (8.5k points)

5.12.9 Page Replacement: GATE CSE 2001 | Question: 1.21 top

https://gateoverflow.in/714

✔ Answer is (**C**).

Belady **anomaly** is the name given to the phenomenon in which increasing the number of page frames results in an increase in the number of page faults for certain memory access patterns. This phenomenon is commonly experienced when using the First in First Out (FIFO) page replacement algorithm

References

21 votes

-- dheerajkhanna (143 points)

5.12.10 Page Replacement: GATE CSE 2002 | Question: 1.23 top

https://gateoverflow.in/828

✔ Optimal page replacement algorithm will always select the page that will not be used for the longest time in the future for replacement, and that is why the it is called optimal page replacement algorithm. Hence, (**B**) choice.

36 votes

-- Arjun Suresh (332k points)

5.12.11 Page Replacement: GATE CSE 2004 | Question: 21, ISRO2007-44 top

https://gateoverflow.in/1018

✔ Its instruction set architecture .if you have no indirect addressing then you need at least two pages in physical memory. One for instruction (code part) and another for if the data references memory.if there is one level of indirection then you will need at least three pages one for the instruction(code) and another two for the indirect addressing. If there three indirection then minimum 4 frames are allocated.

http://stackoverflow.com/questions/11213013/minimum-page-frames

References

64 votes

-- Prasanna Ranganathan (3.9k points)

5.12.12 Page Replacement: GATE CSE 2005 | Question: 22, ISRO2015-36 top

https://gateoverflow.in/1358

So, answer $\rightarrow$ (C).

1. Virtual Memory increases $\rightarrow$ This option is false. Because Virtual Memory of Computer do not depend on RAM. Virtual Memory concept iteself was introduced so Programs larger than RAM can be executed.
2. Larger RAMs are faster $\rightarrow$ No This option is false. Size of ram does not determine it's speed, Type of ram does, SRAM is faster, DRAM is slower.
3. Fewer page faults occur $\rightarrow$ This is true, more pages can be in Main memory .
4. Fewer segmentation faults occur $\rightarrow$ "Segementation Fault" $\rightarrow$ A **segmentation fault** (aka segfault) is a common condition that causes programs to crash; they are often associated with a file named core . Segfaults are caused by a program trying to read or write an illegal memory location. It is clear that segmentation fault is not related to size of main memory. This is false.

61 votes

-- Akash Kanase (36k points)

5.12.13 Page Replacement: GATE CSE 2007 | Question: 56 top

https://gateoverflow.in/1254

P: Increasing the number of page frames allocated to a process sometimes increases the page fault rate.

This is true,
example : FIFO suffers from Bélády's anomaly which means that on Increasing the number of page frames allocated to a process it may sometimes increase the total number of page faults.

Q: Some programs do not exhibit locality of reference.

This is true : it is easy to write a program which jumps around a lot & which do not exhibit locality of reference.

Example : Assume that array is stored in Row Major order & We are accessing it in column major order.

So, answer is **option (B).** (As there is no relation between P & Q. As it is clear from example, they are independent.)

References

55 votes

-- Akash Kanase (36k points)

5.12.14 Page Replacement: GATE CSE 2007 | Question: 82 top

https://gateoverflow.in/1274

Optimal replacement policy means a page which is "farthest" in the future to be accessed will be replaced next.

Frame 0	1
Frame 1	$\cancel{2}$ $\cancel{7}$ $\cancel{4}$ $\cancel{5}$ 6
Frame 2	3

3 initial page faults for pages $1, 2, 3$ and then for pages $7, 4, 5, 6 \implies 7$ page faults occur.

Answer is **(A).**

18 votes

-- Pooja Palod (24.1k points)

5.12.15 Page Replacement: GATE CSE 2007 | Question: 83 top

https://gateoverflow.in/43510

Using $LRU = 9$ Page Fault

Using Optimal$=$ 7 Page Fault

So, LRU-OPTIMAL $= 2$

Option (C).

17 votes

-- Manoj Kumar (26.7k points)

5.12.16 Page Replacement: GATE CSE 2009 | Question: 9, ISRO2016-52 top

https://gateoverflow.in/1301

✔ It is (**A**).

http://en.wikipedia.org/wiki/B%C3%A9l%C3%A1dy%27s_anomaly

References

17 votes

-- Gate Keeda (15.9k points)

5.12.17 Page Replacement: GATE CSE 2010 | Question: 24 top

https://gateoverflow.in/2203

✔ Answer is (**A**).

When we access 100 distinct page in some order (for example $1, 2, 3 \dots 100$) then total number of page faults $= 100$. At last, the 4 page frames will contain the pages $100, 99, 98$ and 97. When we reverse the string $(100, 99, 98, \dots, 1)$ then first four page accesses will not cause the page fault because they are already present in page frames. But the remaining 96 page accesses will cause 96 page faults. So, total number of page faults $= 100 + 96 = 196$.

35 votes

-- neha pawar (3.3k points)

5.12.18 Page Replacement: GATE CSE 2012 | Question: 42 top

https://gateoverflow.in/2150

✔ Page fault for LRU $= 9$, FIFO $= 6$, OPTIMAL $= 5$

Answer is (**B**).

18 votes

-- Keith Kr (4.5k points)

5.12.19 Page Replacement: GATE CSE 2014 Set 1 | Question: 33 top

https://gateoverflow.in/1805

✔ In Optimal page replacement a page which will be farthest accessed in future will be replaced first.

Here, we have 3 page frames. Since, initially they are empty the first 3 distinct page references will cause page faults.

After 3 distinct page accesses we have : $\underset{\text{Page Frames}}{\begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline\end{array}} \quad \underset{\text{Next Use Order}}{\begin{array}{|c|c|c|}\hline 2 & 1 & 3 \\ \hline\end{array}}$.

Based on the Next Use Order, the next replacement will be 3. Proceeding like this we get

$\underset{\text{Request}}{\boxed{4}} : \underset{\text{Page Frames}}{\begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline\end{array}} \quad \underset{\text{Next Use Order}}{\begin{array}{|c|c|c|}\hline 2 & 1 & 4 \\ \hline\end{array}}$ – Miss.

$\underset{\text{Request}}{\boxed{2}} : \underset{\text{Page Frames}}{\begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline\end{array}} \quad \underset{\text{Next Use Order}}{\begin{array}{|c|c|c|}\hline 2 & 1 & 4 \\ \hline\end{array}}$ – Hit.

$\underset{\text{Request}}{\boxed{1}} : \underset{\text{Page Frames}}{\begin{array}{|c|c|c|}\hline 1 & 2 & 4 \\ \hline\end{array}} \quad \underset{\text{Next Use Order}}{\begin{array}{|c|c|c|}\hline 2 & 4 & 1 \\ \hline\end{array}}$ – Hit.

$\underset{\text{Request}}{\boxed{5}} : \underset{\text{Page Frames}}{\begin{array}{|c|c|c|}\hline 5 & 2 & 4 \\ \hline\end{array}} \quad \underset{\text{Next Use Order}}{\begin{array}{|c|c|c|}\hline 2 & 4 & 5 \\ \hline\end{array}}$ – Miss.

Request		Page Frames	Next Use Order	
$\boxed{3}$	:	3 | 2 | 4	2 | 4 | 3	– Miss.
$\boxed{2}$	:	3 | 2 | 4	4 | 3 | 2	– Hit.
$\boxed{4}$	:	1 | 2 | 4	4 | 3 | 2	– Hit.
$\boxed{6}$	:	1 | 2 | 6	2 | 1 | 6	– Miss.

(When multiple pages are not going to be accessed again in future, replacing **any** of them is allowed in Optimal page replacement algorithm)

Now, counting the misses which includes the 3 initial ones we get number of page faults as $3 + 4 = 7$.

Correct Answer: 7.

6 votes

-- Arjun Suresh (`332k` points)

5.12.20 Page Replacement: GATE CSE 2014 Set 2 | Question: 33 top

https://gateoverflow.in/1992

✔ It will be **(D)** i.e Most-recently-used.

To be clear "repeats the access sequence THRICE" means totally the sequence of page numbers are accessed 4 times though this is not important for the answer here.

If we go optimal page replacement algorithm it replaces the page which will be least used in near future.

Now we have frame size 20 and reference string is

$$1, 2, \ldots, 100, 1, 2, \ldots, 100, 1, 2, \ldots, 100, 1, 2, \ldots, 100$$

First 20 accesses will cause page faults - the initial pages are no longer used and hence optimal page replacement replaces them first. Now, for page 21, according to reference string page 1 will be used again after 100 and similarly 2 will be used after 1 so, on and so the least likely to be used page in future is page 20. So, for 21^{st} reference page 20 will be replaced and then for 22^{nd} page reference, page 21 will be replaced and so on which is MOST RECENTLY USED page replacement policy.

PS: Even for Most Recently Used page replacement at first all empty (invalid) pages frames are replaced and then only most recently used ones are replaced.

67 votes

-- Kalpish Singhal (`1.6k` points)

5.12.21 Page Replacement: GATE CSE 2014 Set 3 | Question: 20 top

https://gateoverflow.in/2054

✔ Total page faults $= 6$.

4	**7**	**6**	**1**	**7**	**6**	**1**	**2**	**7**	**2**
		6 $\boxed{F}$	6	6	6	6	6	7 $\boxed{F}$	7
	7 $\boxed{F}$	7	7	7	7	7	2 $\boxed{F}$	2	2
4 $\boxed{F}$	4	4	1 $\boxed{F}$	1	1	1	1	1	1

$\Longrightarrow$ 6 faults

Another way of answering the same.

6	6	$\cancel{6}$ 7	7
7	$\cancel{7}$ 2	2	2
$\cancel{4}$ 1	1	1	1

$\Longrightarrow$ 3 faults + 3 initial access faults = 6 page faults

OR

$\cancel{6}$ 7
$\cancel{7}$ 2
$\cancel{4}$ 1

$\Longrightarrow$ 3 faults + 3 initial access faults = 6 page faults

👍 22 votes -- Akhil Nadh PC (16.5k points)

5.12.22 Page Replacement: GATE CSE 2015 Set 1 | Question: 47 top https://gateoverflow.in/8353

✔ Requested Page references are $3, 8, 2, 3, 9, 1, 6, 3, 8, 9, 3, 6, 2, 1, 3$ and number of page frames is 5.

In FIFO Page replacement will take place in sequence in pattern First In first Out, as following

Request	**3**	**8**	**2**	**3**	**9**	**1**	**6**	**3**	**8**	**9**	**3**	**6**	**2**	**1**	**3**
Frame 5						1	1	1	1	1	1	1	1	1	1
Frame 4					9	9	9	9	9	9	9	9	2	2	2
Frame 3			2	2	2	2	2	2	8	8	8	8	8	8	8
Frame 2		8	8	8	8	8	8	3	3	3	3	3	3	3	3
Frame 1	3	3	3	3	3	3	6	6	6	6	6	6	6	6	6
Miss/hit	F	F	F	H	F	F	F	F	F	H	H	H	F	H	H

Number of Faults $= 9$. Number of Hits $= 6$

Using Least Recently Used (LRU) page replacement will be the page which is visited least recently (which is not used for the longest time), as following:

Request	**3**	**8**	**2**	**3**	**9**	**1**	**6**	**3**	**8**	**9**	**3**	**6**	**2**	**1**	**3**
Frame 5						1	1	1	1	1	1	1	2	2	2
Frame 4					9	9	9	9	9	9	9	9	9	9	9
Frame 3			2	2	2	2	2	2	8	8	8	8	8	1	1
Frame 2		8	8	8	8	8	6	6	6	6	6	6	6	6	6
Frame 1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Miss/hit	F	F	F	H	F	F	F	H	F	H	H	H	F	F	H

Number of Faults $= 9$. Number of Hits $= 6$

So, both incur the same number of page faults.

Correct Answer: A

👍 31 votes -- Raghuveer Dhakad (1.6k points)

5.12.23 Page Replacement: GATE CSE 2016 Set 1 | Question: 49 top https://gateoverflow.in/39711

✔ Answer is 1.

In LIFO first 20 are page faults followed by next 9 hits then next 11 page faults. (After a_{10}, a_{11} replaces a_{10}, a_{12} replaces a_{11} and so on)

In optimal first 20 are page faults followed by next 9 hits then next 10 page faults followed by last page hit.

👍 70 votes -- Krishna murthy (271 points)

5.12.24 Page Replacement: GATE CSE 2016 Set 2 | Question: 20 top https://gateoverflow.in/39559

✔ Option D. FIFO suffers from Belady's anomaly.

Check this out:

- https://gateoverflow.in/1301/gate2009_9
- https://gateoverflow.in/1254/gate2007_56
- https://gateoverflow.in/2595/gate1995_1-8

References

19 votes -- Shashank Chavan (2.4k points)

5.12.25 Page Replacement: GATE CSE 2017 Set 1 | Question: 40 top https://gateoverflow.in/118323

✔ A page replacement algorithm suffers from Belady's anamoly when it is not a stack algorithm.

A stack algorithm is one that satisfies the inclusion property. The inclusion property states that, at a given time, the contents(pages) of a memory of size k page-frames is a subset of the contents of memory of size $k+1$ page-frames, for the same sequence of accesses. The advantage is that running the same algorithm with more pages(i.e. larger memory) will never increase the number of page faults.

Is LRU a stack algorithm?

Yes, LRU is a stack algorithm. Therefore, it doesn't suffer from Belady's anamoly.

Ref : Ref1 and Ref2

Is Random page replacement algorithm a stack algorithm?

No, as it may choose a page to replace in FIFO manner or in a manner which does not satisfy inclusion property. This means it could suffer from Belady's anamoly.

$\therefore$ (**B**) should be answer.

References

65 votes -- Kantikumar (3.4k points)

5.12.26 Page Replacement: GATE CSE 2021 Set 1 | Question: 11 top https://gateoverflow.in/357441

✔ 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

Correct ans A and C

2 votes -- Meetdoshi90 (281 points)

5.12.27 Page Replacement: GATE CSE 2021 Set 2 | Question: 48 top https://gateoverflow.in/357489

✔ Given :

- Virtual address $(\text{VA}) = 39$ bits
- Page size $= 4\text{KB}$
- Physical address $(\text{PA}) = 2\text{GB}$
- Page table entry size $(\text{PTE}) = 8\text{B}$
- Three level pages tables with address division $(9, 9, 9, 12)$

Three level pages tables with address division $(9, 9, 9, 12)$ means:

- 9 most significant bits for indexing into the level-1(outer level),
- 9 bits for the level-2 index,
- 9 bits for the level-3 index, and
- 12 bits for the offset within a page.

The entries of the level-1 page table are pointers to a level-2 page table, the entries of the level-2 page table are pointers to a level-3 page table, and the entries of the level-3 page table are PTEs that contain actual frame number where our desired word resides.

9 bits for a level means 2^9 entries in one-page table of that level.

For our process P :

P is using 2 GB of its VM. The rest of its VM is unused.

2 GB VM will have $2 \text{ GB}/4 \text{ KB} = 2^{19}$ Pages.

But level 3 page table has only 2^9 entries. So, one-page table of level 3 can point to 2^9 pages of VM only, So, we need 2^{10} level-3 page tables of process P.

So, at level-3, we have 2^{10} page tables, So, we need 2^{10} entries in Level-2 But level 2 page table has only 2^9 entries, so, one-page table of level 2 can only point to 2^9 page tables of level-3, So, we need 2 level-2 page tables.

So, we need 1 Level-1 page table to point to level-2 page tables.

So, for process P, we need only 1 Level-1 page table, 2 level-2 page tables, and 2^{10} level-3 page tables.

Note that All the page tables, at every level, have same size which is $2^9 \times 8 \text{ B} = 2^{12} \text{ B} = 4 \text{ KB}$

(Because every page table at every level has 2^9 entries and Page table entry size at every level is 8 B)

So, in total, we need $1 + 2 + 2^{10}$ page tables (1 Level-1, 2 Level-2, 2^{10} level-3), and each page table size is 4 KB

So, total page tables size $= 1027 \times 4 \text{ KB} = 4108 \text{ KB}$

So, the answer is 4108.

NOTE :

In this question, in place of Multilevel paging, If we had used Single Level Page table (also known as Flat level page table OR linear page table), then size of page table would be 1 GB.

Single Level Page Table :

Single-Level Page Tables are single linear array of page-table entries (PTEs). Each PTE contains information about the page, such as its physical page number ("frame" number) as well as status bits, such as whether or not the page is valid, and other bits. the ith entry in the array gives the frame number in which the ith page is stored.

- Virtual address(VA) = 39 bits
- Page size = 4 KB

So, number of pages in Virtual address space (VAS) of each process $= 2^{39} B/4KB = 2^{27}$

So, we need 2^{27} entries in the page table. Each PTE size = 8 B

So, size of page table for the process $= 2^{29} \times 8 \text{ B} = 1 \text{ GB}$

NOTE that Single level paging CANNOT take advantage of the unused space by the process. The single level page table needs one entry per page. Furthermore, since the process has a very sparse virtual address space, so, the vast majority of these PTEs would simply be marked invalid. BUT space taken by single level page table will be 1GB only. It only depends on the virtual address space, NOT depend on the used memory of process.

A Common Mistake that students make :

In this question, if in place of Multilevel paging, If we had used Single Level Page table, then what would be size pf page table ??

The mistake is that some students will consider 2 GB memory that the process is using, and will get answer $(2 \text{ GB}/4 \text{ KB}) \times 8 \text{ B} = 4 \text{ MB}$ which is wrong.

Remember that the CORE reason why we use multilevel paging in place of single level paging is that we want to reduce size of page table by taking advantage of unused space of process and making most entries in the outer level page table as invalid entries.

- https://people.cs.umass.edu/~emery/classes/cmpsci377/current/notes/lecture_15_vm.pdf
- https://www.youtube.com/watch?v=PKy9Jxc3blw
- https://www.youtube.com/watch?v=pcTAoyzW2rY

References

👍 6 votes

-- Deepak Poonia (23.4k points)

5.12.28 Page Replacement: GATE IT 2007 | Question: 12 top

https://gateoverflow.in/3445

✔ 0100 - page fault, addresses till 199 in memory

0200 - page fault, addresses till 299 in memory

0430 - page fault, addresses till 529 in memory

0499 - no page fault

0510 - no page fault

0530 - page fault, addresses till 629 in memory

0560 - no page fault

0120 - page fault, addresses till 219 in memory

0220 - page fault, addresses till 319 in memory

0240 - no page fault

0260 - no page fault

0320 - page fault, addresses till 419 in memory

0410 - no page fault

So, 7 is the answer- **(C)**

👍 67 votes

-- Arjun Suresh (332k points)

5.12.29 Page Replacement: GATE IT 2007 | Question: 58 top

https://gateoverflow.in/3500

✔ $p(p \times 300 + (1-p) \times 100) + (1-p) \times 1 = 3$

$\implies p(300p + 100 - 100p) + 1 - p = 3$

$\implies 200p^2 + 99p - 2 = 0$

After solving this equation using Sridharacharya formula: $\frac{-b+\sqrt{b^2-4ac}}{2a}$, we get

$$p \approx 0.0194.$$

👍 72 votes

-- Laxmi (793 points)

5.12.30 Page Replacement: GATE IT 2008 | Question: 41 top

https://gateoverflow.in/3351

✔ At first we have to translate the given virtual addresses (which addresses a byte) to page addresses (which again is virtual but addresses a page). This can be done simply by dividing the virtual addresses by page size and taking the floor value (equivalently by removing the page offset bits). Here, page size is 16 bytes which requires 4 offset bits. So,

$0, 4, 8, 20, 24, 36, 44, 12, 68, 72, 80, 84, 28, 32, 88, 92 \implies 0, 0, 0, 1, 1, 2, 2, 0, 4, 4, 5, 5, 1, 2, 5, 5$

We have 4 spaces for a page and there will be a replacement only when a 5^{th} distinct page comes. Lets see what happens for the sequence of memory accesses:

Incoming Virtual Address	Page Address	No. of Page Faults	Pages in Memory in LRU Order
0	0	1	0
4	0	1	0
8	0	1	0
20	1	2	0, 1
24	1	2	0, 1
36	2	3	0, 1, 2
44	2	3	0, 1, 2
12	0	3	1, 2, 0
68	4	4	1, 2, 0, 4
72	4	4	1, 2, 0, 4
80	5	5	2, 0, 4, 5
84	5	5	2, 0, 4, 5
28	1	6	0, 4, 5, 1
32	2	7	4, 5, 1, 2
88	5	7	4, 1, 2, 5
92	5	7	4, 1, 2, 5

So, (**B**) choice.

69 votes -- Arjun Suresh (332k points)

5.13 Precedence Graph (3) top

5.13.1 Precedence Graph: GATE CSE 1989 | Question: 11b top

https://gateoverflow.in/91096

Consider the following precedence graph (Fig.6) of processes where a node denotes a process and a directed edge from node P_i to node P_j implies; that P_i must complete before P_j commences. Implement the graph using FORK and JOIN constructs. The actual computation done by a process may be indicated by a comment line.

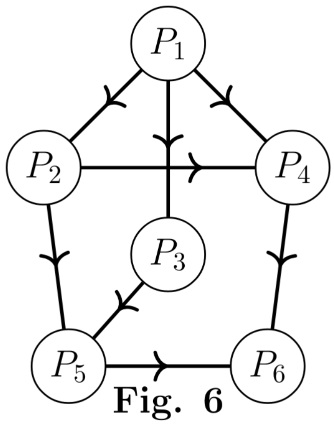


Fig. 6

gate1989 descriptive operating-system precedence-graph process-synchronization

Answer

5.13.2 Precedence Graph: GATE CSE 1991 | Question: 01-xii top

https://gateoverflow.in/508

A given set of processes can be implemented by using only **parbegin/parend** statement, if the precedence graph of these processes is _______

gate1991 operating-system normal precedence-graph fill-in-the-blanks

Answer

5.13.3 Precedence Graph: GATE CSE 1992 | Question: 12-a top

https://gateoverflow.in/591

Draw the precedence graph for the concurrent program given below

```
S1
parbegin
    begin
        S2:S4
    end;
    begin
        S3;
        parbegin
            S5;
            begin
                S6:S8
            end
        parend
    end;
   S7
parend;
S9
```

gate1992 operating-system normal concurrency precedence-graph descriptive

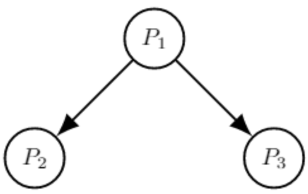
Answer

Answers: Precedence Graph

5.13.1 Precedence Graph: GATE CSE 1989 | Question: 11b top

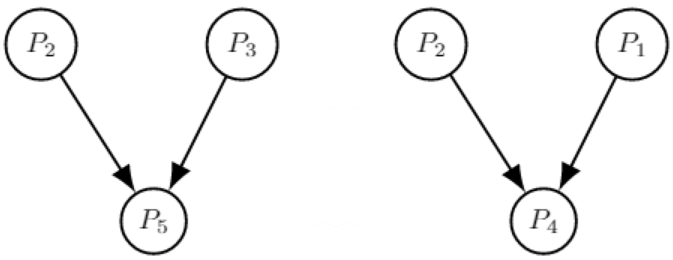
https://gateoverflow.in/91096

Step 1 :



- P_1
- fork L_1
- P_2
- fork L_2
- L_1 : fork L_2
- P_3
- goto L_3

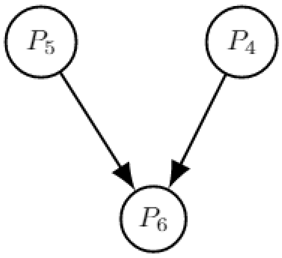
Step 2 :



and

- L_2 : Join C_1
- P_4
- goto L_4
- L_3 : Join C_2
- P_5
- goto L_4

Step 3 :



- L_4 : Join C_3
- P_6

3 votes

-- pankaj borah (41 points)

5.13.2 Precedence Graph: GATE CSE 1991 | Question: 01-xii top https://gateoverflow.in/508

A given set of processes can be implemented by using only **parbegin/parend**statement, if the precedence graph of these processes is **properly nested**

Reference : http://nob.cs.ucdavis.edu/classes/ecs150-2008-04/handouts/sync.pdf

1. It should be closed under par begin and par end.
2. Process execute concurrently.

https://gateoverflow.in/1739/gate1998_24#viewbutton

In this question precedence graph is nested.

1. All the process execute concurrently, closed under par begin and par end.
2. If you see all the serial execution come then signal the resource and and parallel process down the value (resource) similar all the process which are which are dependent to other one, other one release the resource then it will be got that with down and after release the its own resource. In the sense all the process are executing concurrently.

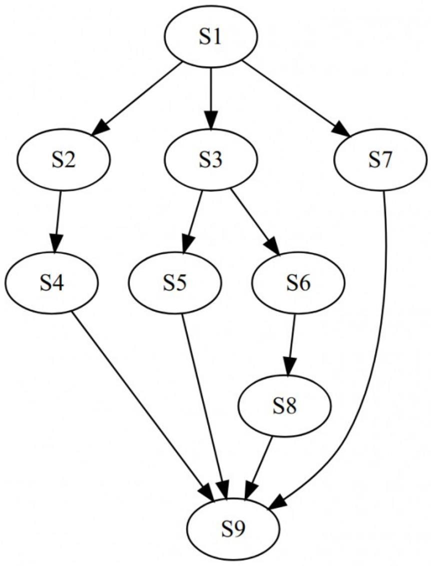
References

15 votes

-- minal (13.1k points)

5.13.3 Precedence Graph: GATE CSE 1992 | Question: 12-a top https://gateoverflow.in/591

parbegin-parend shows parallel execution while begin-end shows serial execution



👍 21 votes

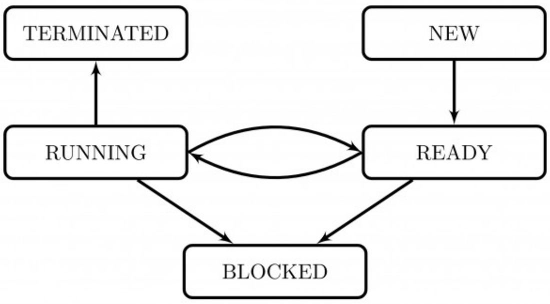
-- Sheshang M. Ajwalia (2.6k points)

5.14 Process (4) top

5.14.1 Process: GATE CSE 1996 | Question: 1.18 top

https://gateoverflow.in/2722

The process state transition diagram in the below figure is representative of



A. a batch operating system

B. an operating system with a preemptive scheduler

C. an operating system with a non-preemptive scheduler

D. a uni-programmed operating system

gate1996 operating-system normal process

Answer

5.14.2 Process: GATE CSE 2001 | Question: 2.20 top

https://gateoverflow.in/738

Which of the following does not interrupt a running process?

A. A device
B. Timer
C. Scheduler process
D. Power failure

gate2001-cse operating-system easy process

Answer

5.14.3 Process: GATE CSE 2002 | Question: 2.21 top https://gateoverflow.in/851

Which combination of the following features will suffice to characterize an OS as a multi-programmed OS?

a. More than one program may be loaded into main memory at the same time for execution
b. If a program waits for certain events such as I/O, another program is immediately scheduled for execution
c. If the execution of a program terminates, another program is immediately scheduled for execution.

A. (a)
B. (a) and (b)
C. (a) and (c)
D. (a), (b) and (c)

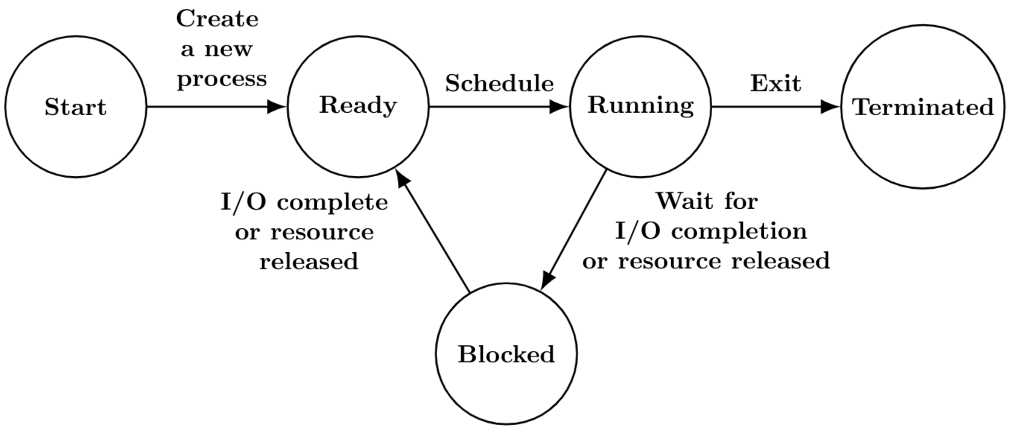
gate2002-cse operating-system normal process

Answer

5.14.4 Process: GATE IT 2006 | Question: 13 top https://gateoverflow.in/3552

The process state transition diagram of an operating system is as given below.

Which of the following must be FALSE about the above operating system?



A. It is a multiprogrammed operating system
B. It uses preemptive scheduling
C. It uses non-preemptive scheduling
D. It is a multi-user operating system

gate2006-it operating-system normal process

Answer

Answers: Process

5.14.1 Process: GATE CSE 1996 | Question: 1.18 top https://gateoverflow.in/2722

✔ Answer is (**B**). The transition from running to ready indicates that the process in the running state can be preempted and brought back to ready state.

35 votes

-- kireeti (1k points)

5.14.2 Process: GATE CSE 2001 | Question: 2.20 top https://gateoverflow.in/738

✔ Answer is (**C**).

Timer and disk both makes interrupt and power failure will also interrupt the system. Only a scheduler process will not interrupt the running process as schduler process gets called only when no other process is running (preemption if any would have happened before scheduler starts execution).

Quote from wikipedia

> In the Linux kernel, the scheduler is called after each timer interrupt (that is, quite a few times per second). It determines what process to run next based on a variety of factors, including priority, time already run, etc. The implementation of preemption in other kernels is likely to be similar.

https://www.quora.com/How-does-the-timer-interrupt-invoke-the-process-scheduler

References

52 votes -- jayendra (6.7k points)

5.14.3 Process: GATE CSE 2002 | Question: 2.21 top

https://gateoverflow.in/851

✔ (A) and (B) suffice multi programming concept. For multi programming more than one program should be in memory and if any program goes for Io another can be scheduled to use CPU as shown below:

So ans is **(B)**.

52 votes -- Pooja Palod (24.1k points)

5.14.4 Process: GATE IT 2006 | Question: 13 top

https://gateoverflow.in/3552

✔ **Answer (B).**

Explanation:

A. It is a multiprogrammed operating system.
Correct, it has ready state. We can have multiple processes in ready state here so this is Multiprogrammed OS.

B. It uses preemptive scheduling
False : There is no arrow transition from running to ready state. So, this is non preemptive.

C. It uses non-preemptive scheduling
True.

D. It is a multi-user operating system.
We can have multiple user processes in ready state. So, this is also correct.

42 votes -- Akash Kanase (36k points)

5.15 Process Scheduling (43) top

5.15.1 Process Scheduling: GATE CSE 1988 | Question: 2xa top

https://gateoverflow.in/93951

State any undesirable characteristic of the following criteria for measuring performance of an operating system:

Turn around time

gate1988 normal descriptive operating-system process-scheduling

Answer

5.15.2 Process Scheduling: GATE CSE 1988 | Question: 2xb top

https://gateoverflow.in/93953

State any undesirable characteristic of the following criteria for measuring performance of an operating system:

Waiting time

gate1988 normal descriptive operating-system process-scheduling

Answer

5.15.3 Process Scheduling: GATE CSE 1990 | Question: 1-vi top

https://gateoverflow.in/83850

The highest-response ratio next scheduling policy favours ___________ jobs, but it also limits the waiting time of _________ jobs.

gate1990 operating-system process-scheduling fill-in-the-blanks

Answer

5.15.4 Process Scheduling: GATE CSE 1993 | Question: 7.10 top

https://gateoverflow.in/2298

Assume that the following jobs are to be executed on a single processor system

Job Id	CPU Burst Time
p	4
q	1
r	8
s	1
t	2

The jobs are assumed to have arrived at time 0^+ and in the order p, q, r, s, t. Calculate the departure time (completion time) for job p if scheduling is round robin with time slice 1

A. 4
B. 10
C. 11
D. 12
E. None of the above

gate1993 operating-system process-scheduling normal

Answer

5.15.5 Process Scheduling: GATE CSE 1995 | Question: 1.15 top

https://gateoverflow.in/2602

Which scheduling policy is most suitable for a time shared operating system?

A. Shortest Job First
B. Round Robin
C. First Come First Serve
D. Elevator

gate1995 operating-system process-scheduling easy

Answer

5.15.6 Process Scheduling: GATE CSE 1995 | Question: 2.6 top

https://gateoverflow.in/2618

The sequence __________ is an optimal non-preemptive scheduling sequence for the following jobs which leaves the CPU idle for ________ unit(s) of time.

Job	Arrival Time	Burst Time
1	0.0	9
2	0.6	5
3	1.0	1

A. $\{3, 2, 1\}, 1$
B. $\{2, 1, 3\}, 0$
C. $\{3, 2, 1\}, 0$
D. $\{1, 2, 3\}, 5$

gate1995 operating-system process-scheduling normal

Answer

5.15.7 Process Scheduling: GATE CSE 1996 | Question: 2.20, ISRO2008-15 top

https://gateoverflow.in/2749

Four jobs to be executed on a single processor system arrive at time 0 in the order A, B, C, D. Their burst CPU time requirements are $4, 1, 8, 1$ time units respectively. The completion time of A under round robin scheduling with time slice of one time unit is

A. 10
B. 4
C. 8
D. 9

gate1996 operating-system process-scheduling normal isro2008

Answer

5.15.8 Process Scheduling: GATE CSE 1998 | Question: 2.17, UGCNET-Dec2012-III: 43 top

https://gateoverflow.in/1690

Consider n processes sharing the CPU in a round-robin fashion. Assuming that each process switch takes s seconds, what must be the quantum size q such that the overhead resulting from process switching is minimized but at the same time each process is guaranteed to get its turn at the CPU at least every t seconds?

A. $q \leq \frac{t-ns}{n-1}$
B. $q \geq \frac{t-ns}{n-1}$
C. $q \leq \frac{t-ns}{n+1}$
D. $q \geq \frac{t-ns}{n+1}$

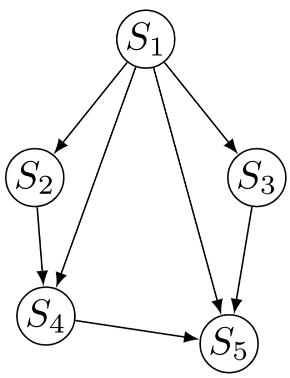
gate1998 operating-system process-scheduling normal ugcnetdec2012iii

Answer

5.15.9 Process Scheduling: GATE CSE 1998 | Question: 24 top

https://gateoverflow.in/1739

a. Four jobs are waiting to be run. Their expected run times are $6, 3, 5$ and x. In what order should they be run to minimize the average response time?
b. Write a concurrent program using par begin-par end to represent the precedence graph shown below.



gate1998 operating-system process-scheduling descriptive

Answer

5.15.10 Process Scheduling: GATE CSE 1998 | Question: 7-b top

https://gateoverflow.in/12963

In a computer system where the 'best-fit' algorithm is used for allocating 'jobs' to 'memory partitions', the following situation was encountered:

Partitions size in KB	$4K$ $8K$ $20K$ $2K$
Job sizes in KB	$2K$ $14K$ $3K$ $6K$ $6K$ $10K$ $20K$ $2K$
Time for execution	4 10 2 1 4 1 8 6

When will the $20K$ job complete?

gate1998 operating-system process-scheduling normal

Answer

5.15.11 Process Scheduling: GATE CSE 2002 | Question: 1.22 top

https://gateoverflow.in/827

Which of the following scheduling algorithms is non-preemptive?

A. Round Robin
B. First-In First-Out
C. Multilevel Queue Scheduling
D. Multilevel Queue Scheduling with Feedback

gate2002-cse operating-system process-scheduling easy

Answer

5.15.12 Process Scheduling: GATE CSE 2003 | Question: 77 top

https://gateoverflow.in/963

A uni-processor computer system only has two processes, both of which alternate 10 ms CPU bursts with 90 ms I/O bursts. Both the processes were created at nearly the same time. The I/O of both processes can proceed in parallel. Which of the following scheduling strategies will result in the *least* CPU utilization (over a long period of time) for this system?

A. First come first served scheduling
B. Shortest remaining time first scheduling
C. Static priority scheduling with different priorities for the two processes
D. Round robin scheduling with a time quantum of 5 ms

gate2003-cse operating-system process-scheduling normal

Answer

5.15.13 Process Scheduling: GATE CSE 2004 | Question: 46 top

https://gateoverflow.in/1043

Consider the following set of processes, with the arrival times and the CPU-burst times gives in milliseconds.

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	3
P4	4	1

What is the average turnaround time for these processes with the preemptive shortest remaining processing time first (SRPT) algorithm?

A. 5.50
B. 5.75
C. 6.00
D. 6.25

gate2004-cse operating-system process-scheduling normal

Answer

5.15.14 Process Scheduling: GATE CSE 2006 | Question: 06, ISRO2009-14 top

https://gateoverflow.in/885

Consider three CPU-intensive processes, which require 10, 20 and 30 time units and arrive at times 0, 2 and 6, respectively. How many context switches are needed if the operating system implements a shortest remaining time first scheduling algorithm? Do not count the context switches at time zero and at the end.

A. 1
B. 2
C. 3
D. 4

gate2006-cse operating-system process-scheduling normal isro2009

Answer

5.15.15 Process Scheduling: GATE CSE 2006 | Question: 64 top

https://gateoverflow.in/1842

Consider three processes (process id 0, 1, 2 respectively) with compute time bursts 2, 4 and 8 time units. All processes arrive at time zero. Consider the longest remaining time first (LRTF) scheduling algorithm. In LRTF ties are broken by giving priority to the process with the lowest process id. The average turn around time is:

A. 13 units
B. 14 units
C. 15 units
D. 16 units

gate2006-cse operating-system process-scheduling normal

Answer

5.15.16 Process Scheduling: GATE CSE 2006 | Question: 65 top

https://gateoverflow.in/1843

Consider three processes, all arriving at time zero, with total execution time of 10, 20 and 30 units, respectively. Each process spends the first 20% of execution time doing I/O, the next 70% of time doing computation, and the last 10% of time doing I/O again. The operating system uses a shortest remaining compute time first scheduling algorithm and schedules a new process either when the running process gets blocked on I/O or when the running process finishes its compute burst. Assume that all I/O operations can be overlapped as much as possible. For what percentage of time does the CPU remain idle?

A. 0%
B. 10.6%
C. 30.0%
D. 89.4%

gate2006-cse operating-system process-scheduling normal

Answer

5.15.17 Process Scheduling: GATE CSE 2007 | Question: 16 top

https://gateoverflow.in/1214

Group 1 contains some CPU scheduling algorithms and Group 2 contains some applications. Match entries in Group 1 to entries in Group 2.

Group I		Group II	
(P)	Gang Scheduling	(1)	Guaranteed Scheduling
(Q)	Rate Monotonic Scheduling	(2)	Real-time Scheduling
(R)	Fair Share Scheduling	(3)	Thread Scheduling

A. $P-3; Q-2; R-1$
B. $P-1; Q-2; R-3$
C. $P-2; Q-3; R-1$
D. $P-1; Q-3; R-2$

gate2007-cse operating-system process-scheduling normal

Answer

5.15.18 Process Scheduling: GATE CSE 2007 | Question: 55 top https://gateoverflow.in/1253

An operating system used Shortest Remaining System Time first (SRT) process scheduling algorithm. Consider the arrival times and execution times for the following processes:

Process	Execution Time	Arrival Time
P1	20	0
P2	25	15
P3	10	30
P4	15	45

What is the total waiting time for process $P2$?

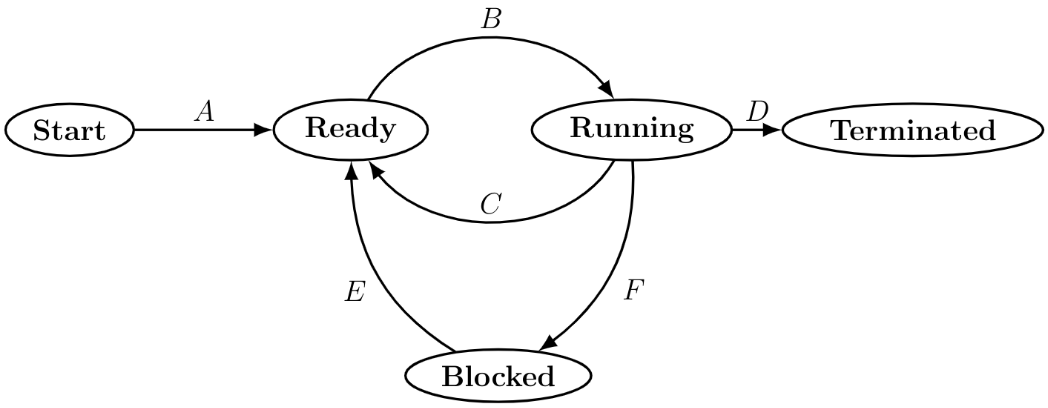
A. 5
B. 15
C. 40
D. 55

gate2007-cse operating-system process-scheduling normal

Answer

5.15.19 Process Scheduling: GATE CSE 2009 | Question: 32 top https://gateoverflow.in/1318

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:

I. If a process makes a transition D, it would result in another process making transition A immediately.
II. A process P_2 in blocked state can make transition E while another process P_1 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. II and IV

gate2009-cse operating-system process-scheduling normal

Answer

5.15.20 Process Scheduling: GATE CSE 2010 | Question: 25 top https://gateoverflow.in/2204

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

gate2010-cse operating-system process-scheduling easy

Answer

5.15.21 Process Scheduling: GATE CSE 2011 | Question: 35 top https://gateoverflow.in/2137

Consider the following table of arrival time and burst time for three processes $P0, P1$ and $P2$.

Process	Arrival Time	Burst Time
P0	0 ms	9
P1	1 ms	4
P2	2 ms	9

The pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of processes. What is the average waiting time for the three processes?

A. 5.0 ms
B. 4.33 ms
C. 6.33 ms
D. 7.33 ms

gate2011-cse operating-system process-scheduling normal

Answer

5.15.22 Process Scheduling: GATE CSE 2012 | Question: 31 top https://gateoverflow.in/1749

Consider the 3 processes, $P1$, $P2$ and $P3$ shown in the table.

Process	Arrival Time	Time Units Required
P1	0	5
P2	1	7
P3	3	4

The completion order of the 3 processes under the policies FCFS and RR2 (round robin scheduling with CPU quantum of 2 time units) are

A. **FCFS**: $P1, P2, P3$ **RR2**: $P1, P2, P3$
B. **FCFS**: $P1, P3, P2$ **RR2**: $P1, P3, P2$
C. **FCFS**: $P1, P2, P3$ **RR2**: $P1, P3, P2$
D. **FCFS**: $P1, P3, P2$ **RR2**: $P1, P2, P3$

gate2012-cse operating-system process-scheduling normal

Answer

5.15.23 Process Scheduling: GATE CSE 2013 | Question: 10 top https://gateoverflow.in/1419

A scheduling algorithm assigns priority proportional to the waiting time of a process. Every process starts with zero (the lowest priority). The scheduler re-evaluates the process priorities every T time units and decides the next process to schedule. Which one of the following is **TRUE** if the processes have no I/O operations and all arrive at time zero?

A. This algorithm is equivalent to the first-come-first-serve algorithm.
B. This algorithm is equivalent to the round-robin algorithm.
C. This algorithm is equivalent to the shortest-job-first algorithm.
D. This algorithm is equivalent to the shortest-remaining-time-first algorithm.

gate2013-cse operating-system process-scheduling normal

Answer

5.15.24 Process Scheduling: GATE CSE 2014 Set 1 | Question: 32 top

https://gateoverflow.in/1803

Consider the following set of processes that need to be scheduled on a single CPU. All the times are given in milliseconds.

Process Name	Arrival Time	Execution Time
A	0	6
B	3	2
C	5	4
D	7	6
E	10	3

Using the *shortest remaining time first* scheduling algorithm, the average process turnaround time (in msec) is ____________________.

gate2014-cse-set1 operating-system process-scheduling numerical-answers normal

Answer

5.15.25 Process Scheduling: GATE CSE 2014 Set 2 | Question: 32 top

https://gateoverflow.in/1991

Three processes A, B and C each execute a loop of 100 iterations. In each iteration of the loop, a process performs a single computation that requires t_c CPU milliseconds and then initiates a single I/O operation that lasts for t_{io} milliseconds. It is assumed that the computer where the processes execute has sufficient number of I/O devices and the OS of the computer assigns different I/O devices to each process. Also, the scheduling overhead of the OS is negligible. The processes have the following characteristics:

Process id	t_c	t_{io}
A	100 ms	500 ms
B	350 ms	500 ms
C	200 ms	500 ms

The processes A, B, and C are started at times 0, 5 and 10 milliseconds respectively, in a pure time sharing system (round robin scheduling) that uses a time slice of 50 milliseconds. The time in milliseconds at which process C would ***complete*** its first I/O operation is ___________.

gate2014-cse-set2 operating-system process-scheduling numerical-answers normal

Answer

5.15.26 Process Scheduling: GATE CSE 2014 Set 3 | Question: 32 top

https://gateoverflow.in/2066

An operating system uses *shortest remaining time first* scheduling algorithm for pre-emptive scheduling of processes. Consider the following set of processes with their arrival times and CPU burst times (in milliseconds):

Process	Arrival Time	Burst Time
P1	0	12
P2	2	4
P3	3	6
P4	8	5

The average waiting time (in milliseconds) of the processes is ______.

gate2014-cse-set3 operating-system process-scheduling numerical-answers normal

Answer

5.15.27 Process Scheduling: GATE CSE 2015 Set 1 | Question: 46 top

https://gateoverflow.in/8330

Consider a uniprocessor system executing three tasks T_1, T_2 and T_3 each of which is composed of an infinite sequence of jobs (or instances) which arrive periodically at intervals of 3, 7 and 20 milliseconds, respectively. The priority of each task is the inverse of its period, and the available tasks are scheduled in order of priority, which is the highest priority task scheduled first. Each instance of T_1, T_2 and T_3 requires an execution time of 1, 2 and 4 milliseconds, respectively. Given that all tasks initially arrive at the beginning of the 1^{st} millisecond and task preemptions are allowed, the first instance of T_3 completes its execution at the end of______________________milliseconds.

gate2015-cse-set1 operating-system process-scheduling normal numerical-answers

Answer

5.15.28 Process Scheduling: GATE CSE 2015 Set 3 | Question: 1 top https://gateoverflow.in/8390

The maximum number of processes that can be in $Ready$ state for a computer system with n CPUs is :

A. n
B. n^2
C. 2^n
D. Independent of n

gate2015-cse-set3 operating-system process-scheduling easy

Answer

5.15.29 Process Scheduling: GATE CSE 2015 Set 3 | Question: 34 top https://gateoverflow.in/8492

For the processes listed in the following table, which of the following scheduling schemes will give the lowest average turnaround time?

Process	Arrival Time	Process Time
A	0	3
B	1	6
C	4	4
D	6	2

A. First Come First Serve
B. Non-preemptive Shortest job first
C. Shortest Remaining Time
D. Round Robin with Quantum value two

gate2015-cse-set3 operating-system process-scheduling normal

Answer

5.15.30 Process Scheduling: GATE CSE 2016 Set 1 | Question: 20 top https://gateoverflow.in/39655

Consider an arbitrary set of CPU-bound processes with unequal CPU burst lengths submitted at the same time to a computer system. Which one of the following process scheduling algorithms would minimize the average waiting time in the ready queue?

A. Shortest remaining time first
B. Round-robin with the time quantum less than the shortest CPU burst
C. Uniform random
D. Highest priority first with priority proportional to CPU burst length

gate2016-cse-set1 operating-system process-scheduling normal

Answer

5.15.31 Process Scheduling: GATE CSE 2016 Set 2 | Question: 47 top https://gateoverflow.in/39625

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.

Process	Arrival Time	Burst Time
P_1	0	10
P_2	3	6
P_3	7	1
P_4	8	3

The average turn around time of these processes is ___________ milliseconds.

gate2016-cse-set2 operating-system process-scheduling normal numerical-answers

Answer

5.15.32 Process Scheduling: GATE CSE 2017 Set 1 | Question: 24 top

https://gateoverflow.in/118304

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
P_1	0	7
P_2	3	3
P_3	5	5
P_4	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.

gate2017-cse-set1 operating-system process-scheduling numerical-answers

Answer

5.15.33 Process Scheduling: GATE CSE 2017 Set 2 | Question: 51 top

https://gateoverflow.in/118558

Consider the set of process with arrival time (in milliseonds), CPU burst time (in millisecods) and priority (0 is the highest priority) shown below. None of the process have I/O burst time

Process	Arrival Time	Burst Time	Priority
P_1	0	11	2
P_2	5	28	0
P_3	12	2	3
P_4	2	10	1
P_5	9	16	4

The average waiting time (in milli seconds) of all the process using premtive priority scheduling algorithm is ______

gate2017-cse-set2 operating-system process-scheduling numerical-answers

Answer

5.15.34 Process Scheduling: GATE CSE 2019 | Question: 41 top

https://gateoverflow.in/302807

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 preemptive Shortest Remaining Time First scheduling algorithm. If the average waiting time of the processes is 1 millisecond, then the value of Z is _____

gate2019-cse numerical-answers operating-system process-scheduling

Answer

5.15.35 Process Scheduling: GATE CSE 2020 | Question: 12 top

https://gateoverflow.in/333219

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, and III only
B. II and III only
C. I, II, and IV only
D. I, II, III and IV only

gate2020-cse operating-system process-scheduling

Answer

5.15.36 Process Scheduling: GATE CSE 2020 | Question: 50 top

https://gateoverflow.in/333181

Consider the following set of processes, assumed to have arrived at time 0. 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 to 2 decimal places is_______

gate2020-cse numerical-answers operating-system process-scheduling

Answer

5.15.37 Process Scheduling: GATE CSE 2021 Set 1 | Question: 25 top

https://gateoverflow.in/357426

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 three processes in a non-preemptive scheduler (rounded to nearest integer) is _____________ milliseconds.

gate2021-cse-set1 operating-system process-scheduling numerical-answers

Answer

5.15.38 Process Scheduling: GATE CSE 2021 Set 2 | Question: 14 top

https://gateoverflow.in/357526

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

A. Turnaround time includes waiting time
B. The goal is to only maximize CPU utilization and minimize throughput
C. Round-robin policy can be used even when the CPU time required by each of the processes is not known apriori
D. Implementing preemptive scheduling needs hardware support

gate2021-cse-set2 multiple-selects operating-system process-scheduling

Answer

5.15.39 Process Scheduling: GATE IT 2005 | Question: 60 top

https://gateoverflow.in/3821

We wish to schedule three processes $P1$, $P2$ and $P3$ on a uniprocessor system. The priorities, CPU time requirements and arrival times of the processes are as shown below.

Process	Priority	CPU time required	Arrival time (hh:mm:ss)
P1	10 (highest)	20 sec	00 : 00 : 05
P2	9	10 sec	00 : 00 : 03
P3	8 (lowest)	15 sec	00 : 00 : 00

We have a choice of preemptive or non-preemptive scheduling. In preemptive scheduling, a late-arriving higher priority process can

preempt a currently running process with lower priority. In non-preemptive scheduling, a late-arriving higher priority process must wait for the currently executing process to complete before it can be scheduled on the processor.
What are the turnaround times (time from arrival till completion) of $P2$ using preemptive and non-preemptive scheduling respectively?

A. 30 sec, 30 sec
B. 30 sec, 10 sec
C. 42 sec, 42 sec
D. 30 sec, 42 sec

gate2005-it operating-system process-scheduling normal

Answer

5.15.40 Process Scheduling: GATE IT 2006 | Question: 12 top

https://gateoverflow.in/3551

In the working-set strategy, which of the following is done by the operating system to prevent thrashing?

I. It initiates another process if there are enough extra frames.
II. It selects a process to suspend if the sum of the sizes of the working-sets exceeds the total number of available frames.

A. I only
B. II only
C. Neither I nor II
D. Both I and II

gate2006-it operating-system process-scheduling normal

Answer

5.15.41 Process Scheduling: GATE IT 2006 | Question: 54 top

https://gateoverflow.in/3597

The arrival time, priority, and duration of the CPU and I/O bursts for each of three processes P_1, P_2 and P_3 are given in the table below. Each process has a CPU burst followed by an I/O burst followed by another CPU burst. Assume that each process has its own I/O resource.

Process	Arrival Time	Priority	Burst duration (CPU)	Burst duration (I/O)	Burst duration) (CPU)
P_1	0	2	1	5	3
P_2	2	3 (lowest)	3	3	1
P_3	3	1 (highest)	2	3	1

The multi-programmed operating system uses preemptive priority scheduling. What are the finish times of the processes P_1, P_2 and P_3?

A. 11, 15, 9
B. 10, 15, 9
C. 11, 16, 10
D. 12, 17, 11

gate2006-it operating-system process-scheduling normal

Answer

5.15.42 Process Scheduling: GATE IT 2007 | Question: 26 top

https://gateoverflow.in/3459

Consider n jobs $J_1, J_2 \dots J_n$ such that job J_i has execution time t_i and a non-negative integer weight w_i. The weighted mean completion time of the jobs is defined to be $\frac{\sum_{i=1}^{n} w_i T_i}{\sum_{i=1}^{n} w_i}$, where T_i is the completion time of job J_i. Assuming that there is only one processor available, in what order must the jobs be executed in order to minimize the weighted mean completion time of the jobs?

A. Non-decreasing order of t_i
B. Non-increasing order of w_i
C. Non-increasing order of $w_i t_i$
D. Non-increasing order of w_i / t_i

gate2007-it operating-system process-scheduling normal

Answer

5.15.43 Process Scheduling: GATE IT 2008 | Question: 55 top

https://gateoverflow.in/3365

If the time-slice used in the round-robin scheduling policy is more than the maximum time required to execute any process, then the policy will

A. degenerate to shortest job first
B. degenerate to priority scheduling
C. degenerate to first come first serve
D. none of the above

gate2008-it operating-system process-scheduling easy

Answer

Answers: Process Scheduling

5.15.1 Process Scheduling: GATE CSE 1988 | Question: 2xa top

https://gateoverflow.in/93951

By the way the turnaround time should not be a metre to evalaute the performance of an OS .

But here they ask so

Undesirable is that : (i) long burst process are running first and smaller run after long.

4 votes

-- hem chandra joshi (2.9k points)

5.15.2 Process Scheduling: GATE CSE 1988 | Question: 2xb top

https://gateoverflow.in/93953

✔ "Waiting time" is one of the metric for deciding the schedule of processes. If the OS tries to minimize the average waiting time of the processes it'll follow the Shortest Remaining Time First algorithm which though reduces the average waiting time of processes can still cause a long burst time process to starve.

0 votes

-- Arjun Suresh (332k points)

5.15.3 Process Scheduling: GATE CSE 1990 | Question: 1-vi top

https://gateoverflow.in/83850

✔ **Highest response ratio next** (**HRRN**) scheduling is a non-preemptive discipline, similar to shortest job next (SJN), in which the priority of each job is dependent on its estimated run time, and also the amount of time it has spent waiting.

Jobs gain higher priority the longer they wait, which prevents indefinite waiting or in other words what we say starvation. In fact, the jobs that have spent a long time waiting compete against those estimated to have short run times.

$$\text{Priority} = \frac{\text{waiting time} + \text{estimated runtime}}{\text{estimated runtime}}$$

So, the conclusion is it gives priority to those processes which have less burst time (or execution time) but also takes care of the waiting time of longer processes,thus preventing starvation.

So, the answer is ***"shorter , longer"***

48 votes

-- HABIB MOHAMMAD KHAN (67.5k points)

5.15.4 Process Scheduling: GATE CSE 1993 | Question: 7.10 top

https://gateoverflow.in/2298

✔ Answer: (C)

Execution order: $p\ q\ r\ s\ t\ p\ r\ t\ p\ r\ \mathbf{p}\ r\ r\ r\ r\ r$

24 votes

-- Rajarshi Sarkar (27.9k points)

5.15.5 Process Scheduling: GATE CSE 1995 | Question: 1.15 top https://gateoverflow.in/2602

✔ Answer is Round Robin (RR), option (B).

Now question is Why RR is most suitable for time shared OS?

First of all we are discussing about **Time shared OS**, so obviously **We need to consider pre-emption .**

So, FCFS and Elevator these 2 options removed first , remain SJF and RR from two remaining options.

Now in case of pre-emptive SJF which is also known as shortest remaining time first or SRTF (where we can predict the next burst time using exponential averaging), SRTF would *NOT be optimal* than RR.

- There is **no starvation** in case of RR, since every process shares a time slice.
- But In case of SRTF, **there can be a starvation** , in **worse case** you may have the highest priority process, with a huge burst time have to wait.That means long process may have to wait indefinite time in case of SRTF.

That's why RR can be chosen over SRTF in case of time shared OS.

👍 44 votes

-- Bikram (58.4k points)

5.15.6 Process Scheduling: GATE CSE 1995 | Question: 2.6 top https://gateoverflow.in/2618

✔ Answer is (**A**).
Here, in option B and C they have given CPU idle time is 0 which is not possible as per schedule (B) and (C).
So, (B) and (C) are eliminated.

Now, lets see (A) and (D):

For (A),

Idle	Job_3	Job_2	Job_1

0 1 2 7 16

So, idle time is between 0 and 1 which is 1 in case of option (A).

For option (D),

Job_1	Job_2	Job_3

0 9 14 15

We can see that there is no idle time at all, but in option given idle time is 5, which is not matching with our chart so option (D) is eliminated.

Therefore, the correct sequence is option (A).

👍 43 votes

-- jayendra (6.7k points)

5.15.7 Process Scheduling: GATE CSE 1996 | Question: 2.20, ISRO2008-15 top https://gateoverflow.in/2749

✔ The completion time of A will be 9 Unit.

Hence, option (D) is correct.

Here, is the sequence (Consider each block takes one time unit)

A	B	C	D	A	C	A	C	A

Completion time of A will be 9.

👍 29 votes

-- Muktinath Vishwakarma (23.9k points)

5.15.8 Process Scheduling: GATE CSE 1998 | Question: 2.17, UGCNET-Dec2012-III: 43 top https://gateoverflow.in/1690

✔ Answer: (A)

Each process runs for q period and if there are n process: p_1, p_2,p_3,, ..., p_n,.
Then p_1's turn comes again when it has completed time quanta for remaining process p_2 to p_n, i.e, it would take at most $(n-1)q$ time.
So,, each process in round robin gets its turn after $(n-1)q$ time when we don't consider overheads but if we consider overheads then it would be $ns+(n-1)q$
So, we have $ns+(n-1)q \leq t$

77 votes -- Rajarshi Sarkar (27.9k points)

5.15.9 Process Scheduling: GATE CSE 1998 | Question: 24 top https://gateoverflow.in/1739

✔

a. Here, all we need to do for minimizing response time is to run jobs in increasing order of burst time.
b. Schedule shorter jobs first, which will decrease the waiting time of longer jobs, and consequently average waiting time and average response time decreases.
c.

   $6, 3, 5$ and x.

   If $X < 3 < 5 < 6$, then order should be $x, 3, 5, 6$

   If $3 < 5 < 6 < x$, then order is $3, 5, 6, x$.

   If $3 < x < 5 < 6$, then order is $3, x, 5, 6$. If $5 < x < 6$, then order is $3, 5, x, 6$.

Idea is that if you have $S_1 \rightarrow S_2$ then you create new semaphore a, assume that initial value of all semaphores is 0. Then S_2 thread will invoke $P(a)$ & will get blocked. When S_1 get executed, after that it'll do $V(a)$ which will enable S_2 to run. Do like this for all edges in graph.
Let me write program for it.

Begin
Semaphores a, b, c, d, e, f, g
ParBegin $S_1 V(a) V(b) V(c) V(d)$ Parend
ParBegin $P(a) S_2 V(e)$ Parend
ParBegin $P(b) S_3 V(f)$ Parend
ParBegin $P(c) P(e) S_4 V(g)$ Parend
ParBegin $P(d) P(f) P(g) S_5$ Parend
End

IF you reverse engineer this program you can get how this diagram came.

Parbegin Parend – Parallel execution

$P-$ Down, $V-$ Up

30 votes -- Akash Kanase (36k points)

5.15.10 Process Scheduling: GATE CSE 1998 | Question: 7-b top https://gateoverflow.in/12963

✔ The partitions are $4k, 8k, 20k, 2k$, now due to the best-fit algorithm,

1. Size of $2k$ job will fit in $2k$ partition and execute for 4 unit
2. Size of $14k$ job will be fit in $20k$ partition and execute for 10 unit
3. Size of $3k$ job will be fit in $4k$ partition and execute for 2 unit
4. Size of $6k$ job will be fit in $8k$ partition now execute for 1 unit. All partitions are full.

And next job size of $10k$ (5) wait for the partition of $20k$ and after completion of no. 2 job, job no. 5 will be executed for 1 unit $(10$ to $11)$. Now, $20k$ is also waiting for a partition of $20k$ because it is the best fit for it. So after completion of job 5, it will be fit. So, it will execute for 8 unit which is 11 to 19. So, at 19 unit $20k$ job will be completed.

The answer should be 19 units.

👍 32 votes -- minal (13.1k points)

5.15.11 Process Scheduling: GATE CSE 2002 | Question: 1.22 top

https://gateoverflow.in/827

✔

A. Here we preempt when Time quantum is expired.

B. We never preempt, so answer is (**B**) FIFO

C. Here we preempt when process of higher priority arrives.

D. Here we preempt when process of higher priority arrives or when time slice of higher level finishes & we need to move process to lower priority.

👍 38 votes -- Akash Kanase (36k points)

5.15.12 Process Scheduling: GATE CSE 2003 | Question: 77 top

https://gateoverflow.in/963

✔ CPU utilization $=$ CPU burst time/Total time.

FCFS:
from $0-10$: process 1
from $10-20$: process 2
from $100-110$: process 1
from $110-120$: process 2
....
So, in every 100 ms, CPU is utilized for 20 ms, CPU utilization $= 20\%$

SRTF:
Same as FCFS as CPU burst time is same for all the processes

Static priority scheduling:
Suppose process 1 is having higher priority. Now, the scheduling will be same as FCFS. If process 2 is having higher priority, then the scheduling will be as FCFS with process 1 and process 2 interchanged. So, CPU utilization remains at 20%

Round Robin:
Time quantum given as 5 ms.
from $0-5$: process 1
from $5-10$: process 2
from $10-15$: process 1
from $15-20$: process 2
from $105-110$: process 1
from $110-115$: process 2
...
So, in 105 ms, 20 ms of CPU burst is there. So, utilization $= 20/105 = 19.05\%$

19.05 is less than 20, so answer is (**D**).

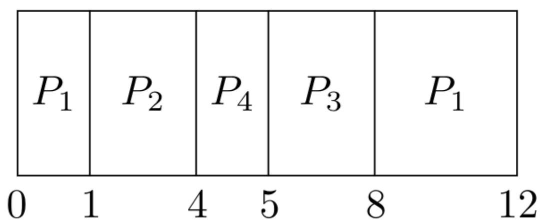
(Round robin with time quantum 10ms would have made the CPU utilization same for all the schedules)

👍 132 votes -- Arjun Suresh (332k points)

5.15.13 Process Scheduling: GATE CSE 2004 | Question: 46 top

https://gateoverflow.in/1043

✔



Process	Waiting Time = (Turnaround Time - Burst time)	Turnaround Time = (Completion Time - Arrival Time)
P1	7	12
P3	0	3
P2	3	6
P4	0	1

Average turnaround time $= 12 + 3 + 6 + 1/4 = 22/4 = 5.5$

Correct Answer: A

25 votes

-- Pooja Palod (24.1k points)

5.15.14 Process Scheduling: GATE CSE 2006 | Question: 06, ISRO2009-14 top

https://gateoverflow.in/885

✔ Processes execute as per the following Gantt chart

P_1	P_2	P_3
0 – 10	10 – 30	30 – 60

So, here only 2 switching possible (when we did not consider the starting and ending switching)

now here might be confusion that at $t = 2$ $p1$ is preempted and check that available process have shortest job time or not, but he did not get anyone so it should not be consider as context switching.(same happened at $t = 6$)

Reference: http://stackoverflow.com/questions/8997616/does-a-context-switch-occur-in-a-system-whose-ready-queue-has-only-one-process-a(thanks to anurag_s)

Answer is **(B)**

References

53 votes

-- minal (13.1k points)

5.15.15 Process Scheduling: GATE CSE 2006 | Question: 64 top

https://gateoverflow.in/1842

✔ A.

Gantt Chart is as follows.

P_2	P_2	P_2	P_2	P_1	P_2	P_1	P_2	P_0	P_1	P_2	P_0	P_1	P_2

Scheduling Table

P.ID	A.T	B.T	C.T	T.A.T.	W.T.
P0	0	2	12	12	10
P1	0	4	13	13	9
P2	0	8	14	14	6
TOTAL				39	25

A.T.= Arrival Time

B.T.= Burst Time

C.T.= Completion Time.

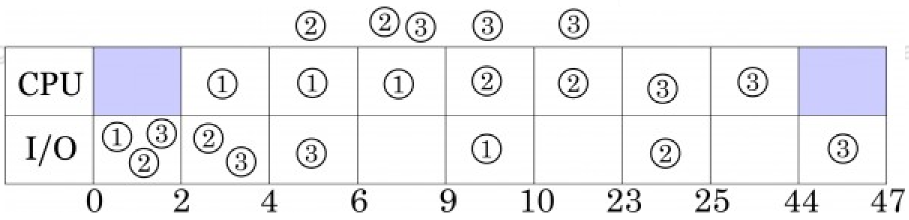
T.A.T.= Turn Around Time

W.T.= Waiting Time.

Average TAT $= 39/3 = 13$ units.

37 votes -- Gate Keeda (15.9k points)

5.15.16 Process Scheduling: GATE CSE 2006 | Question: 65 top https://gateoverflow.in/1843



$$\text{CPU Idle time} = \frac{2+3}{47} \times 100 = 10.6383\%$$

Answer is **option (B).**

74 votes -- Amar Vashishth (25.2k points)

5.15.17 Process Scheduling: GATE CSE 2007 | Question: 16 top https://gateoverflow.in/1214

(**A**) is the answer.

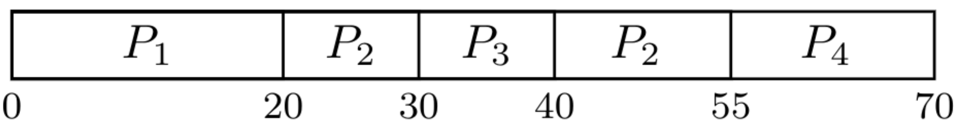
- http://en.wikipedia.org/wiki/Rate-monotonic_scheduling
- http://en.wikipedia.org/wiki/Gang_scheduling
- http://en.wikipedia.org/wiki/Fair-share_scheduling

References

37 votes -- Arjun Suresh (332k points)

5.15.18 Process Scheduling: GATE CSE 2007 | Question: 55 top https://gateoverflow.in/1253

The answer is (**B**).



Gantt Chart

Waiting time for process $P2$ = Completion time – Arrival time – burst time $= 55 – 15 – 25 = 15$

24 votes -- Gate Keeda (15.9k points)

5.15.19 Process Scheduling: GATE CSE 2009 | Question: 32 top https://gateoverflow.in/1318

1. If a process makes a transition D, it would result in another process making transition A immediately. - This is false. It is not said anywhere that one process terminates, another process immediately come into Ready state. It depends on availability of process to run & Long term Scheduler.
2. A process $P2$ in blocked state can make transition E while another process $P2$ is in running state. - This is correct. There is no dependency between running process & Process getting out of blocked state.
3. The OS uses preemptive scheduling. :- This is true because we got transition C from Running to Ready.
4. The OS uses non-preemptive scheduling.Well as previous statement is true, this becomes false.

So answer is (**C**) II and III .

45 votes

-- Akash Kanase (36k points)

5.15.20 Process Scheduling: GATE CSE 2010 | Question: 25 top

https://gateoverflow.in/2204

Answer is (**D**).

I. In SRTF ,job with the shorest CPU burst will be scheduled first bcz of this process with large CPU burst may suffer from starvation

II. In preemptive scheduling , suppose process $P1$ is executing in CPU and after some time process $P2$ with high priority then $P1$ will arrive in ready queue then $p1$ is prrempted and $p2$ will brought into CPU for execution. In this way if process which is arriving in ready queue is of higher prioirity then $p1$, then $p1$ is always preempted and it may possible that it suffer from starvation.

III. Round robin will give better response time then FCFS ,in FCFS when process is executing ,it executed upto its complete burst time, but in round robin it will execute upto time quantum.

46 votes

-- neha pawar (3.3k points)

5.15.21 Process Scheduling: GATE CSE 2011 | Question: 35 top

https://gateoverflow.in/2137

Answer is (**A**).

5ms

Gantt Chart

P_0	P_1	P_1	P_0	P_2
0 – 1	1 – 2	2 – 5	5 – 13	13 – 22

$$\text{Average Waiting Time} = \frac{(0+4)+(0)+(11)}{3} = 5\text{ms}.$$

26 votes

-- Sona Praneeth Akula (3.4k points)

5.15.22 Process Scheduling: GATE CSE 2012 | Question: 31 top

https://gateoverflow.in/1749

FCFS First Come First Server

P_1	P_2	P_3
0 – 5	5 – 12	12 – 16

RR2

In Round Robin We are using the concept called Ready Queue.

Note
at $t = 2$,

- $P1$ finishes and sent to Ready Queue
- $P2$ arrives and schedules $P2$

This is the Ready Queue

$--$	$--$	$--$	P_3

At $t = 3$

- $P3$ arrives at ready queue

$--$	$--$	P_3	P_1

At $t = 4$

- $P1$ is scheduled as it is the first process to arrive at Ready Queue

P_1	P_2	P_1	P_3	P_2	P_1	P_3	P_2

Option (C) is correct

43 votes -- Akhil Nadh PC (16.5k points)

5.15.23 Process Scheduling: GATE CSE 2013 | Question: 10 top https://gateoverflow.in/1419

✔ **(B)** Because here the quanta for round robin is T units, after a process is scheduled it gets executed for T time units and waiting time becomes least and it again gets chance when every other process has completed T time units.

50 votes -- debanjan sarkar (2.9k points)

5.15.24 Process Scheduling: GATE CSE 2014 Set 1 | Question: 32 top https://gateoverflow.in/1803

✔

A	B	A	C	E	D

0 3 5 8 12 15 21

$$\text{Average Turnaround Time} = \frac{(8-0)+(5-3)+(12-5)+(21-7)+(15-10)}{5}$$

$$= \frac{36}{5} = 7.2$$

So, answer is $7.2\ ms$

28 votes -- Jay (831 points)

5.15.25 Process Scheduling: GATE CSE 2014 Set 2 | Question: 32 top https://gateoverflow.in/1991

✔ Gantt chart : $ABCABCBCBC$

C completes it CPU burst at$=500$ milli second.

IO time $= 500$ milli second
C completes 1st IO burst at $t = 500 + 500 = 1000$ ms

👍 49 votes

-- Digvijay (44.9k points)

5.15.26 Process Scheduling: GATE CSE 2014 Set 3 | Question: 32 top

https://gateoverflow.in/2066

✔

P_1	P_2	P_2	P_3	P_3	P_4	P_1
0 – 2	2 – 3	3 – 6	6 – 8	8 – 12	12 – 17	17 – 27

Gantt Chart

Process	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time = CT - BT -AT
P1	0	12	27	27	15
P2	2	4	6	4	0
P3	3	6	12	9	3
P4	8	5	17	9	4

Average Waiting Time $= (15 + 0 + 3 + 4)/4 = 5.5$ msec

👍 23 votes

-- Sourav Roy (2.9k points)

5.15.27 Process Scheduling: GATE CSE 2015 Set 1 | Question: 46 top

https://gateoverflow.in/8330

✔ Answer is **12**

T_1, T_2 and T_3 have infinite instances, meaning infinite burst times. Here, problem say Run "T_1 for 1 ms", "T_2 for 2 ms", and "T_3 for 4ms". i.e., every task is run in parts. Now for timing purpose we consider t for the **end** of cycle number t.

- $T_1 : 0, 3, 6, 9, 12, \dots \infty$ (T_1 repeats every 3 ms)
- $T_2 : 0, 7, 14, 21, \dots \infty$ (T_2 repeats every 7 ms)
- $T_3 : 0, 20, 40, 60, \dots \infty$ (T_3 repeats every 20 ms)

1. Priority of $T_1 = \frac{1}{3}$
2. Priority of $T_2 = \frac{1}{7}$
3. Priority of $T_3 = \frac{1}{20}$

Gantt Chart

0	T_1	T_2	T_2	T_1	T_3	T_3	T_1	T_2	T_2	T_1	T_3	T_3	....	......
	1	2	3	4	5	6	7	8	9	10	11	12		

At $t = 0$, No process is available
At $t = 2$, T_2 runs because it has higher priority than T_3 and no instance of T_1 present
At $t = 4$, We have T_1 arrive again and T_3 waiting but T_1 runs because it has higher priority
At $t = 5$, T_3 runs because no instance of T_1 or T_2 is present
At $t = 11$, T_3 runs because no instance of T_1 or T_2 is present
At $t = 12$, T_3 continue run because no instance of T_1 or T_2 is present and first instance of T_3 completes

👍 88 votes

-- Prashant Singh (47.2k points)

5.15.28 Process Scheduling: GATE CSE 2015 Set 3 | Question: 1 top

https://gateoverflow.in/8390

✔ (**D**) independent of n.

The number of processes that can be in READY state depends on the Ready Queue size and is independent of the number of CPU's.

👍 54 votes

-- Arjun Suresh (332k points)

5.15.29 Process Scheduling: GATE CSE 2015 Set 3 | Question: 34 top https://gateoverflow.in/8492

✔ Turn Around Time = Completion Time − Arrival Time

FCFS

Average turn around time $= [3 \text{ for A} + (2+6) \text{ for B} + (5+4) \text{ for C} + (7+2) \text{ for D}]/4 = 7.25$

Non-preemptive Shortest Job First

Average turn around time $= [3 \text{ for A} + (2+6) \text{ for B} + (3+2) \text{ for D} + (7+4) \text{ for C}] = 6.75$

Shortest Remaining Time

Average turn around time
$= [3 \text{ for A } + (2+1) \text{ for B } + (0+4) \text{ for C} + (2+2) \text{ for D} + (6+5) \text{ for remaining B }]/4 = 6.25$

Round Robin

Average turn around time =

$[2 \text{ for A (B comes after 1)}$
$+(1+2) \text{ for B \{C comes\}}$
$+(2+1) \text{ for A (A finishes after 3 cycles with turnaround time of } 2+3=5)$
$+(1+2) \text{ for C \{D comes\}}$
$+(3+2) \text{ for B}$
$+(3+2) \text{ for D (D finishes with turnaround time of } 3+2=5)$
$+(4+2) \text{ for C (C finishes with turnaround time of } 3+6=9)$
$+(4+2) \text{ for B (B finishes after turnaround time of } 3+5+6=14]$
$/4$
$= 8.25$

A	B	A	C	B	D	C	B

0 2 4 5 7 9 11 13 15

Shortest Remaining Time First scheduling which is the preemptive version of the SJF scheduling is provably optimal for the shortest waiting time and hence always gives the best (minimal) turn around time (waiting time + burst time). So, we can directly give the answer here.

Correct Answer: C

👍 43 votes

-- Arjun Suresh (332k points)

5.15.30 Process Scheduling: GATE CSE 2016 Set 1 | Question: 20 top https://gateoverflow.in/39655

✔ Answer should be (**A**) SRTF
SJF minimizes average waiting time. Probably optimal.
Now, here as all processes arrive at the same time, SRTF would be same as SJF. and hence, the answer.

Reference: [http://www.cs.columbia.edu/~junfeng/10sp-w4118/lectures/l13-sched.pdf See Slide 16,17 and 23](http://www.cs.columbia.edu/~junfeng/10sp-w4118/lectures/l13-sched.pdf)

References

👍 50 votes

-- Abhilash Panicker (7.6k points)

5.15.31 Process Scheduling: GATE CSE 2016 Set 2 | Question: 47 top https://gateoverflow.in/39625

✔ SRTF Preemptive hence,

P_1 P_2 P_3 P_2 P_4 P_1
0 3 7 8 10 13 20

Process TAT=Completion time − Arrival time
P_1 20
P_2 7
P_3 1
P_4 5

AvgTAT$= 33/4 = 8.25$

39 votes

-- Shashank Chavan (2.4k points)

5.15.32 Process Scheduling: GATE CSE 2017 Set 1 | Question: 24 top

https://gateoverflow.in/118304

✔

P_1	P_2	P_4	P_1	P_3

0 3 6 8 12 17

Gantt Chart

Process	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time = CT - BT -AT
P1	0	7	12	12	5
P2	3	3	6	3	0
P3	5	5	17	12	7
P4	6	2	8	2	0

Average Waiting Time $= \frac{(5+0+7+0)}{4} = 3$ milliseconds

35 votes

-- Ahwan Mishra (10.2k points)

5.15.33 Process Scheduling: GATE CSE 2017 Set 2 | Question: 51 top

https://gateoverflow.in/118558

✔ Gantt Chart for above problem looks like :

P_1	P_4	P_2	P_4	P_1	P_3	P_5

0 2 5 33 40 49 51 67

Waiting Time = Completion time − Arrival time − Burst Time

$\sum AT = 0 + 5 + 12 + 2 + 9 = 28$

$\sum BT = 11 + 28 + 2 + 10 + 16 = 67$

$\sum CT = 67 + 51 + 49 + 40 + 33 = 240$

Waiting time $= 240 - 28 - 67 = 145$

Average Waiting Time $= \frac{145}{5} = 29$ msec.

55 votes

-- Manish Joshi (20.5k points)

5.15.34 Process Scheduling: GATE CSE 2019 | Question: 41 top

https://gateoverflow.in/302807

$P1$	$P2$	$P1$	
0	1	2	4

Till $t = 4$, the waiting time of $P1 = 1$ and $P2 = 0$ and $P3 = 1$ but $P3$ has not started yet.

Case 1:

Note that if $P4$ burst time is less than $P3$ then $P4$ will complete and after that $P3$ will complete. Therefore Waiting time of $P4$ should be 0. And total waiting time of $P3 = 1 +$ (Burst time of $P4$) because until $P4$ completes $P3$ does not get a chance.

Then average waiting time $= \frac{1+0+(1+x)+0}{4} = 1$

$\frac{2+x}{4} = 1 \Rightarrow x = 2.$

Case 2:

Note that if $P4$ burst time is greater than $P3$ then $P4$ will complete after $P3$ will complete. Therefore, Waiting time of $P3$ remains the same. And total waiting time of $P4 =$ (Burst time of $P3$) because until $P3$ completes $P4$ does not get a chance.

Then average waiting time $= \frac{1+0+1+3}{4} = 1$

$\frac{5}{4} \neq 1 \Rightarrow$ This case is invalid.

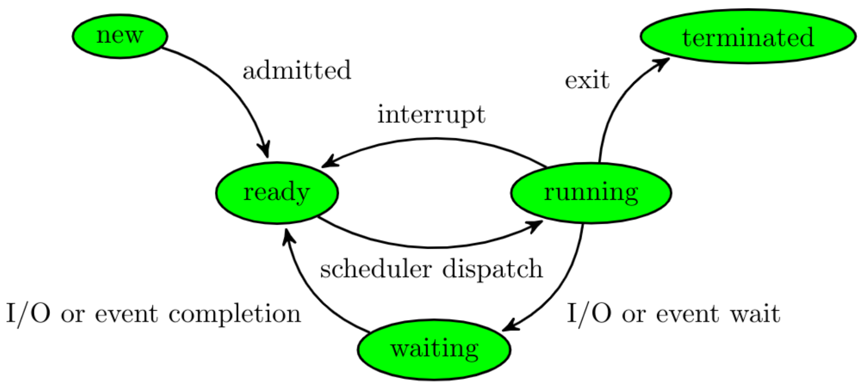
Correct Answer: 2

31 votes

-- Shaik Masthan (50.4k points)

5.15.35 Process Scheduling: GATE CSE 2020 | Question: 12 top https://gateoverflow.in/333219

A blocked process cannot go to running state directly. Except (III), every option is viable.



Answer-(C)

14 votes

-- Ayush Upadhyaya (28.4k points)

5.15.36 Process Scheduling: GATE CSE 2020 | Question: 50 top https://gateoverflow.in/333181

SJF:

Process	Burst Time	Completion Time	Turn Around Time
P_1	8	21	21
P_2	7	13	13
P_3	2	2	2
P_4	4	6	6

Average Turn-Around Time : $\frac{21+13+2+6}{4} = 10.5$

RR:

Process	Burst Time	Completion Time	Turn Around Time
P_1	8	18	18
P_2	7	21	21
P_3	2	10	10
P_4	4	14	14

Average Turn-Around Time : $\frac{18+21+10+14}{4} = 15.75$

Absolute Difference $= \mid 10.5 - 15.75 \mid = 5.25.$

5 votes

-- Aditya Patel (775 points)

5.15.37 Process Scheduling: GATE CSE 2021 Set 1 | Question: 25 top

https://gateoverflow.in/357426

✔ We **get minimum achievable average waiting time using SJF scheduling**.

Lets just name these processes for explanation purpose only as $A = 16, B = 20$ and $C = 10$.

Order them according to burst time as $C < A < B$

C will not wait for anyone, schedule first (wait time = 0)

A will wait for only C (wait time = 10)

B will wait for both C and A (wait time = 10 + 16)

Average wait time = $\frac{0+10+(10+16)}{3} = \frac{36}{3} = 12.$

No need to make any table or chart.

This is all for explaining purpose, you can actually ans this within 10-15 sec after reading the complete question.

2 votes

-- Nikhil Dhama (2.5k points)

5.15.38 Process Scheduling: GATE CSE 2021 Set 2 | Question: 14 top

https://gateoverflow.in/357526

✔

A. Turnaround time includes waiting time
   - TRUE. $\text{Turnaround Time} = \text{Waiting Time} + \text{Burst Time}$

B. The goal is to only maximize CPU utilization and minimize throughput
   - FALSE. CPU scheduling must aim to maximize CPU utilization as well as throughput. Throughput of CPU scheduling is defined as the number of processes completed in unit time. SJF scheduling gives the highest throughput.

C. Round-robin policy can be used even when the CPU time required by each of the processes is not known apriori
   - TRUE. Round-robin scheduling gives a fixed time quantum to each process and for this there is no requirement to know the CPU time of the process apriori (which is not the case say for shortest remaining time first).

D. Implementing preemptive scheduling needs hardware support
   - TRUE. Preemptive scheduling needs hardware support to manage context switch which includes saving the execution state of the current process and then loading the next process.

Correct Answer: A;C;D

Reference: Stanford Notes

References

1 votes

-- Arjun Suresh (332k points)

5.15.39 Process Scheduling: GATE IT 2005 | Question: 60 top

https://gateoverflow.in/3821

✔ Answer will be (**D**).

TAT = Completion Time - Arrival Time.

The Gannt Chart for Non Preemptive scheduling will be $(0)P3, (15)P1, (35)P2(45)$.

From above this can be inferred easily that completion time for $P2$ is 45, for $P1$ is 35 and $P3$ is 15.

Gantt Chart for Preemptive- $(0)P3, (1)P3, (2)P3, (3)P2, (4)P2, (5)P1, (25)P2, (33)P3(45)$.

Similarly take completion time from above for individual processes and subtract it from the Arrival time to get TAT.

31 votes

-- Gate Keeda (15.9k points)

5.15.40 Process Scheduling: GATE IT 2006 | Question: 12 top

https://gateoverflow.in/3551

✔ Extract from Galvin "*If there are enough extra frames, another process can be initiated. If the sum of the working-set sizes increases, exceeding the total number of available frames,the operating system selects a process to suspend. The process's pages are written out (swapped), and its frames are reallocated to other processes. The suspended process can be restarted later.*"

So Option (D)

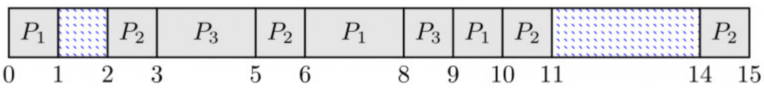
57 votes

-- Danish (3.4k points)

5.15.41 Process Scheduling: GATE IT 2006 | Question: 54 top

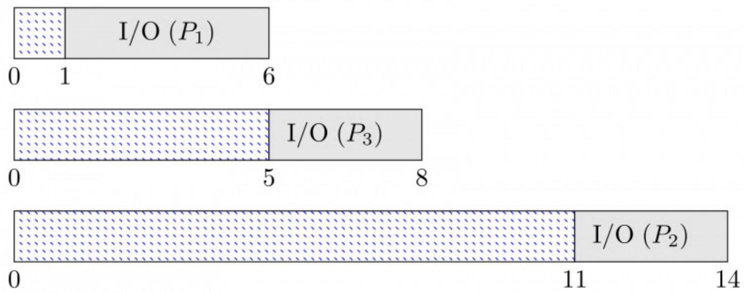
https://gateoverflow.in/3597

✔



GIVEN : assuming that each process has its own i/o resource.

(GANTT CHART FOR I/O OF PROCESSOR P_1, P_2, P_3)



EXPLANATION :

Here, P_2 has the least priority and P_1 has the highest.

P_1 enters CPU at 0 and utilizes it for 1 time unit. Then it performs i/o for 5 time units.

Then P_2 enters at time unit 2 and requires 3 time units of CPU. But P_3 whose priority is greater than P_2 arrives at time unit 3.

So, P_2 IS ***PREEMPTED*** (only 1 unit of P_2 is done out of 3 units. Therefore 2 units of P_2 are left out) AND P_3 ACQUIRES THE CPU. Once P_3 finishes, P_2 enters the CPU to complete its pending 2 units job at time unit 5. AGAIN BY THEN P_1 finishes its i/o and arrives with a higher priority. Therefore of 2 units P_2 performs only one unit and the CPU is given to P_1.Then when P_1 is performing in CPU, P_3 completes its i/o and arrives with a higher priority.Thus the CPU is given to P_3 (1 UNIT IS USED). **P_3 FINISHES AT TIME UNIT 9**. NOW PRIORITY OF P_1 IS MORE THAN P_2, SO, CPU IS USED BY P_1. **P_1 FINISHES BY TIME UNIT 10**. THEN CPU IS ALLOCATED FOR PROCESS P_2. **P_2 PERFORMS REST OF ITS WORK AND FINISHES AT TIME UNIT 15.**

THEREFORE,

FINISH TIME OF P_1, P_2, P_3 ARE $10, 15$ AND 9 RESPECTIVELY. 😊

Correct Answer: B

29 votes

-- Tejashwini B (139 points)

5.15.42 Process Scheduling: GATE IT 2007 | Question: 26 top https://gateoverflow.in/3459

✔ Lets take an example:

Process	Weight	Execution time
P_1	1	3
P_2	2	5
P_3	3	2
$p4$	4	4

For option **1** non decreasing t_i

$$= (3 \times 2 + 1 \times 5 + 4 \times 9 + 2 \times 14)/10 = (6 + 5 + 36 + 28)/10 = 7.5$$

For option **2** non increasing w_i

$$= (4 \times 4 + 3 \times 6 + 2 \times 11 + 1 \times 14)/10 = (16 + 18 + 22 + 14)/10 = 7$$

For option **3** non increasing $w_i t_i$

$$= (16 + 2 \times 9 + 3 \times 11 + 1 \times 14)/10 = (16 + 18 + 33 + 14)/10 = 8.1$$

For option **4** non increasing w_i/t_i

$$= (3 \times 2 + 4 \times 6 + 2 \times 11 + 1 \times 14)/10 = (6 + 10 + 22 + 14)/10 = 6.6$$

Minimum weighted mean obtained from non increasing w_i/t_i **(option D)**

The solution above is a classical example of greedy algorithm - that is at every point we choose the best available option and this leads to a global optimal solution. In this problem, we require to minimize the weighted mean completion time and the denominator in it is independent of the order of execution of the jobs. So, we just need to focus on the numerator and try to reduce it. Numerator here is a factor of the job weight and its completion time and since both are multiplied, our greedy solution must be

- to execute the shorter jobs first (so that remaining jobs have smaller completion time) and
- to execute highest weighted jobs first (so that it is multiplied by smaller completion time)

So, combining both we can use w_i/t_i to determine the execution order of processes - which must then be executed in non-increasing order.

111 votes -- khush tak (5.9k points)

5.15.43 Process Scheduling: GATE IT 2008 | Question: 55 top https://gateoverflow.in/3365

✔ Answer is (**C**).

RR follows FCFS with time slice if time slice is larger than the max time required to execute any process then it is simply converged into fcfs as every process will finish in first cycle itself

29 votes -- sanjay (36.6k points)

5.16 Process Synchronization (51) top

5.16.1 Process Synchronization: GATE CSE 1987 | Question: 1-xvi top https://gateoverflow.in/80362

A critical region is

A. One which is enclosed by a pair of P and V operations on semaphores.
B. A program segment that has not been proved bug-free.
C. A program segment that often causes unexpected system crashes.
D. A program segment where shared resources are accessed.

gate1987 operating-system process-synchronization

Answer

5.16.2 Process Synchronization: GATE CSE 1987 | Question: 8a top https://gateoverflow.in/82433

Consider the following proposal to the "readers and writers problem."

Shared variables and semaphores:

```
aw, ar, rw, rr : interger;
mutex, reading, writing: semaphore:
initial values of variables and states of semaphores:
ar=rr=aw=rw=0
reading_value = writing_value = 0
mutex_value = 1.                        Process writer;
Process reader;                             begin
begin                                    while true do
repeat                                   begin
         P(mutex);                              P(mutex);
         ar := ar+1;                            aw := aw + 1;
         grantread;                             grantwrite;
         V(mutex);                              V(mutex);
         P(reading);                            P(writing);
         read;                         Write;
         P(mutex);                     P(mutex);
         rr := rr - 1;                 rw := rw - 1;
         ar := ar - 1;                 ar := aw - 1;
         grantwrite;                   grantread;
         V(mutex);                     V(mutex);
         other-work;                   other-work;
until false                     end
end.                            end.

Procedure grantread;
begin
    if aw = 0
    then while (rr < ar) do
        begin rr := rr + 1;
            V (reading)
        end
end;
Procedure grantwrite;
begin
    if rr = 0
    then while (rw < aw) do
        begin rw := rw + 1;
            V (writing)
        end
end;
```

a. Give the value of the shared variables and the states of semaphores when 12 readers are reading and writers are writing.
b. Can a group of readers make waiting writers starve? Can writers starve readers?
c. Explain in two sentences why the solution is incorrect.

gate1987 operating-system process-synchronization descriptive

Answer

5.16.3 Process Synchronization: GATE CSE 1988 | Question: 10iib top https://gateoverflow.in/94393

Given below is solution for the critical section problem of two processes P_0 and P_1 sharing the following variables:

```
    var flag :array [0..1] of boolean; (initially false)
        turn: 0 .. 1;
```

The program below is for process $P_i (i = 0 \text{ or } 1)$ where process $P_j (j = 1 \text{ or } 0)$ being the other one.

```
repeat
        flag[i]:= true;
        while turn != i
        do begin
            while flag [j] do skip
            turn:=i;
        end

        critical section

        flag[i]:=false;
until false
```

Determine of the above solution is correct. If it is incorrect, demonstrate with an example how it violates the conditions.

gate1988 descriptive operating-system process-synchronization

Answer

5.16.4 Process Synchronization: GATE CSE 1990 | Question: 2-iii top https://gateoverflow.in/83859

Match the pairs:

(a)	Critical region	(p)	Hoare's monitor
(b)	Wait/Signal	(q)	Mutual exclusion
(c)	Working Set	(r)	Principle of locality
(d)	Deadlock	(s)	Circular Wait

match-the-following gate1990 operating-system process-synchronization

Answer

5.16.5 Process Synchronization: GATE CSE 1991 | Question: 11,a top

https://gateoverflow.in/538

Consider the following scheme for implementing a critical section in a situation with three processes P_i, P_j and P_k.

```
Pi;
repeat
    flag[i] := true;
    while flag [j] or flag[k] do
        case turn of
        j: if flag [j] then
        begin
            flag [i] := false;
            while turn != i do skip;
            flag [i] := true;
        end;
        k: if flag [k] then
        begin
            flag [i] := false,
            while turn != i do skip;
            flag [i] := true
        end
    end
    critical section
    if turn = i then turn := j;
        flag [i] := false
    non-critical section
until false;
```

a. Does the scheme ensure mutual exclusion in the critical section? Briefly explain.

gate1991 process-synchronization normal operating-system descriptive

Answer

5.16.6 Process Synchronization: GATE CSE 1991 | Question: 11,b top

https://gateoverflow.in/43000

Consider the following scheme for implementing a critical section in a situation with three processes P_i, P_j and P_k.

Pi;

```
repeat
    flag[i] := true;
    while flag [j] or flag[k] do
        case turn of
        j: if flag [j] then
        begin
            flag [i] := false;
            while turn != i do skip;
            flag [i] := true;
        end;
        k: if flag [k] then
        begin
            flag [i] := false,
            while turn != i do skip;
            flag [i] := true
        end
    end
    critical section
    if turn = i then turn := j;
        flag [i] := false
    non-critical section
until false;
```

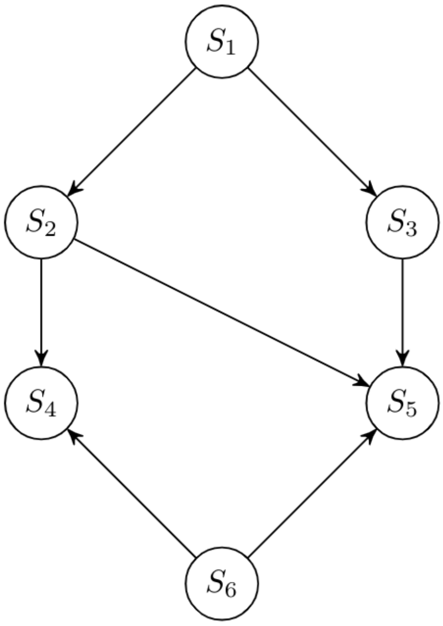
Is there a situation in which a waiting process can never enter the critical section? If so, explain and suggest modifications to the code to solve this problem

gate1991 process-synchronization normal operating-system descriptive

Answer

5.16.7 Process Synchronization: GATE CSE 1993 | Question: 22 top https://gateoverflow.in/2319

Write a concurrent program using parbegin-parend and semaphores to represent the precedence constraints of the statements S_1 to S_6, as shown in figure below.



gate1993 operating-system process-synchronization normal descriptive

Answer

5.16.8 Process Synchronization: GATE CSE 1994 | Question: 27 top https://gateoverflow.in/2523

A. Draw a precedence graph for the following sequential code. The statements are numbered from S_1 to S_6

S_1	read n
S_2	i := 1
S_3	if i > n next
S_4	a(i) := i+1
S_5	i := i+1
S_6	next : write a(i)

B. Can this graph be converted to a concurrent program using parbegin-parend construct only?

gate1994 operating-system process-synchronization normal descriptive

Answer

5.16.9 Process Synchronization: GATE CSE 1995 | Question: 19 top https://gateoverflow.in/2656

Consider the following program segment for concurrent processing using semaphore operators P and V for synchronization. Draw the precedence graph for the statements S_1 to S_9.

```
var
a,b,c,d,e,f,g,h,i,j,k : semaphore;
begin
cobegin
    begin S1; V(a); V(b) end;
    begin P(a); S2; V(c); V(d) end;
    begin P(c); S4; V(e) end;
    begin P(d); S5; V(f) end;
    begin P(e); P(f); S7; V(k) end
    begin P(b); S3; V(g); V(h) end;
    begin P(g); S6; V(i) end;
    begin P(h); P(i); S8; V(j) end;
    begin P(j); P(k); S9 end;
coend
end;
```

gate1995 operating-system process-synchronization normal descriptive

Answer

5.16.10 Process Synchronization: GATE CSE 1996 | Question: 1.19, ISRO2008-61 top https://gateoverflow.in/2723

A critical section is a program segment

A. which should run in a certain amount of time
B. which avoids deadlocks
C. where shared resources are accessed
D. which must be enclosed by a pair of semaphore operations, P and V

gate1996 operating-system process-synchronization easy isro2008

Answer

5.16.11 Process Synchronization: GATE CSE 1996 | Question: 2.19 top https://gateoverflow.in/2748

A solution to the Dining Philosophers Problem which avoids deadlock is to

A. ensure that all philosophers pick up the left fork before the right fork
B. ensure that all philosophers pick up the right fork before the left fork
C. ensure that one particular philosopher picks up the left fork before the right fork, and that all other philosophers pick up the right fork before the left fork
D. None of the above

gate1996 operating-system process-synchronization normal

Answer

5.16.12 Process Synchronization: GATE CSE 1996 | Question: 21 top https://gateoverflow.in/2773

The concurrent programming constructs fork and join are as below:

Fork <label> which creates a new process executing from the specified label

Join <variable> which decrements the specified synchronization variable (by 1) and terminates the process if the new value is not 0.

Show the precedence graph for $S1$, $S2$, $S3$, $S4$, and $S5$ of the concurrent program below.

$N = 2$
$M = 2$
Fork $L3$
Fork $L4$
$S1$
$L1$: join N
$S3$
$L2$: join M
$S5$
$L3$: $S2$
Goto $L1$
$L4$: $S4$
Goto $L2$
Next:

gate1996 operating-system process-synchronization normal descriptive

Answer

5.16.13 Process Synchronization: GATE CSE 1997 | Question: 6.8 top https://gateoverflow.in/2264

Each Process $P_i, i = 1 \dots 9$ is coded as follows

```
repeat
    P(mutex)
    {Critical section}
    V(mutex)
forever
```

The code for P_{10} is identical except it uses V(mutex) in place of P(mutex). What is the largest number of processes that can be inside the critical section at any moment?

A. 1
B. 2
C. 3
D. None

gate1997 operating-system process-synchronization normal

Answer

5.16.14 Process Synchronization: GATE CSE 1997 | Question: 73 top

https://gateoverflow.in/19703

A concurrent system consists of 3 processes using a shared resource R in a non-preemptible and mutually exclusive manner. The processes have unique priorities in the range $1 \dots 3$, 3 being the highest priority. It is required to synchronize the processes such that the resource is always allocated to the highest priority requester. The pseudo code for the system is as follows.

Shared data

```
mutex:semaphore = 1:/* initialized to 1*/

process[3]:semaphore = 0; /*all initialized to 0 */

R_requested [3]:boolean = false; /*all initialized to flase */

busy: boolean = false; /*initialized to false */
```

Code for processes

```
begin process
my-priority:integer;
my-priority:=___; /*in the range 1..3*/
repeat
    request_R(my-priority);
    P (proceed [my-priority]);
    {use shared resource R}
    release_R (my-priority);
forever
end process;
```

Procedures

```
procedure request_R(priority);
P(mutex);
if busy = true then
    R_requested [priority]:=true;
else
begin
    V(proceed [priority]);
    busy:=true;
end
V(mutex)
```

Give the pseudo code for the procedure release_R.

gate1997 operating-system process-synchronization descriptive

Answer

5.16.15 Process Synchronization: GATE CSE 1998 | Question: 1.30 top

https://gateoverflow.in/1667

When the result of a computation depends on the speed of the processes involved, there is said to be

A. cycle stealing
B. race condition
C. a time lock
D. a deadlock

gate1998 operating-system easy process-synchronization

Answer

5.16.16 Process Synchronization: GATE CSE 1999 | Question: 20-a top https://gateoverflow.in/1519

A certain processor provides a 'test and set' instruction that is used as follows:

```
TSET register, flag
```

This instruction atomically copies flag to register and sets flag to 1. Give pseudo-code for implementing the entry and exit code to a critical region using this instruction.

gate1999 operating-system process-synchronization normal descriptive

Answer

5.16.17 Process Synchronization: GATE CSE 1999 | Question: 20-b top https://gateoverflow.in/205817

Consider the following solution to the producer-consumer problem using a buffer of size 1. Assume that the initial value of count is 0. Also assume that the testing of count and assignment to count are atomic operations.

```
Producer:
Repeat
    Produce an item;
    if count = 1 then sleep;
    place item in buffer.
    count = 1;
    Wakeup(Consumer);
Forever

Consumer:
Repeat
    if count = 0 then sleep;
    Remove item from buffer;
    count = 0;
    Wakeup(Producer);
    Consume item;
Forever;
```

Show that in this solution it is possible that both the processes are sleeping at the same time.

gate1999 operating-system process-synchronization normal descriptive

Answer

5.16.18 Process Synchronization: GATE CSE 2000 | Question: 1.21 top https://gateoverflow.in/645

Let $m[0] \dots m[4]$ be mutexes (binary semaphores) and $P[0] \dots P[4]$ be processes.
Suppose each process $P[i]$ executes the following:

```
        wait (m[i]); wait (m(i+1) mod 4]);
        ...........
        release (m[i]); release (m(i+1) mod 4]);
```

This could cause

A. Thrashing
B. Deadlock
C. Starvation, but not deadlock
D. None of the above

gate2000-cse operating-system process-synchronization normal

Answer

5.16.19 Process Synchronization: GATE CSE 2000 | Question: 20 top https://gateoverflow.in/691

a. Fill in the boxes below to get a solution for the reader-writer problem, using a single binary semaphore, mutex (initialized to 1) and busy waiting. Write the box numbers (1, 2 and 3), and their contents in your answer book.

```
int R = 0, W = 0;

Reader () {
    wait (mutex);
    if (W == 0) {
        R = R + 1;
        ▭ ______________(1)
    }
    else {
      ▭ ______________(2)
        goto L1;
    }
    ..../* do the read*/
    wait (mutex);
    R = R - 1;
    signal (mutex);
}
```

L1:

```
Writer () {
    wait (mutex);
    if (▭) { _________ (3)
        signal (mutex);
        goto L2;
    }
    W=1;
    signal (mutex);
    ...../*do the write*/
    wait( mutex);
    W=0;
    signal (mutex);
}
```

L2:

b. Can the above solution lead to starvation of writers?

gate2000-cse operating-system process-synchronization normal descriptive

Answer

5.16.20 Process Synchronization: GATE CSE 2001 | Question: 2.22 top https://gateoverflow.in/740

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

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

gate2001-cse operating-system process-synchronization normal

Answer

5.16.21 Process Synchronization: GATE CSE 2002 | Question: 18-a top https://gateoverflow.in/871

Draw the process state transition diagram of an OS in which (i) each process is in one of the five states: created, ready, running, blocked (i.e., sleep or wait), or terminated, and (ii) only non-preemptive scheduling is used by the OS. Label the transitions appropriately.

gate2002-cse operating-system process-synchronization normal descriptive

Answer

5.16.22 Process Synchronization: GATE CSE 2002 | Question: 18-b top https://gateoverflow.in/205818

The functionality of atomic TEST-AND-SET assembly language instruction is given by the following C function

```
int TEST-AND-SET (int *x)
{
    int y;
```

```
    A1: y=*x;
    A2: *x=1;
    A3: return y;
}
```


i. Complete the following C functions for implementing code for entering and leaving critical sections on the above TEST-AND-SET instruction.

```
int mutex=0;
void enter-cs()
{
    while(......................);

}
void leave-cs()
{ .........................;

}
```

ii. Is the above solution to the critical section problem deadlock free and starvation-free?

iii. For the above solution, show by an example that mutual exclusion is not ensured if TEST-AND-SET instruction is not atomic?

gate2002-cse operating-system process-synchronization normal descriptive

Answer

5.16.23 Process Synchronization: GATE CSE 2002 | Question: 20 top

https://gateoverflow.in/873

The following solution to the single producer single consumer problem uses semaphores for synchronization.

```
#define BUFFSIZE 100
buffer buf[BUFFSIZE];
int first = last = 0;
semaphore b_full = 0;
semaphore b_empty = BUFFSIZE

void producer()
{
while(1) {
    produce an item;
    p1:.................;
    put the item into buff (first);
    first = (first+1)%BUFFSIZE;
    p2: ...............;
    }
}

void consumer()
{
while(1) {
    c1:...........
    take the item from buf[last];
    last = (last+1)%BUFFSIZE;
    c2:...........;
    consume the item;
    }
}
```

A. Complete the dotted part of the above solution.

B. Using another semaphore variable, insert one line statement each immediately after $p1$, immediately before $p2$, immediately after $c1$ and immediately before $c2$ so that the program works correctly for multiple producers and consumers.

gate2002-cse operating-system process-synchronization normal descriptive

Answer

5.16.24 Process Synchronization: GATE CSE 2003 | Question: 80 top

https://gateoverflow.in/964

Suppose we want to synchronize two concurrent processes P and Q using binary semaphores S and T. The code for the processes P and Q is shown below.

Process P:	**Process Q:**
`while(1){` `W:` `    print '0';` `    print '0';` `X:` `}`	`while(1){` `Y:` `    print '1';` `    print '1';` `Z:` `}`

Synchronization statements can be inserted only at points $W, X, Y,$ and Z

Which of the following will always lead to an output staring with '001100110011'?

A. $P(S)$ at $W, V(S)$ at $X, P(T)$ at $Y, V(T)$ at Z, S and T initially 1

B. $P(S)$ at $W, V(T)$ at $X, P(T)$ at $Y, V(S)$ at Z, S initially 1, and T initially 0

C. $P(S)$ at $W, V(T)$ at $X, P(T)$ at $Y, V(S)$ at Z, S and T initially 1

D. $P(S)$ at $W, V(S)$ at $X, P(T)$ at $Y, V(T)$ at Z, S initially 1 , and T initially 0

gate2003-cse operating-system process-synchronization normal

Answer

5.16.25 Process Synchronization: GATE CSE 2003 | Question: 81 top

https://gateoverflow.in/43574

Suppose we want to synchronize two concurrent processes P and Q using binary semaphores S and T. The code for the processes P and Q is shown below.

Process P:	Process Q:
while(1) {	while(1) {
W:	Y:
print '0';	print '1';
print '0';	print '1';
X:	Z:
}	}

Synchronization statements can be inserted only at points $W, X, Y,$ and Z

Which of the following will ensure that the output string never contains a substring of the form 01^n0 and 10^n1 where n is odd?

A. $P(S)$ at $W, V(S)$ at $X, P(T)$ at $Y, V(T)$ at Z, S and T initially 1
B. $P(S)$ at $W, V(T)$ at $X, P(T)$ at $Y, V(S)$ at Z, S and T initially 1
C. $P(S)$ at $W, V(S)$ at $X, P(S)$ at $Y, V(S)$ at Z, S initially 1
D. $V(S)$ at $W, V(T)$ at $X, P(S)$ at $Y, P(T)$ at Z, S and T initially 1

gate2003-cse operating-system process-synchronization normal

Answer

5.16.26 Process Synchronization: GATE CSE 2004 | Question: 48 top

https://gateoverflow.in/1044

Consider two processes P_1 and P_2 accessing the shared variables X and Y protected by two binary semaphores S_X and S_Y respectively, both initialized to 1. P and V denote the usual semaphore operators, where P decrements the semaphore value, and V increments the semaphore value. The pseudo-code of P_1 and P_2 is as follows:

P_1:	P_2:
While true do {	While true do {
$L_1 : \ldots\ldots$	$L_3 : \ldots\ldots$
$L_2 : \ldots\ldots$	$L_4 : \ldots\ldots$
$\mathrm{X} = \mathrm{X} + 1;$	$\mathrm{Y} = \mathrm{Y} + 1;$
$Y = Y - 1;$	$X = Y - 1;$
$V(S_X);$	$V(S_Y);$
$V(S_Y);$	$V(S_X);$
}	}

In order to avoid deadlock, the correct operators at L_1, L_2, L_3 and L_4 are respectively.

A. $P(S_Y), P(S_X); P(S_X), P(S_Y)$

B. $P(S_X), P(S_Y); P(S_Y), P(S_X)$

C. $P(S_X), P(S_X); P(S_Y), P(S_Y)$

D. $P(S_X), P(S_Y); P(S_X), P(S_Y)$

gate2004-cse operating-system process-synchronization normal

Answer

5.16.27 Process Synchronization: GATE CSE 2006 | Question: 61 top

https://gateoverflow.in/1839

The atomic *fetch-and-set* x, y instruction unconditionally sets the memory location x to 1 and fetches the old value of x in y without allowing any intervening access to the memory location x. Consider the following implementation of P and V functions on a binary semaphore S.

```
void P (binary_semaphore *s) {
    unsigned y;
    unsigned *x = &(s->value);
    do {
        fetch-and-set x, y;
    } while (y);
}

void V (binary_semaphore *s) {
    S->value = 0;
}
```

Which one of the following is true?

A. The implementation may not work if context switching is disabled in P
B. Instead of using *fetch-and –set*, a pair of normal load/store can be used
C. The implementation of V is wrong
D. The code does not implement a binary semaphore

gate2006-cse operating-system process-synchronization normal

Answer

5.16.28 Process Synchronization: GATE CSE 2006 | Question: 78 top

https://gateoverflow.in/1853

Barrier is a synchronization construct where a set of processes synchronizes globally i.e., each process in the set arrives at the barrier and waits for all others to arrive and then all processes leave the barrier. Let the number of processes in the set be three and S be a binary semaphore with the usual P and V functions. Consider the following C implementation of a barrier with line numbers shown on left.

void barrier (void) {

```
1: P(S);
2: process_arrived++;
3: V(S);
4: while (process_arrived !=3);
5: P(S);
6: process_left++;
7: if (process_left==3) {
8:     process_arrived = 0;
9:     process_left = 0;
10: }
11: V(S);
```

}

The variables *process_arrived* and *process_left* are shared among all processes and are initialized to zero. In a concurrent program all the three processes call the barrier function when they need to synchronize globally.

The above implementation of barrier is incorrect. Which one of the following is true?

A. The barrier implementation is wrong due to the use of binary semaphore S
B. The barrier implementation may lead to a deadlock if two barrier in invocations are used in immediate succession.
C. Lines 6 to 10 need not be inside a critical section
D. The barrier implementation is correct if there are only two processes instead of three.

gate2006-cse operating-system process-synchronization normal

Answer

5.16.29 Process Synchronization: GATE CSE 2006 | Question: 79 top

https://gateoverflow.in/43564

Barrier is a synchronization construct where a set of processes synchronizes globally i.e., each process in the set arrives at the barrier and waits for all others to arrive and then all processes leave the barrier. Let the number of processes in the set be three and S be a binary semaphore with the usual P and V functions. Consider the following C implementation of a barrier with line numbers shown on left.

void barrier (void) {

```
1 P(S);
2 process_arrived++;
3 V(S);
4 while (process_arrived !=3);
5    P(S);
6    process_left++;
7    if (process_left==3) {
8       process_arrived = 0;
9       process_left = 0;
10   }
11 V(S);
```

}

The variables process_arrived and process_left are shared among all processes and are initialized to zero. In a concurrent program all the three processes call the barrier function when they need to synchronize globally.

Which one of the following rectifies the problem in the implementation?

A. Lines 6 to 10 are simply replaced by process_arrived--
B. At the beginning of the barrier the first process to enter the barrier waits until process_arrived becomes zero before proceeding to execute $P(S)$.
C. Context switch is disabled at the beginning of the barrier and re-enabled at the end.
D. The variable process_left is made private instead of shared

gate2006-cse operating-system process-synchronization normal

Answer

5.16.30 Process Synchronization: GATE CSE 2007 | Question: 58

top

https://gateoverflow.in/1256

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

/* P1 */	/* P2 */
``while (true) { wants1 = true; while (wants2 == true); /* Critical Section */ wants1 = false; } /* Remainder section */```	```while (true) { wants2 = true; while (wants1 == true); /* Critical Section */ wants2=false; } /* Remainder section */```

```
/*  P1   */
while (true) {
    wants1 = true;
    while (wants2 == true);
    /* Critical Section */
    wants1 = false;
}
/* Remainder section */
```

```
/*  P2   */
while (true) {
    wants2 = true;
    while (wants1 == true);
    /* Critical Section */
    wants2=false;
}
/* Remainder section */
```

Here, wants1 and wants2 are shared variables, which are initialized to false.

Which one of the following statements is TRUE about the 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 alteration.

D. It does not prevent deadlocks, but ensures mutual exclusion.

gate2007-cse operating-system process-synchronization normal

Answer

5.16.31 Process Synchronization: GATE CSE 2009 | Question: 33

top

https://gateoverflow.in/1319

The **enter_CS()** and **leave_CS()** functions to implement critical section of a process are realized using test-and-set instruction as follows:

```
void enter_CS(X)
{
    while(test-and-set(X));
}

void leave_CS(X)
{
    X = 0;
}
```

In the above solution, X is a memory location associated with the CS and is initialized to 0. Now consider the following statements:

I. The above solution to CS problem is deadlock-free

II. The solution is starvation free

III. The processes enter CS in FIFO order

IV. More than one process can enter CS at the same time

Which of the above statements are TRUE?

A. (I) only
B. (I) and (II)
C. (II) and (III)
D. (IV) only

gate2009-cse operating-system process-synchronization normal

Answer

5.16.32 Process Synchronization: GATE CSE 2010 | Question: 23 top

https://gateoverflow.in/2202

Consider the methods used by processes $P1$ and $P2$ for accessing their critical sections whenever needed, as given below. The initial values of shared boolean variables $S1$ and $S2$ are randomly assigned.

Method used by P1	**Method used by P2**
while (S1 == S2);	while (S1 != S2);
Critical Section	Critical Section
S1 = S2;	S2 = not(S1);

Which one of the following statements describes the properties achieved?

A. Mutual exclusion but not progress
B. Progress but not mutual exclusion
C. Neither mutual exclusion nor progress
D. Both mutual exclusion and progress

gate2010-cse operating-system process-synchronization normal

Answer

5.16.33 Process Synchronization: GATE CSE 2010 | Question: 45 top

https://gateoverflow.in/2347

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

Process P0	Process P1	Process P2
while (true) {	wait (S1);	wait (S2);
wait (S0);	release (S0);	release (S0);
print '0';		
release (S1);		
release (S2);		
}		

How many times will process $P0$ print '0'?

A. At least twice
B. Exactly twice
C. Exactly thrice
D. Exactly once

gate2010-cse operating-system process-synchronization normal

Answer

5.16.34 Process Synchronization: GATE CSE 2012 | Question: 32 top

https://gateoverflow.in/1750

Fetch_And_Add(X,i) is an atomic Read-Modify-Write instruction that reads the value of memory location X, increments it by the value i, and returns the old value of X. It is used in the pseudocode shown below to implement a busy-wait lock. L is an unsigned integer shared variable initialized to 0. The value of 0 corresponds to lock being available, while any non-zero value corresponds to the lock being not available.

```
AcquireLock(L){
   while (Fetch_And_Add(L,1))
       L = 1;
}

ReleaseLock(L){
   L = 0;
}
```

This implementation

A. fails as L can overflow
B. fails as L can take on a non-zero value when the lock is actually available
C. works correctly but may starve some processes
D. works correctly without starvation

gate2012-cse operating-system process-synchronization normal

Answer

5.16.35 Process Synchronization: GATE CSE 2013 | Question: 34 top

https://gateoverflow.in/1545

A shared variable x, initialized to zero, is operated on by four concurrent processes W, X, Y, Z as follows. Each of the processes W and X reads x from memory, increments by one, stores it to memory, and then terminates. Each of the processes Y and Z reads x from memory, decrements by two, stores it to memory, and then terminates. Each process before reading x invokes the P operation (i.e., wait) on a counting semaphore S and invokes the V operation (i.e., signal) on the semaphore S after storing x to memory. Semaphore S is initialized to two. What is the maximum possible value of x after all processes complete execution?

A. -2
B. -1
C. 1
D. 2

gate2013-cse operating-system process-synchronization normal

Answer

5.16.36 Process Synchronization: GATE CSE 2013 | Question: 39 top

https://gateoverflow.in/1550

A certain computation generates two arrays a and b such that $a[i] = f(i)$ for $0 \leq i < n$ and $b[i] = g(a[i])$ for $0 \leq i < n$. Suppose this computation is decomposed into two concurrent processes X and Y such that X computes the array a and Y computes the array b. The processes employ two binary semaphores R and S, both initialized to zero. The array a is shared by the two processes. The structures of the processes are shown below.

Process X:

```
private i;
for (i=0; i< n; i++) {
 a[i] = f(i);
 ExitX(R, S);
}
```

Process Y:

```
private i;
for (i=0; i< n; i++) {
  EntryY(R, S);
  b[i] = g(a[i]);
}
```

Which one of the following represents the **CORRECT** implementations of ExitX and EntryY?

A.
```
ExitX(R, S) {
 P(R);
 V(S);
}
EntryY(R, S) {
 P(S);
 V(R);
}
```

B.
```
ExitX(R, S) {
```

```
 V(R);
 V(S);
}
EntryY(R, S) {
 P(R);
 P(S);
}
```

C.
```
ExitX(R, S) {
 P(S);
 V(R);
}
EntryY(R, S) {
 V(S);
 P(R);
}
```

D.
```
ExitX(R, S) {
 V(R);
 P(S);
}
EntryY(R, S) {
 V(S);
 P(R);
}
```

gate2013-cse operating-system process-synchronization normal

Answer

5.16.37 Process Synchronization: GATE CSE 2014 Set 2 | Question: 31 top

https://gateoverflow.in/1990

Consider the procedure below for the *Producer-Consumer* problem which uses semaphores:

```
semaphore n = 0;
semaphore s = 1;
```

```
void producer()
{
    while(true)
    {
        produce();
        semWait(s);
        addToBuffer();
        semSignal(s);
        semSignal(n);
    }
}
```

```
void consumer()
{
    while(true)
    {
        semWait(s);
        semWait(n);
        removeFromBuffer();
        semSignal(s);
        consume();
    }
}
```

Which one of the following is **TRUE**?

A. The producer will be able to add an item to the buffer, but the consumer can never consume it.
B. The consumer will remove no more than one item from the buffer.
C. Deadlock occurs if the consumer succeeds in acquiring semaphore s when the buffer is empty.
D. The starting value for the semaphore n must be 1 and not 0 for deadlock-free operation.

gate2014-cse-set2 operating-system process-synchronization normal

Answer

5.16.38 Process Synchronization: GATE CSE 2015 Set 1 | Question: 9 top

https://gateoverflow.in/8121

The following two functions $P1$ and $P2$ that share a variable B with an initial value of 2 execute concurrently.

P1() { C = B - 1; B = 2 * C; }	P2(){ D = 2 * B; B = D - 1; }

The number of distinct values that B can possibly take after the execution is______________________.

gate2015-cse-set1 operating-system process-synchronization normal numerical-answers

Answer

5.16.39 Process Synchronization: GATE CSE 2015 Set 3 | Question: 10 top

https://gateoverflow.in/8405

Two processes X and Y need to access a critical section. Consider the following synchronization construct used by both the processes

Process X	Process Y
/* other code for process X*/ while (true) { varP = true; while (varQ == true) { /* Critical Section */ varP = false; } } /* other code for process X */	/* other code for process Y */ while (true) { varQ = true; while (varP == true) { /* Critical Section */ varQ = false; } } /* other code for process Y */

Here *varP* and *varQ* are shared variables and both are initialized to false. Which one of the following statements is true?

A. The proposed solution prevents deadlock but fails to guarantee mutual exclusion
B. The proposed solution guarantees mutual exclusion but fails to prevent deadlock
C. The proposed solution guarantees mutual exclusion and prevents deadlock
D. The proposed solution fails to prevent deadlock and fails to guarantee mutual exclusion

gate2015-cse-set3 operating-system process-synchronization normal

Answer

5.16.40 Process Synchronization: GATE CSE 2016 Set 2 | Question: 48 top

https://gateoverflow.in/39600

Consider the following two-process synchronization solution.

PROCESS 0	Process 1
Entry: loop while (turn == 1); (critical section) Exit: turn = 1;	Entry: loop while (turn == 0); (critical section) Exit turn = 0;

The shared variable turn is initialized to zero. Which one of the following is TRUE?

A. This is a correct two- process synchronization solution.
B. This solution violates mutual exclusion requirement.
C. This solution violates progress requirement.
D. This solution violates bounded wait requirement.

gate2016-cse-set2 operating-system process-synchronization normal

Answer

5.16.41 Process Synchronization: GATE CSE 2017 Set 1 | Question: 27 top

https://gateoverflow.in/118307

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-reentrant*, 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. $x = 1, y = 1$

gate2017-cse-set1 operating-system process-synchronization normal

Answer

5.16.42 Process Synchronization: GATE CSE 2018 | Question: 40 top

https://gateoverflow.in/204114

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 $sigmal()$.

Producer:	Consumer:
do {	do {
wait (P);	wait (R);
wait (mutex);	wait (mutex);
//Add item to buffer	//consume item from buffer
signal (mutex);	signal (mutex);
signal (Q);	signal (S);
}while (1);	}while (1);

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

A. $P: full,\quad Q: full,\quad R: empty,\quad S: empty$
B. $P: empty,\quad Q: empty,\quad R: full,\quad S: full$
C. $P: full,\quad Q: empty,\quad R: empty,\quad S: full$
D. $P: empty,\quad Q: full,\quad R: full,\quad S: empty$

gate2018-cse operating-system process-synchronization normal

Answer

5.16.43 Process Synchronization: GATE CSE 2019 | Question: 23 top

https://gateoverflow.in/302825

Consider three concurrent processes P_1, P_2 and P_3 as shown below, which access a shared variable D that has been initialized to 100.

P_1	P_2	P_3
:	:	:
:	:	:
$D = D + 20$	$D = D - 50$	$D = D + 10$
:	:	:
:	:	:

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 ______

gate2019-cse numerical-answers operating-system process-synchronization

Answer

5.16.44 Process Synchronization: GATE CSE 2019 | Question: 39 top

https://gateoverflow.in/302809

Consider the following snapshot of a system running n concurrent processes. Process i is holding X_i 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 Y_i additional instances of R while holding the X_i instances it already has. Of the n processes, there are exactly two processes p and q such that $Y_p = Y_q = 0$. Which one of the following conditions guarantees that no other process apart from p and q can complete execution?

A. $X_p + X_q < \text{Min}\{Y_k \mid 1 \leq k \leq n, k \neq p, k \neq q\}$
B. $X_p + X_q < \text{Max}\{Y_k \mid 1 \leq k \leq n, k \neq p, k \neq q\}$
C. $\text{Min}(X_p, X_q) \geq \text{Min}\{Y_k \mid 1 \leq k \leq n, k \neq p, k \neq q\}$
D. $\text{Min}(X_p, X_q) \leq \text{Max}\{Y_k \mid 1 \leq k \leq n, k \neq p, k \neq q\}$

gate2019-cse operating-system process-synchronization

Answer

5.16.45 Process Synchronization: GATE IT 2004 | Question: 65 top

https://gateoverflow.in/3708

The semaphore variables full, empty and mutex are initialized to 0, n and 1, respectively. Process P$_1$ repeatedly adds one item at a time to a buffer of size n, and process P$_2$ repeatedly removes one item at a time from the same buffer using the programs given below. In the programs, K, L, M and N are unspecified statements.

P_1

```
while (1) {
    K;
    P(mutex);
    Add an item to the buffer;
    V(mutex);
    L;
}
```

P_2

```
while (1) {
    M;
    P(mutex);
    Remove an item from the buffer;
    V(mutex);
    N;
}
```

The statements K, L, M and N are respectively

A. P(full), V(empty), P(full), V(empty)
B. P(full), V(empty), P(empty), V(full)
C. P(empty), V(full), P(empty), V(full)
D. P(empty), V(full), P(full), V(empty)

gate2004-it operating-system process-synchronization normal

Answer

5.16.46 Process Synchronization: GATE IT 2005 | Question: 41 top

https://gateoverflow.in/3788

Given below is a program which when executed spawns two concurrent processes :
semaphore $X := 0$;
/* Process now forks into concurrent processes $P1$ & $P2$ */

$P1$	$P2$
repeat forever	repeat forever
$V(X)$;	$P(X)$;
Compute;	Compute;
$P(X)$;	$V(X)$;

Consider the following statements about processes $P1$ and $P2$:

I. It is possible for process $P1$ to starve.
II. It is possible for process $P2$ to starve.

Which of the following holds?

A. Both (I) and (II) are true.
B. (I) is true but (II) is false.
C. (II) is true but (I) is false
D. Both (I) and (II) are false

gate2005-it operating-system process-synchronization normal

Answer

5.16.47 Process Synchronization: GATE IT 2005 | Question: 42 top

https://gateoverflow.in/3789

Two concurrent processes $P1$ and $P2$ use four shared resources $R1, R2, R3$ and $R4$, as shown below.

P1	**P2**
Compute:	Compute;
Use $R1$;	Use $R1$;
Use $R2$;	Use $R2$;
Use $R3$;	Use $R3$;
Use $R4$;	Use $R4$;

Both processes are started at the same time, and each resource can be accessed by only one process at a time The following scheduling constraints exist between the access of resources by the processes:

- $P2$ must complete use of $R1$ before $P1$ gets access to $R1$.
- $P1$ must complete use of $R2$ before $P2$ gets access to $R2$.
- $P2$ must complete use of $R3$ before $P1$ gets access to $R3$.
- $P1$ must complete use of $R4$ before $P2$ gets access to $R4$.

There are no other scheduling constraints between the processes. If only binary semaphores are used to enforce the above scheduling constraints, what is the minimum number of binary semaphores needed?

A. 1
B. 2
C. 3
D. 4

gate2005-it operating-system process-synchronization normal

Answer

5.16.48 Process Synchronization: GATE IT 2006 | Question: 55 top

https://gateoverflow.in/3598

Consider the solution to the bounded buffer producer/consumer problem by using general semaphores $S, F,$ and E. The semaphore S is the mutual exclusion semaphore initialized to 1. The semaphore F corresponds to the number of free slots in the buffer and is initialized to N. The semaphore E corresponds to the number of elements in the buffer and is initialized to 0.

Producer Process	**Consumer Process**
Produce an item;	Wait(E);
Wait(F);	Wait(S);
Wait(S);	Remove an item from the buffer;
Append the item to the buffer;	Signal(S);
Signal(S);	Signal(F);
Signal(E);	Consume the item;

Which of the following interchange operations may result in a deadlock?

I. Interchanging Wait (F) and Wait (S) in the Producer process
II. Interchanging Signal (S) and Signal (F) in the Consumer process

A. (I) only
B. (II) only
C. Neither (I) nor (II)
D. Both (I) and (II)

gate2006-it operating-system process-synchronization normal

Answer

5.16.49 Process Synchronization: GATE IT 2007 | Question: 10 top

https://gateoverflow.in/3443

Processes $P1$ and $P2$ use critical_flag in the following routine to achieve mutual exclusion. Assume that critical_flag is initialized to FALSE in the main program.

```
get_exclusive_access ( )
{
    if (critical _flag == FALSE) {
        critical_flag = TRUE ;
        critical_region () ;
        critical_flag = FALSE;
    }
}
```

Consider the following statements.

i. It is possible for both $P1$ and $P2$ to access critical_region concurrently.
ii. This may lead to a deadlock.

Which of the following holds?

A. (i) is false (ii) is true
B. Both (i) and (ii) are false
C. (i) is true (ii) is false
D. Both (i) and (ii) are true

gate2007-it operating-system process-synchronization normal

Answer

5.16.50 Process Synchronization: GATE IT 2007 | Question: 56 top

https://gateoverflow.in/3498

Synchronization in the classical readers and writers problem can be achieved through use of semaphores. In the following incomplete code for readers-writers problem, two binary semaphores mutex and wrt are used to obtain synchronization

```
wait (wrt)
writing is performed
signal (wrt)
wait (mutex)
readcount = readcount + 1
if readcount = 1 then S1
S2
reading is performed
S3
readcount = readcount - 1
if readcount = 0 then S4
signal (mutex)
```

The values of $S1, S2, S3, S4,$ (in that order) are

A. signal (mutex), wait (wrt), signal (wrt), wait (mutex)
B. signal (wrt), signal (mutex), wait (mutex), wait (wrt)
C. wait (wrt), signal (mutex), wait (mutex), signal (wrt)
D. signal (mutex), wait (mutex), signal (mutex), wait (mutex)

gate2007-it operating-system process-synchronization normal

Answer

5.16.51 Process Synchronization: GATE IT 2008 | Question: 53 top

https://gateoverflow.in/3363

The following is a code with two threads, producer and consumer, that can run in parallel. Further, S and Q are binary semaphores quipped with the standard P and V operations.

```
semaphore S = 1, Q = 0;
integer x;

producer:                    consumer:
while (true) do              while (true) do
    P(S);                        P(Q);
    x = produce ();              consume (x);
    V(Q);                        V(S);
done                         done
```

Which of the following is TRUE about the program above?

A. The process can deadlock
B. One of the threads can starve
C. Some of the items produced by the producer may be lost
D. Values generated and stored in 'x' by the producer will always be consumed before the producer can generate a new value

gate2008-it operating-system process-synchronization normal

Answer

Answers: Process Synchronization

5.16.1 Process Synchronization: GATE CSE 1987 | Question: 1-xvi top https://gateoverflow.in/80362

✔ A critical region is a program segment where shared resources are accessed, that's why we synchronize in the critical section.

PS: It is not necessary that we must use semaphore for critical section access (any other mechanism for mutual exclusion can also be used) and neither do sections enclosed by P and V operations are called critical sections.

Correct Answer : D.

34 votes -- kirti singh (2.6k points)

5.16.2 Process Synchronization: GATE CSE 1987 | Question: 8a top https://gateoverflow.in/82433

12 readers are reading means each reader has incremented the value of **ar**, making final value of **ar** to be 12.

Also each of the reader has executed ***grantread*** in which **rr** is incremented to the value of **ar** making value of **rr** to be 12 finally.

31 writers are waiting means each writer on arrival has incremented the value of **aw**, making final value of **aw** to be 31.

Value of **rw** is incremented in ***grantwrite*** only when value of **rr** is 0 but as 12 readers are already reading, this cannot happen, making value of **rw** to be 0.

Whenever read is granted in ***grantread***, it means value of **reading** semaphore is incremented to number of reader process using ***V(reading)***. But before entering the read section, each reader decrements the **reading** semaphore by 1 using ***P(reading)***. The fact that 12 readers are reading means that 12 ***V(reading)*** operations were performed and the 12 reader processes before entering read section have performed ***P(reading)*** each to decrement the value of **reading** semaphore to 0 again.

Since 12 readers are already reading, value of **rr** is non-zero because of which ***V(writing)*** is not executed leaving the value of **writing** semaphore to be 0.

NO, group of readers will not starve writers as readers execute ***V(reading)*** in ***grantread*** only when **aw** is 0 i.e. no writer is waiting allowing writer to execute first.

YES, writers can starve readers as writers execute ***V(writing)*** without caring about readers (**ar**).

The solution is **incorrect** because:

In reader-writer problem, only single process needs to write at a time.

But in proposed solution, consider the case: When one process is writing and another write process arrives then it is also granted write using ***V(writing)*** without caring about the first process which is still writing.

6 votes -- Pratik Gawali (897 points)

5.16.3 Process Synchronization: GATE CSE 1988 | Question: 10iib top https://gateoverflow.in/94393

✔ the above solution for the critical section isn't correct because it **satisfies Mutual exclusion and Progress** but it **violates the bounded waiting**.

Here is a sample run

```
suppose turn =j initially;
```

Pi runs its first statement then Pj runs its first statement then P_i run $2, 3, 4$ statement, It will block on statement 4

Now P_j start executing its statements goes to critical section and then flag[j] = false

Now suppose P_j comes again immediately after execution then it will again execute its critical section and then flag[j] = false

Now if P_j is coming continuously then process P_i will suffer starvation.

---

the correct implementation (for Bounded waiting) is, at the exit section we have to update the turn variable at the exit section.

```
repeat
    flag[i]:= true;
    while turn != i
    do begin
        while flag [j] do skip
        turn:=i;
    end
    critical section
    flag[i]:=false;
    turn=j;
until false
```

10 votes

-- Aakashpatel (153 points)

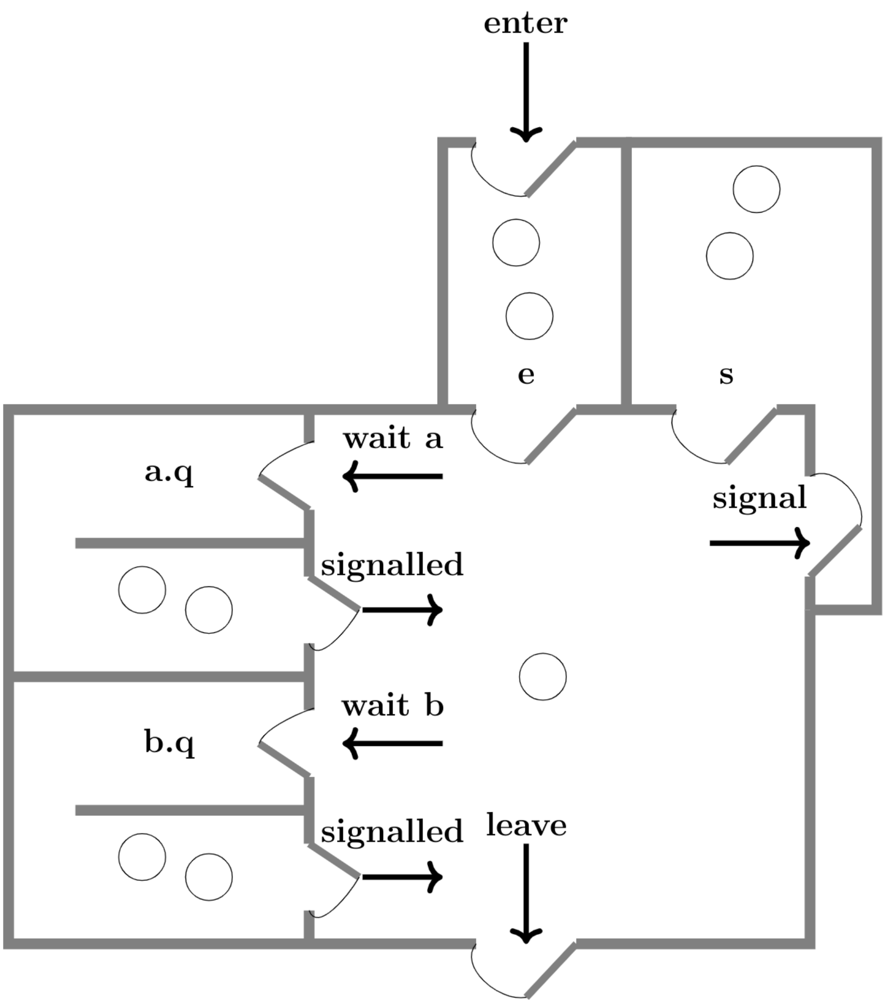
5.16.4 Process Synchronization: GATE CSE 1990 | Question: 2-iii top

https://gateoverflow.in/83859

✔ ***A. Circular Wait*** is one of the conditions for **deadlock.**

B. To avoid race conditions, the execution of **critical sections** must be **mutually exclusive** (e.g., at most one process can be in its critical section at any time).

C. Monitors using blocking condition variables are often called **Hoare-style monitors** or **signal-and-urgent-wait monitors**.



D. locality is commonly used to determine the number of assigned pages. The number of pages that meet the requirement of locality is called a **working set.**

(a)	Critical region	(q)	Mutual exclusion
(b)	Wait/Signal	(p)	Hoare's monitor
(c)	Working Set	(r)	Principle of locality
(d)	Deadlock	(s)	Circular Wait

22 votes

-- Pankaj Kumar (7.8k points)

5.16.5 Process Synchronization: GATE CSE 1991 | Question: 11,a top

https://gateoverflow.in/538

***Pre-requisite**: Assume all 3 processes have same implementation of code except flag variable indices changes accordingly for Pj and Pk and turn is shared variable among 3 process.*

The condition:

```
while flag [j] or flag[k] do
```

ensures **mutual exclusion** as no process can enter critical section until flag of other processes is false.

Consider the case: turn = k

Pj wants to enter the critical section. It enters the critical section easily as

```
flag [k] or flag[i]
```

will be false and the loop will break.

Now, while **Pj** is executing in its critical section **Pi** arrives. For **Pi** :

```
flag [j] or flag[k]
```

will be true and it will enter the while loop. Since, turn = k, **Pi** will execute the loop:

```
while turn != i do skip;
```

Now, even if **Pj** finishes executing it's critical section, it will execute:

```
if turn = j then turn := k;
```

which is false and thus the turn will remain k making **Pi** to execute an infinite loop until **Pk** arrives which can update turn = i.

So if **Pk** never arrives **Pi** will be waiting indefinitely.

8 votes

-- Pratik Gawali (897 points)

5.16.6 Process Synchronization: GATE CSE 1991 | Question: 11,b top

https://gateoverflow.in/43000

“the process which use critical section should hold turn variable otherwise other waiting process will wait for indefinite time if some process does not wants to enter in cs“

1. progress not satisfied
   2. no deadlock

4 votes

-- indra kumar sahu (203 points)

5.16.7 Process Synchronization: GATE CSE 1993 | Question: 22 top

https://gateoverflow.in/2319

✔

```
parbegin
    begin  S1  parbegin   V(a)   V(b)   parend  end
```

```
    begin   P(a)    S2  parbegin    V(c)    V(e)    parend  end
    begin   P(b)    S3  V(d)    end
    begin   P(f)    P(c)    S4  end
    begin   P(g)    P(d)    P(e)    S5  end
    begin   S6  parbegin    V(f)    V(g)    parend  end
parend
```

Here, the statement between parbegin and parend can execute in any order. But the precedence graph shows the order in which the statements should be executed. This strict ordering is achieved using the semaphores.

Initially all the semaphores are 0.

For S_1 there is no need of semaphore because it is the first one to execute.

Next S_2 can execute only when S_1 finishes. For this we have a semaphore a which on signal executed by S_1, gets value 1. Now S_2 which is doing a wait on a can continue execution making $a = 0$;

Likewise this is followed for all other statements.

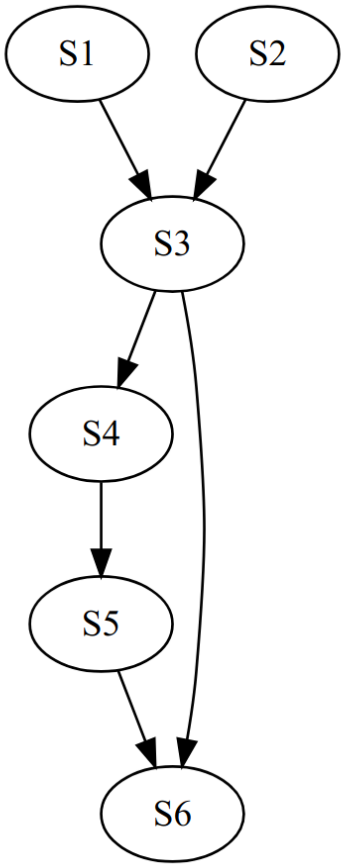
26 votes

-- Sourav Roy (2.9k points)

5.16.8 Process Synchronization: GATE CSE 1994 | Question: 27 top

https://gateoverflow.in/2523

✔ Following must be the correct precedence graph, S_1 abd S_2 are independent, hence these can be executed in parallel.



For all those nodes which are independent we can execute them in parallel by creating a separate process for each node like S_1 and S_2. There is an edge from S_3 to S_6 it means, until the process which executes S_3 finishes its work, we can't start the process which will execute S_6.

For more understanding watch the following NPTEL lectures on process management:

Video:

Video:

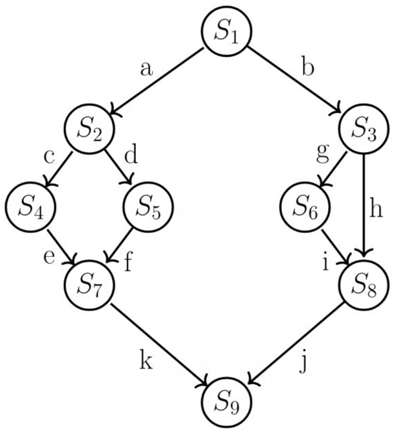
19 votes

-- Manu Thakur (34k points)

5.16.9 Process Synchronization: GATE CSE 1995 | Question: 19 top

https://gateoverflow.in/2656

✔ Precedence graph will be formed as:



46 votes

-- neha pawar (3.3k points)

5.16.10 Process Synchronization: GATE CSE 1996 | Question: 1.19, ISRO2008-61 top

https://gateoverflow.in/2723

✔

A. There is no time guarantee for critical section.

B. Critical section by default doesn't avoid deadlock. While using critical section, programmer must ensure deadlock is avoided.

C. is the answer
http://en.wikipedia.org/wiki/Critical_section

D. This is not a requirement of critical section. Only when semaphore is used for critical section management, this becomes a necessity. But, semaphore is just ONE of the ways for managing critical section.

References

32 votes

-- Gate Keeda (15.9k points)

5.16.11 Process Synchronization: GATE CSE 1996 | Question: 2.19 top

https://gateoverflow.in/2748

✔ Acc. to me it should be (**C**) because: according to condition, out of all, one philosopher will get both the forks. So, deadlock should not be there.

32 votes

-- Sneha Goel (819 points)

5.16.12 Process Synchronization: GATE CSE 1996 | Question: 21 top

https://gateoverflow.in/2773

✔ $S1, S2, S3, S4$ and $S5$ are the statements to be executed.

Fork() creates a child to execute in parallel.

There will be 3 processes running concurrently.
One will execute S1, 2^{nd} will execute $S2$ and 3^{rd} will execute $S4$.

Initially there is one process which started execution. Suppose this process name is P_0.

It executes $N = 2, M = 2$ and after that it executes

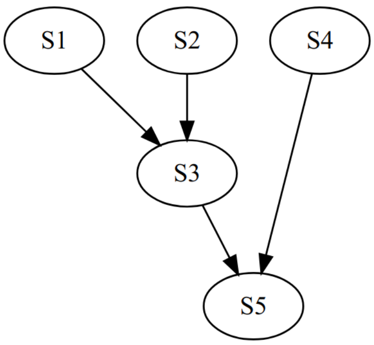
Fork: $L3$,
At $L3$ these is a statement $S2$, Fork creates a new process suppose P_1 which starts its execution from level $L3$, means it starts executing $S2$.

P_0 executes fork $L4$, it creates another new process P_2 which starts its execution from level $L4$ means it starts executing $S4$.

When P_1 finishes executing $S2$, it executes next line which is goto $L1$.
When P_2 finishes executing $S4$, it executes next line which is goto $L2$.

$L1$ is executed by both processes $P_0($ which has executed S1$)$ and $P_1($ which has executed S2$)$
Hence, S1 and S2 are combined together, as either P_0 or P_1 will terminate $(\because N = 2)$ and only one process will continue its execution.

Similarly $L2$ is executed by two processes $P_2($ which executed S4) and one of P_0 or $P_1($ which executed S3$)$. So, $S4$ and S3 are joined together, as one of them will terminate $(\because M = 2)$ and then one which survives will execute the final statement $S5$.



www.csc.lsu.edu/~rkannan/**Fork**_Cobegin_Creationtime.docx
http://www.cis.temple.edu/~giorgio/old/cis307s96/readings/precedence.html

References

👍 37 votes

-- Manu Thakur (34k points)

5.16.13 Process Synchronization: GATE CSE 1997 | Question: 6.8 top

https://gateoverflow.in/2264

✔ Answer is (**D**).
If initial value is 1//execute P_1 or P_{10} first
If initial value is 0, P_{10} can execute and make the value 1.
Since the both code (i.e. P_1 to P_9 and P_{10}) can be executed any number of times and code for P_{10} is

```
repeat
{
    V(mutex)
    C.S.
    V(mutex)
}
forever
```

Now, let me say P_1 is in Critical Section (CS)
then P_{10} comes executes the CS (up on mutex)
now P_2 comes (down on mutex)
now P_{10} moves out of CS (again binary semaphore will be 1)

now P_3 comes (down on mutex)
now P_{10} come (up on mutex)
now P_4 comes (down on mutex)
So, if we take P_{10} out of CS recursively all 10 process can be in CS at same time using Binary semaphore only.

👍 67 votes

-- Kalpish Singhal (1.6k points)

5.16.14 Process Synchronization: GATE CSE 1997 | Question: 73 top

☛ https://gateoverflow.in/19703

✔

```
procedure release_R(priority)
begin
    P(mutex); //only one process must be executing the following part at a time
    R_requested[priority] = false; //this process has requested,
//allocated the resource and now finished using it
    for (i = 3 downto 1)//starting from highest priority process
    begin
        if R_requested[i] then
        begin
            V(proceed[i]);//Process i is now given access to resource
            break;
        end
    end
    if (!R_requested[0] && !R_requested[1] && !R_requested[2]) then
        busy = false;//no process is waiting and so next incoming resource
        //can be served without any wait
    V(mutex); //any other process can now request/release resource
end
```

👍 5 votes

-- Arjun Suresh (332k points)

5.16.15 Process Synchronization: GATE CSE 1998 | Question: 1.30 top

☛ https://gateoverflow.in/1667

✔ When final result depends on ordering of processes it is called [Race condition](#).
Speed of processes corresponds to ordering of processes.

References

👍 50 votes

-- Digvijay (44.9k points)

5.16.16 Process Synchronization: GATE CSE 1999 | Question: 20-a top

☛ https://gateoverflow.in/1519

✔

1. TSET $R1$, flag
2. CMP $R1$, #0
3. JNZ Step1
4. [CS]
5. Store $M[Flag], \#0

👍 16 votes

-- Manu Thakur (34k points)

5.16.17 Process Synchronization: GATE CSE 1999 | Question: 20-b top

☛ https://gateoverflow.in/205817

✔ 1. **Run the Consumer Process**, Test the condition inside "if" (It is given that the testing of count is atomic operation), and since the Count value is initially 0, condition becomes True. **After Testing (But BEFORE "Sleep" executes in consumer process), Preempt the Consumer Process.**

2. **Now Run Producer Process completely** (All statements of Producer process). (Note that in Producer Process, 5th line of code, "Wakeup(Consumer);" will not cause anything because Consumer Process hasn't Slept yet (We had Preempted Consumer process before It could go to sleep). Now at the end of One pass of Producer process, Count value is now 1. So, Now if we again run Producer Process, "if" condition becomes true and **Producer Process goes to sleep.**

3. **Now run the Preempted Consumer process, And It also Goes to Sleep. (Because it executes the Sleep code).**

So, Now Both Processes are sleeping at the same time.

👍 23 votes -- Deepak Poonia (23.4k points)

5.16.18 Process Synchronization: GATE CSE 2000 | Question: 1.21 top https://gateoverflow.in/645

✔ $P0: m[0]; m[1]$

$P1: m[1]; m[2]$

$P2: m[2]; m[3]$

$P3: m[3]; m[0]$

$P4: m[4]; m[1]$

po holding mo waiting for $m1$
$p1$ holding $m1$ waiting for $m2$
$p2$ holding $m2$ waiting for $m3$
$p3$ holding $m3$ waiting for $m0$
$p4$ holding $m4$ waiting for $m1$
So its circular wait and no process can go into critical section even thought its free hence
Answer: (**B**) Deadlock.

👍 65 votes -- Sourav Roy (2.9k points)

5.16.19 Process Synchronization: GATE CSE 2000 | Question: 20 top https://gateoverflow.in/691

✔ There are four conditions that must be satisfied by any reader-writer problem solution

1. When a reader is reading, no writer must be allowed.
2. Multiple readers should be allowed.
3. When a writer is writing, no reader must be allowed.
4. Multiple writers(more than 1) should not be allowed.

Now, here mutex is a semaphore variable that will be used to modify the variables R and W in a mutually exclusive way.

The reader code should be like below

```
Reader()
L1: wait(mutex);
if(w==0){ //no Writer present, so allow Readers to come.

R=R+1; //increment the number of readers presently reading by 1.

signal(mutex);//Reader is allowed to enter,
            //number of readers present "R"
            //is incremented and now make mutex available so that other readers
            //can come.
}
else{  //means some writer is writing,so release mutex, and try to
      //gain access to mutex again by looping back to L1.

signal(mutex);
goto L1;
}
/*reading performed*/
wait(mutex);
R=R-1;
signal(mutex);
```

Value of variable R indicates the number of readers presently reading and the value of W indicates if 1, that some writer is present.

Writer code should be like below

```
Writer()
L2: wait(mutex);
if(R>0 || W!=0) //means if even one reader is present or one writer is writing
                //deny access to this writer process and ask this to release
                //mutex and loop back to L2.
{
 signal(mutex);
 goto L2;
}
//code will come here only if no writer or no reader was present.

W=1; //indicate that a writer has come.

signal(mutex); //now after updating W safely, release mutex, for other writers and
         //readers to place their request.

/*Write performed*/
//writer will leave so change Value of W in a mutual exclusive manner.
wait(mutex);
W=0;
signal(mutex);
```

This will satisfy all requirements of the solution to the reader-writer problem.

(B) Yes, writers can starve. There can be the scenario that whenever a writer tries to enter, it finds some reader $(R! = 0)$, or another writer process $(W! = 0)$ and it can keep waiting forever. Bounded Waiting for the writer's processes is not ensured.

77 votes -- Ayush Upadhyaya (28.4k points)

5.16.20 Process Synchronization: GATE CSE 2001 | Question: 2.22 top https://gateoverflow.in/740

✔ Answer is Option B as used in Peterson's solution for Two Process Critical Section Problem which guarantees

1. Mutual Exclusion
2. Progress
3. Bounded Waiting

Both i and j are concurrent processes. So, whichever process wants to enter critical section(CS) that will execute the given code.

A process i shows it's interest to enter CS by setting $flag[i]$ = TRUE and only when i leaves CS it sets $flag[i]$= FALSE.

From this it's clear that when some process wants to enter CS then it must check value of $flag[]$ of the other process.

$\therefore$ " $flag[j]$ == TRUE " must be one condition that must be checked by process i.

Here, the $turn$ variable specifies whose turn is next i.e. which process can enter the CS next time. "$turn$ " acts like an unbiased scheduler, it ensures giving fair chance to the processes for execution. When a process sets $flag[]$ value, then $turn$ value is set equal to other process so that same process is not executed again (strict alteration when both processes are ready). i.e., usage of turn variable here ensures "Bounded Waiting" property.

Before entering CS every process needs to check whether other process has shown interest first and which process is scheduled by the $turn$ variable. If other process is not ready, $flag[other]$ will be false and the current process can enter the CS irrespective of the value of $turn$. Thus, the usage of $flag$ variable ensures "Progress" property.

If $flag[other]$ = TRUE and $turn$ = other, then the process has to wait until one of the conditions becomes false. (because it is the turn of other process to enter CS). This ensures Mutual Exclusion.

Thus, ans is (b).

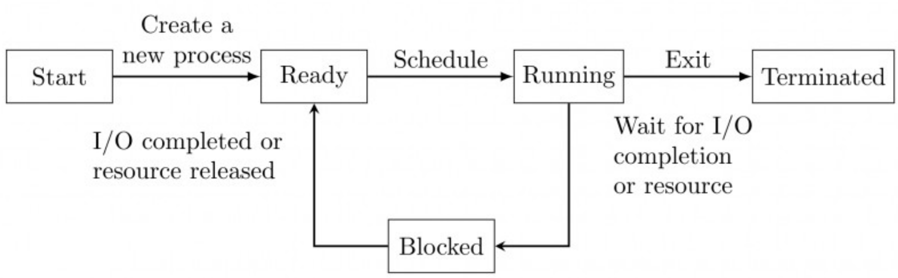
** one interesting point that can be observed is, if 2 processes wants to enter the CS, the process which executes " $turn = j$" statement first is always the first one to enter the CS (after the other process executes $turn = j$".

35 votes -- SameekshaGupta (787 points)

5.16.21 Process Synchronization: GATE CSE 2002 | Question: 18-a top https://gateoverflow.in/871

✔ Process state transition diagram for an OS which satisfy the below two criteria will be as follows:

i. each process is in one of the five states: created, ready, running, blocked (i.e., sleep or wait), or terminated, and
ii. only non-preemptive scheduling is used by the OS.



If in question it is asked about the preemptive scheduling then after the `running state` a process directly go to `ready state`.

15 votes -- Shubhgupta (6.5k points)

Solution will be

```
void enter-cs()
{
    while(TestAndSet(&mutex));
}
void leave-cs()
{
    mutex=0;
}
```

Here there are two possible cases

Case (I) Test And Set is **not ATOMIC**: Consider a scenario where first, a process P1 comes, successfully executed enter-cs() and sets mutex to 1. Now, suppose another process P2 comes and executes while(TestAndSet(&mutex)) to gain access to CS. Suppose after executing line

y=*x=1(mutex is one presently)

P2 gets preempted.

Now P1, resumes and sets mutex to 0.

Now P2 resumes, and executes remaining lines of TestAndSet,

*x=1 (mutex is assigned value 1, value 0 lost permanently)

return y; //1 will be returned.

Now, P2 and any other process which tries to execute enter-cs() will keep looping indefinitely. Now not even process P1 can come.

So, deadlock will occur eventually.

And Deadlock implies starvation so starvation is also there. (But starvation does not imply deadlock).

Yes, in this case, the mutual exclusion will not hold.

Suppose initially mutex=0.

Process P1 comes and executes the first iteration of while loop of enter-cs-->TestAndSet(&mutex)

y=*x; //0 will be stored in y.

Suppose after executing this line of TestAndSet, P1 gets preempted and process P2 comes.

It also executes while loop of enter-cs() and executes TestAndSet(&mutex)

y=*x; //0 will be stored because mutex was not changed to 1.

*x=1; //mutex changed to 1.

return y. //0 will be returned.

Now P2 exits while loop and gains entry into CS.

Say, P1 resumes, and it executes the remaining line of TestAndSet and it returns 0.

P1, also exits while loop of enter-cs() and comes into CS.

Mutual exclusion is broken!!.

Case (II): TestAndSet is ATOMIC: Mutual exclusion will hold, because the first process to execute TestAndSet when mutex will be 0, will enter CS and rest all other processes will keep looping in the while loop of enter-cs().

Deadlock will not occur, because all other processes, which are looping in the while loop, will do so untill mutex!=0. When the process which is in the CS leaves, it sets mutex=0, and one of the waiting processes which successfully finds mutex=0 AND executes the TestAndSet when mutex=0, will gain access to CS.

Starvation will occur, because as you can see in the code, no piece of code can be seen which is responsible for providing access to waiting processes in a fair shared manner. It might happen that one process always finds mutex to be 1, while rest all other processes are able to enter and leave CS, one at a time.

If some code is added to the leave-section(), which ensures that the waiting processes are given chance to enter CS in the order the request was placed, then starvation won't occur. In short, **here BOUNDED WAITING is not ensured.** If BOUNDED WAITING is ensured, STARVATION will not occur.

14 votes

-- Ayush Upadhyaya (28.4k points)

✔ a) In Producer Consumer problem Producer produce item and makes the buffer full and after that Consumer consumes that item and makes the buffer empty

Here b_empty and b_full are two semaphore values

```
p1: P(Empty)
```

means, Producer have to wait only if buffer is full and it waits for consumer to remove at least one item. (See, Empty being initialized to BUFFSIZE)

```
p2: V(Full)
```

buffer is filled, now it gives signal to consumer that it can start consuming

```
c1: P(Full)
```

means here consumer have to wait only if buffer is empty, and it waits for Producer to fill the buffer

```
c2: V(Empty)
```

Now buffer is empty and Empty semaphore gives signal to the producer that it can start filling

It is same as giving water to a thirsty man.

Here u are giving water in a glass to that thirsty man, so u are producer here

and the man drinks and makes the glass empty, so he is consumer here

b) If there are multiple user we can use mutex semaphore, so that exclusively one could enter in Critical section at a time. i.e.

```
p1:P(Empty)
   P(mutex)

p2:V(mutex)
     V(Full)

c1:P(Full)
    P(mutex)

c2: V(mutex)
     V(Empty)
```

PS: One thing to see is P(mutex) is after P(Full) and P(empty)- otherwise deadlock can happen when buffer is full and a producer gets mutex or if buffer is empty and a consumer gets mutex.

27 votes -- srestha (85.2k points)

5.16.24 Process Synchronization: GATE CSE 2003 | Question: 80 top https://gateoverflow.in/964

✔ To get pattern 001100110011

Process P should be executed first followed by **Process Q.**

So, at Process P : **W** $P(S)$ **X** $V(T)$

And at Process Q : **Y** $P(T)$ **Z** $V(S)$

With $\mathbf{S=1}$ **and** $\mathbf{T=0}$ initially (only $\mathbf{P}$ has to be run first then only $\mathbf{Q}$ is run. Both processes run on alternate way starting with $\mathbf{P}$)

So, answer is **(B).**

30 votes -- Pooja Palod (24.1k points)

5.16.25 Process Synchronization: GATE CSE 2003 | Question: 81 top https://gateoverflow.in/43574

✔ output shouldn't contain substring of given form means no concurrent execution process P as well as Q. one semaphore is enough

So ans is **(C)**

43 votes -- Pooja Palod (24.1k points)

5.16.26 Process Synchronization: GATE CSE 2004 | Question: 48 top https://gateoverflow.in/1044

✔

A. deadlock $p1$: line1|$p2$:line3| $p1$: line2(block) |$p2$:line4(block)

So, here $p1$ want $s(x)$ which is held by $p2$ and $p2$ want $s(y)$ which is held by $p1$.
So, its **circular wait (hold and wait condition)**. So. there is **deadlock.**

B. **deadlock** $p1$: line 1| $p2$ line 3|$p1$: line 2(block) |$p2$: line 4(block)
Som here $p1$ wants sy which is held by $p2$ and $p2$ wants sx which is held by $p1$. So its **circular wait (hold and wait) so, deadlock.**

C. $p1$:line 1|$p2$: line 3|$p2$ line 4 (block) |$p1$ line 2 (block) here, $p1$ wants sx and $p2$ wants sy, but both will not be release by its process $p1$ and $p2$ because there is no way to release them. So, stuck in **deadlock**.

D. $p1$:line 1 |$p2$: line 3 (block because need sx) |$p1$ line 2|$p2$: still block |$p1$: execute cs then up the value of sx |p2 :line 3 line 4(block need sy)|$p1$ up thesy |$p2$:lin4 4 and easily get cs.
We can start from $p2$ also, as I answered according only $p1$, but we get same answer.
So, **option (D)** is correct

37 votes

-- minal (13.1k points)

5.16.27 Process Synchronization: GATE CSE 2006 | Question: 61 top https://gateoverflow.in/1839

✔

A. **Answer** :- This is correct because the implementation may not work if context switching is disabled in P , then process which is currently blocked may never give control to the process which might eventually execute V. So Context switching is must !

B. If we use normal load & Store instead of Fetch & Set there is good chance that more than one Process sees S.value as 0 & then mutual exclusion wont be satisfied. So this option is wrong.

C. Here we are setting $S \to$ value to 0, which is correct. (As in fetch & Set we wait if value of $S \to$ value is 1. So implementation is correct. This option is wrong.

D. I don't see why this code does not implement binary semaphore, only one Process can be in critical section here at a time. So this is binary semaphore & Option (D) is wrong

90 votes

-- Akash Kanase (36k points)

5.16.28 Process Synchronization: GATE CSE 2006 | Question: 78 top https://gateoverflow.in/1853

✔ (**B**) is the correct answer.

Let 3 processes p_1, p_2, p_3 arrive at the barrier and after 4^{th} step *process_arrived=3* and the processes enter the barrier. Now suppose process p_1 executes the complete code and makes *process_left=1*, and tries to re-enter the barrier. Now, when it executes 4^{th} step, *process_arrived=4*. p_1 is now stuck. At this point all other processes p_2 and p_3 also execute their section of code and resets *process_arrived=0* and *process_left=0*. Now, p_2 and p_3 also try to re-enter the barrier making *process_arrived=2*. At this point all processes have arrived, but *process_arrived!=3*. Hence, no process can re-enter into the barrier, therefore DEADLOCK!!

122 votes

-- GateMaster Prime (1.2k points)

5.16.29 Process Synchronization: GATE CSE 2006 | Question: 79 top https://gateoverflow.in/43564

✔ The implementation is incorrect because if two barrier invocations are used in immediate succession the system will fall into a DEADLOCK.

Here's how: Let all three processes make process_arrived variable to the value 3, as soon as it becomes 3 previously stuck processes at the while loop are now free, to move out of the while loop.

But for instance let say one process moves out and has bypassed the next if statement & moves out of the barrier function and The SAME process is invoked again(its second invocation) while other processes are preempted still.

That process on its second invocation makes the process_arrived variable to 4 and gets stuck forever in the while loop with other processes.

At this point of time they are in DEADLOCK. as only 3 processes were in the system and all are now stuck in while loop.

Q.79 answer = **option (B)**

option (A) here is false as there will always be a need for some process to help some other process to move out of that while loop waiting. Not all processes together can be said to be completed at a time.

option (C) is false. If context switch is disabled then the process who was stuck in while loop will remain there forever and no other process can play a role in bringing it out of there as Context Switch will be required to bring that other process in the system to do the job.

option (D) is false. everyone will be in a loop forever, if that happens.

option (B) is TRUE. at the beginning of the barrier the 1^{st} process to enter Critical section should wait until process_arrived becomes zero(i.e. before starting its second invocation). this is to prevent it from making process_arrived value greater than 3 i.e. rectifying the flaw observed in Q.78

41 votes -- Amar Vashishth (25.2k points)

5.16.30 Process Synchronization: GATE CSE 2007 | Question: 58 top https://gateoverflow.in/1256

✔ $P1$ can do wants1 $=$ true and then $P2$ can do wants2 $=$ true. Now, both $P1$ and $P2$ will be waiting in the while loop indefinitely without any progress of the system - deadlock.

When $P1$ is entering critical section it is guaranteed that wants1 $=$ true (wants2 can be either true or false). So, this ensures $P2$ won't be entering the critical section at the same time. In the same way, when $P2$ is in critical section, $P1$ won't be able to enter critical section. So, mutual exclusion condition satisfied.

So, **D** is the correct choice.

Suppose $P1$ first enters critical section. Now suppose $P2$ comes and waits for CS by making wants2 $=$ true. Now, $P1$ cannot get access to CS before $P2$ gets and similarly if $P1$ is in wait, $P2$ cannot continue more than once getting access to CS. Thus, there is a bound (of 1) on the number of times another process gets access to CS after a process requests access to it and hence bounded waiting condition is satisfied.

---

https://cs.stackexchange.com/questions/63730/how-to-satisfy-bounded-waiting-in-case-of-deadlock

References

69 votes -- Arjun Suresh (332k points)

5.16.31 Process Synchronization: GATE CSE 2009 | Question: 33 top https://gateoverflow.in/1319

✔ The answer is (**A**) only.

The solution satisfies:

1. Mutual Exclusion as test-and-set is an indivisible (atomic) instruction (makes option (IV) wrong)
2. Progress as at initially X is 0 and at least one process can enter critical section at any time.

But no guarantee that a process eventually will enter CS and hence option (IV) is false. Also, no ordering of processes is maintained and hence III is also false.

So, eliminating all the 3 choices remains A.

39 votes -- Gate Keeda (15.9k points)

5.16.32 Process Synchronization: GATE CSE 2010 | Question: 23 top https://gateoverflow.in/2202

✔ Answer is (**A**). In this mutual exclusion is satisfied,only one process can access the critical section at particular time but here progress will not satisfied because suppose when $s1 = 1$ and $s2 = 0$ and process p1 is not interested to enter into critical section but $p2$ want to enter critical section. $P2$ is not able to enter critical section in this as only when $p1$ finishes execution, then only $p2$ can enter (then only $s1 = s2$ condition be satisfied).

Progress will not be satisfied when any process which is not interested to enter into the critical section will not allow other interested process to enter into the critical section. When P1 wants to enter the critical section it might need to wait till P2 enters and leaves the critical section (or vice verse) which might never happen and hence progress condition is violated.

77 votes -- neha pawar (3.3k points)

5.16.33 Process Synchronization: GATE CSE 2010 | Question: 45 top

https://gateoverflow.in/2347

First P_0 will enter the while loop as S_0 is 1. Now, it releases both S_1 and S_2 and one of them must execute next. Let that be P_1. Now, P_0 will be waiting for P_1 to finish. But in the mean time P_2 can also start execution. So, there is a chance that before P_0 enters the second iteration both P_1 and P_2 would have done release (S_0) which would make S_1 1 only (as it is a binary semaphore). So, P_0 can do only one more iteration printing $'0'$ two times.

If P_2 does release (S_0) only after P_0 starts its second iteration, then P_0 would do three iterations printing $'0'$ three times.

If the semaphore had 3 values possible (an integer semaphore and not a binary one), exactly three $'0's$ would have been printed.

Correct Answer: A, at least twice

49 votes -- Arjun Suresh (332k points)

5.16.34 Process Synchronization: GATE CSE 2012 | Question: 32 top

https://gateoverflow.in/1750

A process acquires a lock only when $L = 0$. When L is 1, the process repeats in the while loop- there is no overflow because after each increment to L, L is again made equal to 1. So, the only chance of overflow is if a large number of processes (larger than sizeof(int)) execute the check condition of while loop but not $L = 1$, which is highly improbable.

Acquire Lock gets success only when Fetch_And_Add gets executed with L = 0. Now suppose $P1$ acquires lock and make $L = 1$. $P2$ waits for a lock iterating the value of L between 1 and 2 (assume no other process waiting for lock). Suppose when $P1$ releases lock by making $L = 0$, the next statement $P2$ executes is $L = 1$. So, value of L becomes 1 and no process is in critical section ensuring L can never be 0 again. Thus, (**B**) choice.

To correct the implementation we have to replace Fetch_And_Add with Fetch_And_Make_Equal_1 and remove $L = 1$ in $AcquireLock(L)$.

163 votes -- Arjun Suresh (332k points)

5.16.35 Process Synchronization: GATE CSE 2013 | Question: 34 top

https://gateoverflow.in/1545

Since, initial value of semaphore is 2, two processes can enter critical section at a time- this is bad and we can see why.

Say, X and Y be the processes. X increments x by 1 and Z decrements x by 2. Now, Z stores back and after this X stores back. So, final value of x is 1 and not -1 and two Signal operations make the semaphore value 2 again. So, now W and Z can also execute like this and the value of x can be **2 which is the maximum possible** in any order of execution of the processes.

(If the semaphore is initialized to 1, processed would execute correctly and we get the final value of x as -2.)

91 votes -- Arjun Suresh (332k points)

5.16.36 Process Synchronization: GATE CSE 2013 | Question: 39 top

https://gateoverflow.in/1550

A. X is waiting on R and Y is waiting on X. So, both cannot proceed.

B. Process X is doing Signal operation on R and S without any wait and hence multiple signal operations can happen on the binary semaphore so Process Y won't be able to get exactly n successful wait operations. i.e., Process Y may not be able to complete all the iterations.

C. Process X does Wait(S) followed by Signal(R) while Process Y does Signal(S) followed by Wait(R). So, this ensures that no two iterations of either X or Y can proceed without an iteration of the other being executed in between. i.e., this ensures that all n iterations of X and Y succeeds and hence the **answer**.

D. Process X does Signal(R) followed by Wait(S) while Process Y does Signal(S) followed by Wait(R). There is a problem here that X can do two Signal(R) operation without a Wait(R) being done in between by Y. This happens in the following

scenario:
Process Y: Does Signal (S); Wait(R) fails; goes to sleep.
Process X: Does Signal(R); Wait(S) succeeds; In next iteration Signal(R) again happens;

So, this can result in some Signal operations getting lost as the semaphore is a binary one and thus Process Y may not be able to complete all the iterations. If we change the order of Signal(S) and Wait(R) in EntryY, then (D) option also can work.

👍 123 votes -- Arjun Suresh (332k points)

5.16.37 Process Synchronization: GATE CSE 2014 Set 2 | Question: 31 top https://gateoverflow.in/1990

✔

A. **False** : Producer $= P$ (let), consumer $= C$(let) , once producer produce the item and put into the buffer. It will up the s and n to 1, so consumer can easily consume the item. So, option (**A**) Is false.
Code can be execute in this way: $P : 1\ 2\ 3\ 4\ 5|\ C : 1\ 2\ 3\ 4\ 5$. So, consumer can consume item after adding the item to buffer.

B. **Is also False**, because whenever item is added to buffer means after producing the item, consumer can consume the item or we can say remove the item, if here statement is like the consumer will remove no more than one item from the buffer just after the removing one then it will be true (due $n = 0$ then, it will be blocked) but here only asking about the consumer will remove no more than one item from the buffer so, its false.

C. **is true** , statement says if consumer execute first means buffer is empty. Then execution will be like this.
$C : 1$ (wait on s, $s = 0$ now) 2(BLOCK $n = -1$) |P : 1 2 (wait on s which is already 0 so, it now block). So, c wants n which is held by producer or we can say up by only producer and P wants s, which will be up by only consumer. **(circular wait)** surely there is **deadlock**.

D. **is false**, if $n = 1$ then, also it will not free from deadlock.
For the given execution: $C : 1\ 2\ 3\ 4\ 5\ 1\ 2$(BLOCK) $|P : 1\ 2$(BLOCK) so, deadlock.
(here, 1 2 3 4 5 are the lines of the given code)

Hence, **answer is (C)**

👍 70 votes -- minal (13.1k points)

5.16.38 Process Synchronization: GATE CSE 2015 Set 1 | Question: 9 top https://gateoverflow.in/8121

✔ 3 distinct values $\{2, 3, 4\}$

$P1 - P2 : B = 3$
$P2 - P1 : B = 4$
$P1 - P2 - P1 : B = 2$

👍 44 votes -- Anoop Sonkar (4.1k points)

5.16.39 Process Synchronization: GATE CSE 2015 Set 3 | Question: 10 top https://gateoverflow.in/8405

✔ When both processes try to enter critical section simultaneously, both are allowed to do so since both shared variables varP and varQ are true. So, clearly there is **NO mutual exclusion.** Also, **deadlock is prevented** because mutual exclusion is one of the necessary condition for deadlock to happen. Hence, answer is (**A**).

👍 92 votes -- Tanaya Pradhan (701 points)

5.16.40 Process Synchronization: GATE CSE 2016 Set 2 | Question: 48 top https://gateoverflow.in/39600

✔ There is strict alternation i.e. after completion of process 0 if it wants to start again.It will have to wait until process 1 gives the lock.
This violates progress requirement which is, that no other process outside critical section can stop any other interested process from entering the critical section.
Hence the answer is that it violates the progress requirement.

The given solution does not violate bounded waiting requirement.

Bounded waiting is : There exists a bound, or limit, on the number of times other processes are allowed to enter their critical sections after a process has made request to enter its critical section and before that request is granted.

Here there are only two processes and when process 0 enters CS, next entry is reserved for process 1 and vice-versa (strict alteration). So, bounded waiting condition is satisfied here.

Correct Answer: C

65 votes -- bahirNaik (2.4k points)

5.16.41 Process Synchronization: GATE CSE 2017 Set 1 | Question: 27 top https://gateoverflow.in/118307

✔ If we see definition of **reentrant Lock :**

In computer science, the **reentrant mutex** (**recursive mutex**, **recursive lock**) is particular type of mutual exclusion (mutex) device that may be locked multiple times by the same process/thread, **without causing a deadlock.**
https://en.wikipedia.org/wiki/Reentrant_mutex

A **Re-entrantLock** is *owned* by the thread last successfully locking, but not yet unlocking it. A thread invoking **lock** will return, successfully acquiring the lock, when the lock is not owned by another thread. **The method will return immediately if the current thread already owns the lock** https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/locks/ReentrantLock.html

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.

Here, the answer is (**D**).

References

64 votes -- harkirat31 (367 points)

5.16.42 Process Synchronization: GATE CSE 2018 | Question: 40 top https://gateoverflow.in/204114

✔ Empty denotes number of Filled slots.

Full number of empty slots.

So, Producer must dealing with Empty and Consumer deals with Full

Producer must checks Full i,e. decrease Full by 1 before entering and Consumer check with Empty decrease Full by 1 before entering

So, (C) must be answer.

25 votes -- Prashant Singh (47.2k points)

5.16.43 Process Synchronization: GATE CSE 2019 | Question: 23 top https://gateoverflow.in/302825

✔ $D = 100$
Arithmetic operations are not ATOMIC.
These are three step process:

1. Read
2. Calculate
3. Update

Maximum value:
Run P2 for Read and Calculate. $D = 100$
Run P1 for read and calculate. $D = 100$
Run P2 update. D = 50
Run P1 update. D = 110
Run P2 read, calculate and update. $D = 130$

Minimum Value:
Run P1, P2, P3 for Read and Calculate. $D = 100$
Run P1 update. D = 110
Run P3 update. D = 120
Run P2 update. D = 50

Difference between Maximum and Minimum $= 130 - 50 = 80$

29 votes

-- Digvijay (44.9k points)

5.16.44 Process Synchronization: GATE CSE 2019 | Question: 39 top https://gateoverflow.in/302809

The process P, holds X_p resources currently and it doesn't request any new resources. Therefore after some time, it will completes it's execution and release the resources which it holds.

The process Q, holds X_q resources currently and it doesn't request any new resources. Therefore after some time, it will completes it's execution and release the resources which it holds.

Total available resources after completion of P and $Q = X_p + X_q$.

If these resources can not satisfy any process new requests, then no process will be able to completes it's execution.

$X_p + X_q < \text{ Min}\{Y_k \mid 1 \leq k \leq n, k \neq p, k \neq q\} \implies$ delivers that no process going to completes except P and Q. Answer is **(A)**

23 votes

-- Shaik Masthan (50.4k points)

5.16.45 Process Synchronization: GATE IT 2004 | Question: 65 top https://gateoverflow.in/3708

✔ P_1 is the producer. So, it must wait for full condition. But semaphore full is initialized to 0 and semaphore empty is initialized to n, meaning $full = 0$ implies no item and $empty = n$ implies space for n items is available. So, P_1 must wait for semaphore empty - $K - P($ empty $)$ and similarly P_2 must wait for semaphore full - $M - P($ full $)$. After accessing the critical section (producing/consuming item) they do their respective V operation. Thus option D.

49 votes

-- Arjun Suresh (332k points)

5.16.46 Process Synchronization: GATE IT 2005 | Question: 41 top https://gateoverflow.in/3788

✔ Check : [What is Starvation?](#)

Here P_2 can go in infinite waiting while process P_1 executes infinitely long.

Also, it can be the case that the Process P_1 starves for ∞ long time on the semaphore S, after it has successfully executed its critical section once, while P_2 executes infinitely long.

Both P_1 and P_2 can starve for ∞ long period of time.

Answer is **option A.**

References

63 votes

-- Amar Vashishth (25.2k points)

5.16.47 Process Synchronization: GATE IT 2005 | Question: 42 top https://gateoverflow.in/3789

✔ Answer is (**B**)

It needs two semaphores. $X = 0, Y = 0$

P1	P2
P(X)	
	R1
	V(X)
R1	P(Y)
R2	
V(Y)	
P(X)	R2
	R3
	V(X)
R3	P(Y)
R4	
V(Y)	
	R4

85 votes

-- Sandeep_Uniyal (6.5k points)

5.16.48 Process Synchronization: GATE IT 2006 | Question: 55 top

https://gateoverflow.in/3598

✔ Suppose the slots are full $\rightarrow F = 0$. Now, if Wait(F) and Wait(S) are interchanged and Wait(S) succeeds, The producer will wait for Wait(F) which is never going to succeed as Consumer would be waiting for Wait(S). So, deadlock can happen.

If Signal(S) and Signal(F) are interchanged in Consumer, deadlock won't happen. It will just give priority to a producer compared to the next consumer waiting.

So, answer **(A)**

62 votes

-- Arjun Suresh (332k points)

5.16.49 Process Synchronization: GATE IT 2007 | Question: 10 top

https://gateoverflow.in/3443

✔ **(C)** Both process can run the critical section concorrently. Lets say p_1 starts and it enters inside if claus and just after its entertence and before execution of critical_flag = TRUE, a context switch happens and p_2 also gets entrance since the flag is still false. So, now both process are in critical section! So, **(i)** is true. **(ii)** is false there is no way that flag is true and no process' are inside the if clause, if someone enters the critical section, it will definetly make flag = false. So. no. deadlock.

57 votes

-- Vicky Bajoria (4.1k points)

5.16.50 Process Synchronization: GATE IT 2007 | Question: 56 top

https://gateoverflow.in/3498

✔ **Answer is (C)**

$S1$: if readcount is 1 i.e., some reader is reading, DOWN on wrt so that no writer can write.

$S2$: After readcount has been updated, UP on mutex.

$S3$: DOWN on mutex to update readcount

$S4$: If readcount is zero i.e., no reader is reading, UP on wrt to allow some writer to write

26 votes

-- Sandeep_Uniyal (6.5k points)

5.16.51 Process Synchronization: GATE IT 2008 | Question: 53 top

https://gateoverflow.in/3363

✔ Producer: consumer: while (true) do while (true) do 1 $P(S)$; 1 $P(Q)$; 2 $x =$ produce (); 2 consume (x); 3 $V(Q)$; 3 $V(S)$; done done

Lets explain the working of this code.

It is mentioned that P and C execute parallely.

P : 123

1. S value is 1, down on 1 makes it 0. Enters the statement 2.

2. Item produced.
3. Up on Q is done (Since the queue of Q is empty, value of Q up to 1).

This being an infinite while loop should infinitely iterate.

In the next iteration of while loop $st1$ is executed.

But S is already 0, further down on 0 sends P to blocked list of S. P is blocked.

C Consumer is scheduled.

Down on Q. value makes Q.value$=0$;

Enters the statement 2, consumes the item.

Up on S,now instead of changing the value of S. value to 1, wakes up the blocked process on Q 's queue.Hence process P is awaken. P resumes from statement 2, since it was blocked at statement 1. So, P now produces the next item.

So, consumer consumes an item before producer produces the next item.

(D) Answer

(A) Deadlock cannot happen has both producer and consumer are operating on different semaphores (no hold and wait)

(B) No starvation happen because there is alteration between P and Consumer. Which also makes them have bounded waiting.

34 votes

-- Sourav Roy (2.9k points)

5.17 Resource Allocation (26) top

5.17.1 Resource Allocation: GATE CSE 1988 | Question: 11 top

https://gateoverflow.in/94397

A number of processes could be in a deadlock state if none of them can execute due to non-availability of sufficient resources. Let $P_i, 0 \leq i \leq 4$ represent five processes and let there be four resources types $r_j, 0 \leq j \leq 3$. Suppose the following data structures have been used.

Available: A vector of length 4 such that if Available $[i] = k$, there are k instances of resource type r_j available in the system.

Allocation. A 5×4 matrix defining the number of each type currently allocated to each process. If Allocation $[i, j] = k$ then process p_i is currently allocated k instances of resource type r_j.

Max. A 5×4 matrix indicating the maximum resource need of each process. If $Max[i, j] = k$ then process p_i, may need a maximum of k instances of resource type r_j in order to complete the task.

Assume that system allocated resources only when it does not lead into an unsafe state such that resource requirements in future never cause a deadlock state. Now consider the following snapshot of the system.

	Allocation				Max			
	r_0	r_1	r_2	r_3	r_0	r_1	r_2	r_3
p_0	0	0	1	2	0	0	1	2
p_1	1	0	0	0	1	7	5	0
p_2	1	3	5	4	2	3	5	6
p_3	0	6	3	2	0	6	5	2
p_4	0	0	1	4	0	6	5	6

Available			
r_0	r_1	r_2	r_3
1	5	2	0

Is the system currently in a safe state? If yes, explain why.

gate1988 normal descriptive operating-system resource-allocation

Answer

5.17.2 Resource Allocation: GATE CSE 1989 | Question: 11a top

https://gateoverflow.in/91093

i. A system of four concurrent processes, P, Q, R and S, use shared resources A, B and C. The sequences in which processes, P, Q, R and S request and release resources are as follows:

Process P:	1.	P requests A
	2.	P requests B
	3.	P releases A
	4.	P releases B
Process Q:	1.	Q requests C
	2.	Q requests A
	3.	Q releases C
	4.	P releases A
Process R:	1.	R requests B
	2.	R requests C
	3.	R releases B
	4.	R releases C
Process S:	1.	S requests A
	2.	S requests C
	3.	S releases A
	4.	S releases C

If a resource is free, it is granted to a requesting process immediately. There is no preemption of granted resources. A resource is taken back from a process only when the process explicitly releases it.

Can the system of four processes get into a deadlock? If yes, give a sequence (ordering) of operations (for requesting and releasing resources) of these processes which leads to a deadlock.

ii. Will the processes always get into a deadlock? If your answer is no, give a sequence of these operations which leads to completion of all processes.
iii. What strategies can be used to prevent deadlocks in a system of concurrent processes using shared resources if preemption of granted resources is not allowed?

descriptive | gate1989 | operating-system | resource-allocation

Answer

5.17.3 Resource Allocation: GATE CSE 1992 | Question: 02-xi top https://gateoverflow.in/568

A computer system has 6 tape devices, with n processes competing for them. Each process may need 3 tape drives. The maximum value of n for which the system is guaranteed to be deadlock-free is:

A. 2
B. 3
C. 4
D. 1

gate1992 | operating-system | resource-allocation | normal | multiple-selects

Answer

5.17.4 Resource Allocation: GATE CSE 1993 | Question: 7.9, UGCNET-Dec2012-III: 41 top https://gateoverflow.in/2297

Consider a system having m resources of the same type. These resources are shared by 3 processes $A, B,$ and C which have peak demands of $3, 4,$ and 6 respectively. For what value of m deadlock will not occur?

A. 7
B. 9
C. 10
D. 13
E. 15

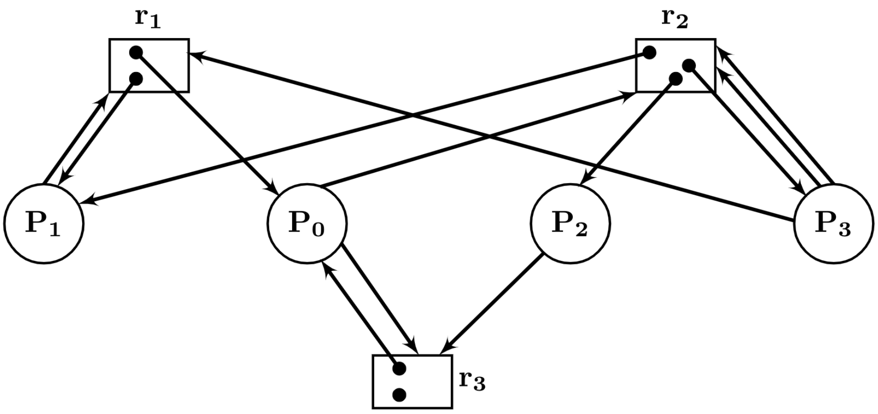
gate1993 | operating-system | resource-allocation | normal | ugcnetdec2012iii | multiple-selects

Answer

5.17.5 Resource Allocation: GATE CSE 1994 | Question: 28 top

https://gateoverflow.in/2524

Consider the resource allocation graph in the figure.



A. Find if the system is in a deadlock state
B. Otherwise, find a safe sequence

gate1994 operating-system resource-allocation normal descriptive

Answer

5.17.6 Resource Allocation: GATE CSE 1996 | Question: 22 top

https://gateoverflow.in/2774

A computer system uses the Banker's Algorithm to deal with deadlocks. Its current state is shown in the table below, where $P0$, $P1$, $P2$ are processes, and $R0$, $R1$, $R2$ are resources types.

	Maximum Need				Current Allocation			Available		
	R0	**R1**	**R2**		**R0**	**R1**	**R2**	**R0**	**R1**	**R2**
P0	4	1	2	**P0**	1	0	2	2	2	0
P1	1	5	1	**P1**	0	3	1			
P2	1	2	3	**P2**	1	0	2			

A. Show that the system can be in this state
B. What will the system do on a request by process $P0$ for one unit of resource type $R1$?

gate1996 operating-system resource-allocation normal descriptive

Answer

5.17.7 Resource Allocation: GATE CSE 1997 | Question: 6.7 top

https://gateoverflow.in/2263

An operating system contains 3 user processes each requiring 2 units of resource R. The minimum number of units of R such that no deadlocks will ever arise is

A. 3
B. 5
C. 4
D. 6

gate1997 operating-system resource-allocation normal

Answer

5.17.8 Resource Allocation: GATE CSE 1997 | Question: 75 top https://gateoverflow.in/19705

An operating system handles requests to resources as follows.

A process (which asks for some resources, uses them for some time and then exits the system) is assigned a unique timestamp are when it starts. The timestamps are monotonically increasing with time. Let us denote the timestamp of a process P by $TS(P)$.

When a process P requests for a resource the OS does the following:

i. If no other process is currently holding the resource, the OS awards the resource to P.
ii. If some process Q with $TS(Q) < TS(P)$ is holding the resource, the OS makes P wait for the resources.
iii. If some process Q with $TS(Q) > TS(P)$ is holding the resource, the OS restarts Q and awards the resources to P. (Restarting means taking back the resources held by a process, killing it and starting it again with the same timestamp)

When a process releases a resource, the process with the smallest timestamp (if any) amongst those waiting for the resource is awarded the resource.

A. Can a deadlock over arise? If yes, show how. If not prove it.
B. Can a process P ever starve? If yes, show how. If not prove it.

gate1997 operating-system resource-allocation normal descriptive

Answer

5.17.9 Resource Allocation: GATE CSE 1998 | Question: 1.32 top https://gateoverflow.in/1669

A computer has six tape drives, with n processes competing for them. Each process may need two drives. What is the maximum value of n for the system to be deadlock free?

A. 6
B. 5
C. 4
D. 3

gate1998 operating-system resource-allocation normal

Answer

5.17.10 Resource Allocation: GATE CSE 2000 | Question: 2.23 top https://gateoverflow.in/670

Which of the following is not a valid deadlock prevention scheme?

A. Release all resources before requesting a new resource.
B. Number the resources uniquely and never request a lower numbered resource than the last one requested.
C. Never request a resource after releasing any resource.
D. Request and all required resources be allocated before execution.

gate2000-cse operating-system resource-allocation normal

Answer

5.17.11 Resource Allocation: GATE CSE 2001 | Question: 19 top https://gateoverflow.in/760

Two concurrent processes $P1$ and $P2$ want to use resources $R1$ and $R2$ in a mutually exclusive manner. Initially, $R1$ and $R2$ are free. The programs executed by the two processes are given below.

Program for P1:		Program for P2:	
S1:	While ($R1$ is busy) do no-op;	Q1:	While ($R1$ is busy) do no-op;
S2:	Set $R1 \leftarrow$ busy;	Q2:	Set $R1 \leftarrow$ busy;
S3:	While ($R2$ is busy) do no-op;	Q3:	While ($R2$ is busy) do no-op;
S4:	Set $R2 \leftarrow$ busy;	Q4:	Set $R2 \leftarrow$ busy;
S5:	Use $R1$ and $R2$;	Q5:	Use $R1$ and $R2$;
S6:	Set $R1 \leftarrow$ free;	Q6:	Set $R2 \leftarrow$ free;
S7:	Set $R2 \leftarrow$ free;	Q7:	Set $R1 \leftarrow$ free;

A. Is mutual exclusion guaranteed for $R1$ and $R2$? If not show a possible interleaving of the statements of $P1$ and $P2$ such mutual exclusion is violated (i.e., both $P1$ and $P2$ use $R1$ and $R2$ at the same time).
B. Can deadlock occur in the above program? If yes, show a possible interleaving of the statements of $P1$ and $P2$ leading to deadlock.
C. Exchange the statements $Q1$ and $Q3$ and statements $Q2$ and $Q4$. Is mutual exclusion guaranteed now? Can deadlock occur?

gate2001-cse operating-system resource-allocation normal descriptive

Answer

5.17.12 Resource Allocation: GATE CSE 2005 | Question: 71 top https://gateoverflow.in/1394

Suppose n processes, $P_1, \dots P_n$ share m identical resource units, which can be reserved and released one at a time. The maximum resource requirement of process P_i is s_i, where $s_i > 0$. Which one of the following is a sufficient condition for ensuring that deadlock does not occur?

A. $\forall i,\ s_i, < m$
B. $\forall i,\ s_i < n$
C. $\sum_{i=1}^{n} s_i < (m+n)$
D. $\sum_{i=1}^{n} s_i < (m \times n)$

gate2005-cse operating-system resource-allocation normal

Answer

5.17.13 Resource Allocation: GATE CSE 2006 | Question: 66 top https://gateoverflow.in/1844

Consider the following snapshot of a system running n processes. Process i is holding x_i instances of a resource R, $1 \leq i \leq n$. Currently, all instances of R are occupied. Further, for all i, process i has placed a request for an additional y_i instances while holding the x_i instances it already has. There are exactly two processes p and q and such that $y_p = y_q = 0$. Which one of the following can serve as a necessary condition to guarantee that the system is not approaching a deadlock?

A. $\min(x_p, x_q) < \max_{k \neq p,q} y_k$
B. $x_p + x_q \geq \min_{k \neq p,q} y_k$
C. $\max(x_p, x_q) > 1$
D. $\min(x_p, x_q) > 1$

gate2006-cse operating-system resource-allocation normal

Answer

5.17.14 Resource Allocation: GATE CSE 2007 | Question: 57 top https://gateoverflow.in/1255

A single processor system has three resource types X, Y and Z, which are shared by three processes. There are 5 units of each resource type. Consider the following scenario, where the column **alloc** denotes the number of units of each resource type allocated to each process, and the column **request** denotes the number of units of each resource type requested by a process in order to complete execution. Which of these processes will finish **LAST**?

	alloc			request		
	X	Y	Z	X	Y	Z
P0	1	2	1	1	0	3
P1	2	0	1	0	1	2
P2	2	2	1	1	2	0

A. $P0$
B. $P1$
C. $P2$
D. None of the above, since the system is in a deadlock

gate2007-cse operating-system resource-allocation normal

Answer

5.17.15 Resource Allocation: GATE CSE 2008 | Question: 65 top

https://gateoverflow.in/488

Which of the following is NOT true of deadlock prevention and deadlock avoidance schemes?

A. In deadlock prevention, the request for resources is always granted if the resulting state is safe
B. In deadlock avoidance, the request for resources is always granted if the resulting state is safe
C. Deadlock avoidance is less restrictive than deadlock prevention
D. Deadlock avoidance requires knowledge of resource requirements *apriori..*

gate2008-cse operating-system easy resource-allocation

Answer

5.17.16 Resource Allocation: GATE CSE 2009 | Question: 30 top

https://gateoverflow.in/1316

Consider a system with 4 types of resources $R1$ (3 units), $R2$ (2 units), $R3$ (3 units), $R4$ (2 units). A non-preemptive resource allocation policy is used. At any given instance, a request is not entertained if it cannot be completely satisfied. Three processes $P1$, $P2$, $P3$ request the resources as follows if executed independently.

Process P1:	**Process P2:**	**Process P3:**
$t = 0$: requests 2 units of $R2$ $t = 1$: requests 1 unit of $R3$ $t = 3$: requests 2 units of $R1$ $t = 5$: releases 1 unit of $R2$ and 1 unit of $R1$ $t = 7$: releases 1 unit of $R3$ $t = 8$: requests 2 units of $R4$ $t = 10$: Finishes	$t = 0$: requests 2 units of $R3$ $t = 2$: requests 1 unit of $R4$ $t = 4$: requests 1 unit of $R1$ $t = 6$: releases 1 unit of $R3$ $t = 8$: Finishes	$t = 0$: requests 1 unit of $R4$ $t = 2$: requests 2 units of $R1$ $t = 5$: releases 2 units of $R1$ $t = 7$: requests 1 unit of $R2$ $t = 8$: requests 1 unit of $R3$ $t = 9$: Finishes

Which one of the following statements is TRUE if all three processes run concurrently starting at time $t = 0$?

A. All processes will finish without any deadlock
B. Only $P1$ and $P2$ will be in deadlock
C. Only $P1$ and $P3$ will be in deadlock
D. All three processes will be in deadlock

gate2009-cse operating-system resource-allocation normal

Answer

5.17.17 Resource Allocation: GATE CSE 2010 | Question: 46 top

https://gateoverflow.in/2348

A system has n resources $R_0, \ldots, R_{n-1}$, and k processes $P_0, \ldots, P_{k-1}$. The implementation of the resource request logic of each process P_i is as follows:

$\text{if}(i\%2 == 0)\{\quad \text{if}(i < n) \text{ request } R_i; \quad \text{if}(i+2 < n) \text{ request } R_{i+2}; \}\text{else}\{\quad \text{if}(i < n) \text{ request } R_{n-i}; \quad \text{if}(i+2 < n) \text{ request } R_{n\cdot}$

In which of the following situations is a deadlock possible?

A. $n = 40,\ k = 26$
B. $n = 21,\ k = 12$
C. $n = 20,\ k = 10$
D. $n = 41,\ k = 19$

gate2010-cse operating-system resource-allocation normal

Answer

5.17.18 Resource Allocation: GATE CSE 2013 | Question: 16 top

https://gateoverflow.in/1438

Three concurrent processes X, Y, and Z execute three different code segments that access and update certain shared variables. Process X executes the P operation (i.e., $wait$) on semaphores a, b and c; process Y executes the P operation on semaphores b, c and d; process Z executes the P operation on semaphores c, d, and a before entering the respective code segments. After completing the execution of its code segment, each process invokes the V operation (i.e., $signal$) on its three semaphores. All semaphores are binary semaphores initialized to one. Which one of the following represents a deadlock-free order of invoking the P operations by the processes?

A. $X: P(a)P(b)P(c)\ Y: P(b)P(c)P(d)\ Z: P(c)P(d)P(a)$
B. $X: P(b)P(a)P(c)\ Y: P(b)P(c)P(d)\ Z: P(a)P(c)P(d)$
C. $X: P(b)P(a)P(c)\ Y: P(c)P(b)P(d)\ Z: P(a)P(c)P(d)$
D. $X: P(a)P(b)P(c)\ Y: P(c)P(b)P(d)\ Z: P(c)P(d)P(a)$

gate2013-cse operating-system resource-allocation normal

Answer

5.17.19 Resource Allocation: GATE CSE 2014 Set 1 | Question: 31 top

https://gateoverflow.in/1800

An operating system uses the *Banker's algorithm* for deadlock avoidance when managing the allocation of three resource types $X, Y,$ and Z to three processes $P0, P1,$ and $P2$. The table given below presents the current system state. Here, the *Allocation matrix* shows the current number of resources of each type allocated to each process and the *Max matrix* shows the maximum number of resources of each type required by each process during its execution.

	Allocation			**Max**		
	X	Y	Z	X	Y	Z
P0	0	0	1	8	4	3
P1	3	2	0	6	2	0
P2	2	1	1	3	3	3

There are 3 units of type X, 2 units of type Y and 2 units of type Z still available. The system is currently in a **safe** state. Consider the following independent requests for additional resources in the current state:

REQ1: $P0$ requests 0 units of X, 0 units of Y and 2 units of Z

REQ2: $P1$ requests 2 units of X, 0 units of Y and 0 units of Z

Which one of the following is **TRUE**?

A. Only REQ1 can be permitted.
B. Only REQ2 can be permitted.
C. Both REQ1 and REQ2 can be permitted.
D. Neither REQ1 nor REQ2 can be permitted.

gate2014-cse-set1 operating-system resource-allocation normal

Answer

5.17.20 Resource Allocation: GATE CSE 2014 Set 3 | Question: 31 top

https://gateoverflow.in/2065

A system contains three programs and each requires three tape units for its operation. The minimum number of tape units which the system must have such that deadlocks never arise is _________.

gate2014-cse-set3 operating-system resource-allocation numerical-answers easy

Answer

5.17.21 Resource Allocation: GATE CSE 2015 Set 2 | Question: 23 top

https://gateoverflow.in/8114

A system has 6 identical resources and N processes competing for them. Each process can request at most 2 requests. Which one of the following values of N could lead to a deadlock?

A. 1
B. 2
C. 3
D. 4

gate2015-cse-set2 operating-system resource-allocation easy

Answer

5.17.22 Resource Allocation: GATE CSE 2015 Set 3 | Question: 52 top

https://gateoverflow.in/8561

Consider the following policies for preventing deadlock in a system with mutually exclusive resources.

I. Process should acquire all their resources at the beginning of execution. If any resource is not available, all resources acquired so far are released.
II. The resources are numbered uniquely, and processes are allowed to request for resources only in increasing resource numbers
III. The resources are numbered uniquely, and processes are allowed to request for resources only in deccreasing resource numbers
IV. The resources are numbered uniquely. A processes is allowed to request for resources only for a resource with resource number larger than its currently held resources

Which of the above policies can be used for preventing deadlock?

A. Any one of (I) and (III) but not (II) or (IV)
B. Any one of (I), (III) and (IV) but not (II)
C. Any one of (II) and (III) but not (I) or (IV)
D. Any one of (I), (II), (III) and (IV)

gate2015-cse-set3 operating-system resource-allocation normal

Answer

5.17.23 Resource Allocation: GATE CSE 2016 Set 1 | Question: 50 top

https://gateoverflow.in/39719

Consider the following proposed solution for the critical section problem. There are n processes : $P_0 \dots P_{n-1}$. In the code, function pmax returns an integer not smaller than any of its arguments .For all $i, t[i]$ is initialized to zero.

Code for P_i;

```
do {
    c[i]=1; t[i]= pmax (t[0],....,t[n-1])+1; c[i]=0;
    for every j != i in {0,....,n-1} {
        while (c[j]);
        while (t[j] != 0 && t[j] <=t[i]);
    }
    Critical Section;
    t[i]=0;

    Remainder Section;

}  while (true);
```

Which of the following is TRUE about the above solution?

A. At most one process can be in the critical section at any time
B. The bounded wait condition is satisfied
C. The progress condition is satisfied
D. It cannot cause a deadlock

gate2016-cse-set1 operating-system resource-allocation difficult ambiguous

Answer

5.17.24 Resource Allocation: GATE CSE 2017 Set 2 | Question: 33 top

https://gateoverflow.in/118375

A system shares 9 tape drives. The current allocation and maximum requirement of tape drives for that processes are shown below:

Process	Current Allocation	Maximum Requirement
P1	3	7
P2	1	6
P3	3	5

Which of the following best describes current state of the system?

A. Safe, Deadlocked
B. Safe, Not Deadlocked
C. Not Safe, Deadlocked
D. Not Safe, Not Deadlocked

gate2017-cse-set2 operating-system resource-allocation normal

Answer

5.17.25 Resource Allocation: GATE IT 2005 | Question: 62 top

https://gateoverflow.in/3823

Two shared resources R_1 and R_2 are used by processes P_1 and P_2. Each process has a certain priority for accessing each resource. Let T_{ij} denote the priority of P_i for accessing R_j. A process P_i can snatch a resource R_k from process P_j if T_{ik} is greater than T_{jk}.
Given the following :

I. $T_{11} > T_{21}$
II. $T_{12} > T_{22}$
III. $T_{11} < T_{21}$
IV. $T_{12} < T_{22}$

Which of the following conditions ensures that P_1 and P_2 can never deadlock?

A. (I) and (IV)
B. (II) and (III)
C. (I) and (II)
D. None of the above

gate2005-it operating-system resource-allocation normal

Answer

5.17.26 Resource Allocation: GATE IT 2008 | Question: 54 top

https://gateoverflow.in/3364

An operating system implements a policy that requires a process to release all resources before making a request for another resource. Select the TRUE statement from the following:

A. Both starvation and deadlock can occur
B. Starvation can occur but deadlock cannot occur
C. Starvation cannot occur but deadlock can occur
D. Neither starvation nor deadlock can occur

gate2008-it operating-system resource-allocation normal

Answer

Answers: Resource Allocation

5.17.1 Resource Allocation: GATE CSE 1988 | Question: 11 top
https://gateoverflow.in/94397

✔ Here, we are asked to "Avoid Deadlock" and Bankers Algorithm is the algorithm for this.

The crux of the algorithm is to allocate resources to a process only if there exist a **safe sequence after** the allocation. i.e., after allocating the requested resources there exist a sequence of execution of the processes such that deadlock would not happen. There can be multiple safe sequences but we need to get any one of them to say that a state is safe.

Now coming to the given question, first lets make the NEED matrix which shows the future need of all the processes and can be obtained by $\text{Max} - \text{Allocation}$.

$$\begin{array}{ccccc} & & \text{Max} & & \\ & r_0 & r_1 & r_2 & r_3 \\ p_0 & 0 & 0 & 1 & 2 \\ p_1 & 1 & 7 & 5 & 0 \\ p_2 & 2 & 3 & 5 & 6 \\ p_3 & 0 & 6 & 5 & 2 \\ p_4 & 0 & 6 & 5 & 6 \end{array} - \begin{array}{ccccc} & & \text{Allocation} & & \\ & r_0 & r_1 & r_2 & r_3 \\ p_0 & 0 & 0 & 1 & 2 \\ p_1 & 1 & 0 & 0 & 0 \\ p_2 & 1 & 3 & 5 & 4 \\ p_3 & 0 & 6 & 3 & 2 \\ p_4 & 0 & 0 & 1 & 4 \end{array} = \begin{array}{ccccc} & & \text{Need} & & \\ & r_0 & r_1 & r_2 & r_3 \\ p_0 & 0 & 0 & 0 & 0 \\ p_1 & 0 & 7 & 5 & 0 \\ p_2 & 1 & 0 & 0 & 2 \\ p_3 & 0 & 0 & 2 & 0 \\ p_4 & 0 & 6 & 4 & 2 \end{array}.$$

Since P_0 does not require any more resource we can finish this first releasing 1 instance of r_2 and 2 instances of r_3. Thus our Available vector becomes

$$[1 \quad 5 \quad 2 \quad 0] + [0 \quad 0 \quad 1 \quad 2] = [1 \quad 5 \quad 3 \quad 2].$$

Now, either p_2 or p_3 can finish as both their requirements are not greater than the Available vector. Say, p_2 finishes. It releases $[2 \quad 3 \quad 5 \quad 6]$ and our Available becomes

$$[1 \quad 5 \quad 3 \quad 2] + [2 \quad 3 \quad 5 \quad 6] = [3 \quad 8 \quad 8 \quad 8].$$

Now, any of p_1, p_3, p_4 can finish and so we do not need to proceed further to determine that the state is safe. One of the possible safe sequence is

$$p_0 - p_2 - p_1 - p_3 - p_4.$$

5 votes

-- Arjun Suresh (332k points)

5.17.2 Resource Allocation: GATE CSE 1989 | Question: 11a top
https://gateoverflow.in/91093

i) Assuming only one instance of a resource is available,

Process P: Hold A, request B

Process Q: Hold C, request A

Process R: Hold B, request C

Process S: Request A, request C

In this instance, Process P,Q,R and S are waiting for the release of resources among each other and none of them can proceed. This is deadlock.

ii) Any sequential ordering will be free from deadlock. An instance(concurrent) can be:

Process P	**Process Q**	**Process R**	**Process S**
request A			
request B			
	request C		
release A			
	request A		
	release C		
release B			
		request B	
		request C	
	release A		
			request A
		release B	
		release C	
			request C

All the requests of all processes are satisfied and leads to completion of all processes.

iii) To prevent deadlock:

- Resources can be shared (violating mutual exclusion)
- Not allowing processes to hold a resource and request for another(violating hold and wait)
- Break circular wait by allocating resources in some order
- Banker's algorithm(safe state)- to avoid deadlock

9 votes -- Manoja Rajalakshmi Aravindakshan (7.7k points)

5.17.3 Resource Allocation: GATE CSE 1992 | Question: 02-xi top https://gateoverflow.in/568

✔ Allocate max-1 resources to all processes and add one more resource to any process (Pigeon hole principle) so that this particular process can be completed (resources can be freed) and there is no deadlock.

Max resources required is 3.

$\therefore (3-1)*n+1=6$

$n = \lfloor \frac{5}{2} \rfloor = 2$

Correct Answer: A

10 votes -- Manoja Rajalakshmi Aravindakshan (7.7k points)

Answer: (**A**).

For $n=3, 2-2-2$ combination of resources leads to deadlock.

For $n=2, 3-3$ is the maximum need and that can always be satisfied.

18 votes -- Rajarshi Sarkar (27.9k points)

5.17.4 Resource Allocation: GATE CSE 1993 | Question: 7.9, UGCNET-Dec2012-III: 41 top https://gateoverflow.in/2297

✔ **13 and 15.**

Consider the worst scenario: all processes require one more instance of the resource. So, $P1$ would have got 2, $P2-3$ and $P3-5$. Now, if one more resource is available at least one of the processes could be finished and all resources allotted to it will be free which will lead to other processes also getting freed. So, $2+3+5=10$ would be the maximum value of m so that a deadlock can occur.

41 votes -- Arjun Suresh (332k points)

5.17.5 Resource Allocation: GATE CSE 1994 | Question: 28 top https://gateoverflow.in/2524

✔ From the RAG we can make the necessary matrices.

Allocation

	r_1	r_2	r_3
P_0	1	0	1
P_1	1	1	0
P_2	0	1	0
P_3	0	1	0

Future Need

	r_1	r_2	r_3
P_0	0	1	1
P_1	1	0	0
P_2	0	0	1
P_3	1	2	0

- $\text{Total} = (2 \quad 3 \quad 2)$
- $\text{Allocated} = (2 \quad 3 \quad 1)$
- $\text{Available} = \text{Total} - \text{Allocated} = (0 \quad 0 \quad 1)$

$P_2's$ need $(0 \quad 0 \quad 1)$ can be met

And it releases its held resources after running to completion

$A = (0 \quad 0 \quad 1) + (0 \quad 1 \quad 0) = (0 \quad 1 \quad 1)$

$P_0's$ need $(0 \quad 1 \quad 1)$ can be met

and it releases
$A = (0 \quad 1 \quad 1) + (1 \quad 0 \quad 1) = (1 \quad 1 \quad 2)$

$P_1's$ needs can be met $(1 \quad 0 \quad 0)$ and it releases
$A = (1 \quad 1 \quad 2) + (1 \quad 1 \quad 0) = (2 \quad 2 \quad 2)$

$P_3's$ need can be met
So, the safe sequence will be $P_2 - P_0 - P_1 - P_3$.

37 votes

-- Sourav Roy (2.9k points)

5.17.6 Resource Allocation: GATE CSE 1996 | Question: 22 top

https://gateoverflow.in/2774

Allocation

	R0	R1	R2
P0	1	0	2
P1	0	3	1
P2	1	0	2

MAX NEED

	R0	R1	R2
P0	4	1	2
P1	1	5	1
P2	1	2	3

Future Need

	R0	R1	R2
P0	3	1	0
P1	1	2	0
P2	0	2	1

Available $= (2 \quad 2 \quad 0)$

$P1(1 \quad 2 \quad 0)$'s needs can be met. P1 executes and completes releases its allocated resources.

$A = (2 \quad 2 \quad 0) + (0 \quad 3 \quad 1) = (2 \quad 5 \quad 1)$

Further $P2(0 \quad 2 \quad 1)$ s needs can be met.

$A = (2 \quad 5 \quad 1) + (1 \quad 0 \quad 2) = (3 \quad 5 \quad 3)$

next $P0$ s needs can be met.

Thus safe sequence exists $P1P2P0$.

Next Request $P0(010)$

Allocation

	R0	R1	R2
P0	1	0+1=1	2
P1	0	3	1
P2	1	0	2

MAX NEED

	R0	R1	R2
P0	4	1	2
P1	1	5	1
P2	1	2	3

Future Need

	R0	R1	R2
P0	3	0	0
P1	1	2	0
P2	0	2	1

Available $= (2 \quad 2-1=1 \quad 0)$
Here, also not a single request need by any process can be made.

a. System is in safe state.
b. Since request of $P0$ can not be met,system would delay the request and wait till resources are available.

24 votes

-- Sourav Roy (2.9k points)

5.17.7 Resource Allocation: GATE CSE 1997 | Question: 6.7 top

https://gateoverflow.in/2263

If we have X number of resources where X is sum of $r_i - 1$ where r_i is the resource requirement of process i, we might have a deadlock. But if we have one more resource, then as per Pigeonhole principle, one of the process must complete and this can eventually lead to all processes completing and thus no deadlock.

Here, $n = 3$ and $r_i = 2$ for all i. So, in order to avoid deadlock **minimum** no. of resources required

$$= \sum_{i=1}^{3}(2-1)+1 = 3+1 = 4.$$

PS: Note the **minimum** word, any higher number will also cause no deadlock.

Correct Answer: C

18 votes

-- hriday (161 points)

5.17.8 Resource Allocation: GATE CSE 1997 | Question: 75 top

https://gateoverflow.in/19705

✔

A. **Can Deadlock occur. No, because every time Older Process who wants some resources which are already acquired by some younger process. In this condition Younger will be killed and release its resources which is now taken by now older process. So never more than one process will wait for some resources indefinitely. Timestamp will also be unique.**
B. **Can a process Starve. No, because every time when Younger process is getting killed, it is restarted with same timestamp which he had at time of killing. So it will act as an elder even after killing for all those who came after it..**

There is No starvation. Consider this scenario:

Say a process p_{12} with TS 12 and another process p_{11} with timestamp 11 so, p_{12} gets killed but again come with **same timestamp**. As timestamp is increasing for newly enter process so at next process p_{13} enter with timestamp 13 which have greater timestamp than p_{12} so, p_{12} gets executed. Hence there is no starvation possible.

35 votes

-- sonu (1.8k points)

5.17.9 Resource Allocation: GATE CSE 1998 | Question: 1.32 top

https://gateoverflow.in/1669

✔ Each process needs 2 drives

Consider this scenario

P_1	P_2	P_3	P_4	P_5	P_6
1	1	1	1	1	1

This is scenario when a deadlock would happen, as each of the process is waiting for 1 more process to run to completion. And there are no more Resources available as max 6 reached. If we could have provided one more R to any of the process, any of the process could have executed to completion, then released its resources, which further when assigned to other and then other would have broken the deadlock situation.

In case of processes, if there are less than 6 processes, then no deadlock occurs.

Consider the maximum case of 5 processes.

P_1	P_2	P_3	P_4	P_5
1	1	1	1	1

In this case system has 6 resources max,and hence we still have 1 more R left which can be given to any of the processes, which in turn runs to completion, releases its resources and in turn others can run to completion too.

Answer (B).

29 votes

-- Sourav Roy (2.9k points)

5.17.10 Resource Allocation: GATE CSE 2000 | Question: 2.23 top

https://gateoverflow.in/670

✔ The answer is (**C**).

A. is valid. Which dissatisfies Hold and Wait but ends up in starvation.
B. is valid. Which is used to dissatisfy circular wait.
C. is invalid.
D. is valid and is used to dissatisfy Hold and Wait.

36 votes

-- Gate Keeda (15.9k points)

5.17.11 Resource Allocation: GATE CSE 2001 | Question: 19 top

https://gateoverflow.in/760

✔

A. Mutual exclusion is not guaranteed;

Initially both $R1$ and $R2$ are free.
Now, consider the scenario:

$P1$ will start and check the condition ($R1$ ==busy) it will be evaluated as false and $P1$ will be preempted.
Then, $P2$ will start and check the condition ($R1$ == busy) it will be evaluated as false and $P2$ will be preempted.
Now, again $P1$ will start execution and set $R1$ = busy then preempted again.
Then $P2$ will start execution and set $R1$ = busy which was already updated by $P1$ and now $P2$ will be preempted.
After that $P1$ will start execution and same scenario happen again with both $P1$ and $P2$.
Both set $R2$ = busy and enter into critical section together.

Hence, Mutual exclusion is not guaranteed.

B. Here, deadlock is not possible, because at least one process is able to proceed and enter into critical section.

C. If $Q1$ and $Q3$; $Q2$ and $Q4$ will be interchanged then Mutual exclusion is guaranteed but deadlock is possible.

Here, both process will not be able to enter critical section together.
For deadlock:

If $P1$ sets $R1$ = busy and then preempted, and $P2$ sets $R2$ = busy then preempted.
In this scenario no process can proceed further, as both holding the resource that is required by other to enter into CS.

Hence, deadlock will be there.

30 votes

-- jayendra (6.7k points)

5.17.12 Resource Allocation: GATE CSE 2005 | Question: 71 top

https://gateoverflow.in/1394

✔ To ensure deadlock never happens allocate resources to each process in following manner:
Worst Case Allocation (maximum resources in use without any completion) will be $(\text{max requirement} - 1)$ allocations for each process. i.e., $s_i - 1$ for each i

Now, if $\sum_{i=1}^{n}(s_i - 1) \leq m$ dead lock can occur if m resources are split equally among the n processes and all of them will be requiring one more resource instance for completion.

Now, if we add just one more resource, one of the process can complete, and that will release the resources and this will eventually result in the completion of all the processes and deadlock can be avoided. i.e., to avoid deadlock

$$\sum_{i=1}^{n}(s_i - 1) + 1 \leq m$$

$$\implies \sum_{i=1}^{n} s_i - n + 1 \leq m$$

$$\implies \sum_{i=1}^{n} s_i < (m + n).$$

Correct Answer: C

104 votes

-- Digvijay (44.9k points)

5.17.13 Resource Allocation: GATE CSE 2006 | Question: 66 top

https://gateoverflow.in/1844

✔ B. $x_p + x_q \geq \min_{k \neq p,q} y_k$

The question asks for "necessary" condition to guarantee no deadlock. i.e., without satisfying this condition "deadlock" MUST be there.

Both the processes p and q have no additional requirements and can be finished releasing $x_p + x_q$ resources. Using this we can finish one more process only if condition B is satisfied.

PS: Condition B just ensures that the system can proceed from the current state. It does not guarantee that there won't be a deadlock before all processes are finished.

65 votes -- Arjun Suresh (332k points)

5.17.14 Resource Allocation: GATE CSE 2007 | Question: 57 top https://gateoverflow.in/1255

The answer is (**C**).

Available Resources

X	Y	Z
0	1	2

Now, $P1$ will execute first, As it meets the needs. After completion, The available resources are updated.

Updated Available Resources

X	Y	Z
2	1	3

Now $P0$ will complete the execution, as it meets the needs.

After completion of $P0$ the table is updated and then $P2$ completes the execution.
Thus $P2$ completes the execution in the last.

27 votes -- Gate Keeda (15.9k points)

5.17.15 Resource Allocation: GATE CSE 2008 | Question: 65 top https://gateoverflow.in/488

(A). In deadlock prevention, we just need to ensure one of the four necessary conditions of deadlock doesn't occur. So, it may be the case that a resource request might be rejected even if the resulting state is safe. (One example, is when we impose a strict ordering for the processes to request resources).

Deadlock avoidance is less restrictive than deadlock prevention. Deadlock avoidance is like a police man and deadlock prevention is like a traffic light. The former is less restrictive and allows more concurrency.

Reference: http://www.cs.jhu.edu/~yairamir/cs418/os4/tsld010.htm

References

72 votes -- Arjun Suresh (332k points)

5.17.16 Resource Allocation: GATE CSE 2009 | Question: 30 top https://gateoverflow.in/1316

At $t = 3$, the process $P1$ has to wait because available $R1 = 1$, but $P1$ needs 2 $R1$. so $P1$ is blocked.
Similarly, at various times what is happening can be analyzed by the table below.

		R1(3)	R2(2)	R3(3)	R4(2)
	t=0	3	0	1	1
	t=1	3	0	0	1
	t=2	1	0	0	0
Block P1	t=3	1	0	0	0
	t=4	0	0	0	0
Unblock P1	t=5	1	1	0	0
	t=6	1	1	1	0
	t=7	1	0	2	0
Block P1	t=8	2	0	2	1
Unblock P1	t=9	2	1	3	0
	t=10				

There are no processes in deadlock, hence **(A) is right choice**

73 votes

-- Sachin Mittal (15.8k points)

5.17.17 Resource Allocation: GATE CSE 2010 | Question: 46 top

https://gateoverflow.in/2348

✔ From the resource allocation logic, it's clear that even numbered processes are taking even numbered resources and all even numbered processes share no more than 1 resource. Now, if we make sure that all odd numbered processes take odd numbered resources without a cycle, then deadlock cannot occur. The "else" case of the resource allocation logic, is trying to do that. But, if n is odd, R_{n-i} and R_{n-i-2} will be even and there is possibility of deadlock, when two processes requests the same R_i and R_j. So, only B and D are the possible answers.

Now, in D, we can see that P_0 requests R_0 and R_2, P_2 requests R_2 and R_4, so on until, P_{18} requests R_{18} and R_{20}. At the same time P_1 requests R_{40} and R_{38}, P_3 requests R_{38} and R_{36}, so on until, P_{17} requests R_{24} and R_{22}. i.e.; there are no two processes requesting the same two resources and hence there can't be a cycle of dependencies which means, no deadlock is possible.

But for B, P_8 requests R_8 and R_{10} and P_{11} also requests R_{10} and R_8. Hence, a deadlock is possible. (Suppose P_8 comes first and occupies R_8. Then P_{11} comes and occupies R_{10}. Now, if P_8 requests R_{10} and P_{11} requests R_8, there will be deadlock)

Correct Answer: B

279 votes

-- Arjun Suresh (332k points)

5.17.18 Resource Allocation: GATE CSE 2013 | Question: 16 top

https://gateoverflow.in/1438

✔ For deadlock-free invocation, X, Y and Z must access the semaphores in the same order so that there won't be a case where one process is waiting for a semaphore while holding some other semaphore. This is satisfied only by option B.

In option A, X can hold a and wait for c while Z can hold c and wait for a
In option C, X can hold b and wait for c, while Y can hold c and wait for b
In option D, X can hold a and wait for c while Z can hold c and wait for a

So, a deadlock is possible for all choices except B.

http://www.eee.metu.edu.tr/~halici/courses/442/Ch5%20Deadlocks.pdf

References

56 votes

-- Arjun Suresh (332k points)

5.17.19 Resource Allocation: GATE CSE 2014 Set 1 | Question: 31 top

https://gateoverflow.in/1800

✔ **Option (B)**

Request 1 if permitted does not lead to a safe state.

After allowing Req 1,

	Allocated			Max			Requirement		
P0	0	0	3	8	4	3	8	4	0
P1	3	2	0	6	2	0	3	0	0
P2	2	1	1	3	3	3	1	2	2

Available : $X = 3, Y = 2, Z = 0$

Now we can satisfy $P1's$ requirement completely. So Available becomes : $X = 6, Y = 4, Z = 0$.

Since, Z is not available now, neither $P0's$ nor $P2's$ requirement can be satisfied. So. it is an unsafe state.

36 votes -- Poulami Das (167 points)

5.17.20 Resource Allocation: GATE CSE 2014 Set 3 | Question: 31 top

https://gateoverflow.in/2065

✔ Up to, 6 resources, there can be a case that all process have 2 each and dead lock can occur. With 7 resources, at least one process's need is satisfied and hence it must go ahead and finish and release all 3 resources it held. So, no dead lock is possible.

25 votes -- Arjun Suresh (332k points)

For these type of problems in which every process is making same number of requests, use the formula

$$n.(m-1)+1 \leq r$$

where,
$n =$ no. of processes
$m =$ resource requests made by processes
$r =$ no. of resources

So, in above problem we get $3.(3-1)+1 \leq r \implies r \geq 7$

Minimum number of resource required to avoid deadlock is 7.

31 votes -- neha pawar (3.3k points)

5.17.21 Resource Allocation: GATE CSE 2015 Set 2 | Question: 23 top

https://gateoverflow.in/8114

✔ $3 \times 2 = 6$
$4 \times 2 = 8$

I guess a question can't get easier than this- (D) choice. (Also, we can simply take the greatest value among choice for this question)

[There are 6 resources and all of them must be in use for deadlock. If the system has no other resource dependence, $N = 4$ cannot lead to a deadlock. But if $N = 4$, the system can be in deadlock in presence of other dependencies.

Why $N = 3$ cannot cause deadlock? It can cause deadlock, only if the system is already in deadlock and so the deadlock is independent of the considered resource. Till $N = 3$, all requests for considered resource will always be satisfied and hence there won't be a waiting and hence no deadlock with respect to the considered resource.]

36 votes -- Arjun Suresh (332k points)

5.17.22 Resource Allocation: GATE CSE 2015 Set 3 | Question: 52 top

https://gateoverflow.in/8561

✔ A deadlock will not occur if any one of the below four conditions are prevented:

1. **hold and wait**
2. **mutual exclusion**
3. **circular wait**

4. **no-preemption**

Now,

Option-1 if implemented violates 1 so deadlock cannot occur.

Option-2 if implemented violates circular wait (making the dependency graph acyclic)

Option-3 if implemented violates circular wait (making the dependency graph acyclic)

Option-4 it is equivalent to options 2 and 3

So, the correct option is 4 as all of them are methods to prevent deadlock.

http://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/7_Deadlocks.html

References

69 votes

-- Tamojit Chatterjee (1.9k points)

5.17.23 Resource Allocation: GATE CSE 2016 Set 1 | Question: 50 top

https://gateoverflow.in/39719

Answer is (**A**).

```
while (t[j] != 0 && t[j] <=t[i]);
```

This ensures that when a process i reaches Critical Section, all processes j which started before it must have its $t[j] = 0$. This means no two process can be in critical section at same time as one of them must be started earlier.

> returns an integer not smaller

is the issue here for deadlock. This means two processes can have same t value and hence

```
while (t[j] != 0 && t[j] <=t[i]);
```

can go to infinite wait. ($t[j] == t[i]$). Starvation is also possible as there is nothing to ensure that a request is granted in a timed manner. But bounded waiting (as defined in Galvin) is guaranteed here as when a process i starts and gets $t[i]$ value, no new process can enter critical section before i (as their t value will be higher) and this ensures that access to critical section is granted only to a finite number of processes (those which started before) before eventually process i gets access.

But in some places bounded waiting is defined as finite waiting (see one here from CMU) and since deadlock is possible here, bounded waiting is not guaranteed as per that definition.

References

78 votes

-- Arjun Suresh (332k points)

Given question is a wrongly modified version of actual bakery algorithm, used for N-process critical section problem.

Bakery algorithm code goes as follows : (as in William stalling book page 209, 7th edition)

```
Entering[i] = true;
Number[i] = 1 + max(Number[1], ..., Number[NUM_THREADS]);
Entering[i] = false;

for (integer j = 1; j <= NUM_THREADS; j++) {
    // Wait until thread j receives its number:
    while (Entering[j]) {
      /* nothing */
    }

    // Wait until all threads with smaller numbers or with the same
```

```
    // number, but with higher priority, finish their work:
    while ((Number[j] != 0) && ((Number[j], j) < (Number[i], i))) {
      /* nothing */
    }
}

    <Critical Section>

    Number[i] = 0;
    /*remainder section */
```

code explanation:

The important point here is that due to lack of atomicity of `max` function multiple processes may calculate the same `Number`.
In that situation to choose between two processes, we prioritize the lower `process_id`.

`(Number[j], j) < (Number[i], i))` this is a tuple comparison and it allows us to correctly select only one process out of `i` and `j`.but not both (when `Number[i] = Number[j]`)

Progress and Deadlock:

The testing condition given in the question is `while (t[j] != 0 && t[j] <=t[i]);` which creates deadlock for both `i` and `j` (and possibly more) processes which have calculated their Numbers as the same value. C and D are wrong.

Bounded waiting :

If the process `i` is waiting and looping inside the for loop. Why is it waiting there ? Two reasons,

1. Its number value is not yet the minimum positive value.
2. Or, its Number value is equal to some other's Number value.

Reason1 does not dissatisfy bounded waiting , because if the process `i` has the Number value = 5 then all processes having less positive `Number` will enter CS first and will exit. Then Process `i` will definitely get a chance to enter into CS.

Reason2 dissatisfy bounded waiting because assume process 3 and 4 are fighting with the equal Number value of 5. whenever one of them (say 4) is scheduled by the short term scheduler to the CPU, it goes on looping on $Number[3] \Leftarrow Number[4]$ .Similarly with process 3 also. But when they are removed from the Running state by the scheduler , other processes may continue normal operation. So for process 3 and 4 although they have requested very early, because of their own reason, other processes are getting a chance of entering into CS. B is wrong.

note : in this all the processes go into deadlock anyway after a while.

How mutual exclusion is satisfied ?

Now we assume all processes calculate their Number value as distinct.
And categorize all concurrent N processes into three groups;

1. Processes which are now testing the while condition inside the for loop.
2. Processes which are now in the reminder section.
3. Processes which are now about to calculate its Number values.

In *Category 1,* assume process `i` wins the testing condition, that means no one else can win the test because `i` has the lowest positive value among the 1st category of processes.

Category 3 processes will calculate Number value more than the Number of `i` using `max` the function.
Same goes with *Category 2* processes if they ever try to re-enter.

---

detail of bakery algorithm Link1 and Link2 and Link3_page53

References

👍 45 votes

-- Debashish Deka (40.8k points)

5.17.24 Resource Allocation: GATE CSE 2017 Set 2 | Question: 33 top

https://gateoverflow.in/118375

✔

Process	Current Allocation	Max Requirement	Need
P1	3	7	4
P2	1	6	5
P3	3	5	2

Given there are total 9 tape drives,

So, according to the above table we can see we have currently allocated (7 tape drive), so **currently Available tape drives** $= 2$

So, $P3$ can use it and after using it will release it 3 resources **New Available** $= 5$

then $P1$ can use it and will release it 3 resources so **New Available** $= 8$

and lastly $P2$ so, all the process are in **SAFE STATE** and there will be **NO DEADLOCK**

Safe Sequence will be $\mathbf{P3} \rightarrow \mathbf{P2} \rightarrow \mathbf{P1}$ or $\mathbf{P3} \rightarrow \mathbf{P1} \rightarrow \mathbf{P2.}$

Answer will be (B) only.

👍 42 votes -- Abhishek Mitra (509 points)

5.17.25 Resource Allocation: GATE IT 2005 | Question: 62 top

https://gateoverflow.in/3823

✔ If any process has highest priority over all the resources then it can snatch any resource from any other process and so no deadlock can occur with another process as this highest priority process will eventually finish and release all the resources for the other less priority process.

In case of (I) and (II) process 1 has given highest priority over all the resources and hence deadlock cannot occur.

Similarly, in the case of (III) and (IV) process 2 has given highest priority over all the resources and hence deadlock cannot occur.

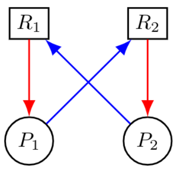
If we consider option (A) (I) and (IV)

- $T_{11} > T_{21}$ // for resource 1, process 1 has the highest priority
- $T_{22} > T_{12}$ // for resource 2 , process 2 has highest priority

Let P_1 be holding R_1 and waiting for R_2.

Let P_2 be holding R_2 and waiting for R_1.

This is deadlock as neither is releasing its held resources.



Similarly in option B also deadlock can occur.

Correct **answer : C**

👍 27 votes -- Dharmendra Lodhi (2.7k points)

5.17.26 Resource Allocation: GATE IT 2008 | Question: 54 top

https://gateoverflow.in/3364

✔ Answer: (**B**)

Starvation can occur as each time a process requests a resource it has to release all its resources. Now, maybe the process has not used the resources properly yet. This will happen again when the process requests another resource. So, the process starves for proper utilisation of resources.

Deadlock will not occur as it is similar to a deadlock prevention scheme.

👍 31 votes -- Rajarshi Sarkar (27.9k points)

5.18 Runtime Environments (3) top

5.18.1 Runtime Environments: GATE CSE 1991 | Question: 02-iii top

https://gateoverflow.in/513

Match the pairs in the following questions by writing the corresponding letters only.

(a)	Buddy system	(p)	Run time type specification
(b)	Interpretation	(q)	Segmentation
(c)	Pointer type	(r)	Memory allocation
(d)	Virtual memory	(s)	Garbage collection

gate1991 operating-system normal match-the-following runtime-environments

Answer

5.18.2 Runtime Environments: GATE CSE 1996 | Question: 2.17 top

https://gateoverflow.in/2746

The correct matching for the following pairs is

(A)	Activation record	(1)	Linking loader
(B)	Location counter	(2)	Garbage collection
(C)	Reference counts	(3)	Subroutine call
(D)	Address relocation	(4)	Assembler

A. A-3 B-4 C-1 D-2
B. A-4 B-3 C-1 D-2
C. A-4 B-3 C-2 D-1
D. A-3 B-4 C-2 D-1

gate1996 operating-system easy runtime-environments

Answer

5.18.3 Runtime Environments: GATE CSE 2002 | Question: 2.20 top

https://gateoverflow.in/850

Dynamic linking can cause security concerns because

A. Security is dynamic
B. The path for searching dynamic libraries is not known till runtime
C. Linking is insecure
D. Cryptographic procedures are not available for dynamic linking

gate2002-cse operating-system runtime-environments easy

Answer

Answers: Runtime Environments

5.18.1 Runtime Environments: GATE CSE 1991 | Question: 02-iii top

https://gateoverflow.in/513

✔ $(a)-(r), (b)-(p), (c)-(s), (d)-(q)$

21 votes

-- Gate Keeda (15.9k points)

5.18.2 Runtime Environments: GATE CSE 1996 | Question: 2.17 top

https://gateoverflow.in/2746

✔ **(D) Option**

Each time a sub routine is called, its activation record is created.

An assembler uses location counter value to give address to each instruction which is needed for relative addressing as well as for jump labels.

Reference count is used by garbage collector to clear the memory whose reference count becomes 0.

Linker Loader is a loader which can load several compiled codes and link them together into a single executable. Thus it needs to do relocation of the object codes.

👍 56 votes

-- Arjun Suresh (332k points)

5.18.3 Runtime Environments: GATE CSE 2002 | Question: 2.20 top

☛ https://gateoverflow.in/850

✔

A. Nonsense option, No idea why it is here.
B. The path for searching dynamic libraries is not known till runtime -> This seems most correct answer.
C. This is not true. Linking in itself not insecure.
D. There is no relation between Cryptographic procedures & Dynamic linking.

👍 46 votes

-- Akash Kanase (36k points)

5.19 Semaphores (8) top

5.19.1 Semaphores: GATE CSE 1990 | Question: 1-vii top

☛ https://gateoverflow.in/83851

Semaphore operations are atomic because they are implemented within the OS _________.

gate1990 operating-system semaphores process-synchronization fill-in-the-blanks

Answer

5.19.2 Semaphores: GATE CSE 1992 | Question: 02,x, ISRO2015-35 top

☛ https://gateoverflow.in/564

At a particular time of computation, the value of a counting semaphore is 7. Then 20 P operations and 15 V operations were completed on this semaphore. The resulting value of the semaphore is :

A. 42
B. 2
C. 7
D. 12

gate1992 operating-system semaphores easy isro2015 multiple-selects process-synchronization

Answer

5.19.3 Semaphores: GATE CSE 1998 | Question: 1.31 top

☛ https://gateoverflow.in/1668

A counting semaphore was initialized to 10. Then $6P$ (wait) operations and $4V$ (signal) operations were completed on this semaphore. The resulting value of the semaphore is

A. 0
B. 8
C. 10
D. 12

gate1998 operating-system process-synchronization semaphores easy

Answer

5.19.4 Semaphores: GATE CSE 2008 | Question: 63 top

☛ https://gateoverflow.in/486

The P and V operations on counting semaphores, where s is a counting semaphore, are defined as follows:

$P(s)$: $s = s - 1$;
If $s < 0$ then wait;

$V(s)$: $s = s + 1$;
If $s \leq 0$ then wake up process waiting on s;

Assume that P_b and V_b the wait and signal operations on binary semaphores are provided. Two binary semaphores x_b and y_b are used to implement the semaphore operations $P(s)$ and $V(s)$ as follows:

$$P(s): \quad \begin{array}{l} P_b(x_b); \\ s = s - 1; \\ \text{if } (s < 0) \\ \{ \\ \quad V_b(x_b); \\ \quad P_b(y_b); \\ \} \\ \ \text{else } V_b(x_b); \end{array}$$

$$V(s): \quad \begin{array}{l} P_b(x_b); \\ s = s + 1; \\ \text{if } (s \leq 0) V_b(y_b); \\ V_b(x_b); \end{array}$$

The initial values of x_b and y_b are respectively

A. 0 and 0
B. 0 and 1
C. 1 and 0
D. 1 and 1

gate2008-cse operating-system normal semaphores

Answer

5.19.5 Semaphores: GATE CSE 2016 Set 2 | Question: 49 top https://gateoverflow.in/39576

Consider a non-negative counting semaphore S. The operation $P(S)$ decrements S, and $V(S)$ increments S. During an execution, 20 $P(S)$ operations and 12 $V(S)$ operations are issued in some order. The largest initial value of S for which at least one $P(S)$ operation will remain blocked is _______

gate2016-cse-set2 operating-system semaphores normal numerical-answers

Answer

5.19.6 Semaphores: GATE CSE 2020 | Question: 34 top https://gateoverflow.in/333197

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 no process executes CODE SECTION Q before every process has finished CODE SECTION P.
B. It ensures that two processes are in CODE SECTION Q at any time.
C. It ensures that all processes execute CODE SECTION P mutually exclusively.
D. It ensures that at most $n-1$ processes are in CODE SECTION P at any time.

gate2020-cse operating-system semaphores

Answer

5.19.7 Semaphores: GATE CSE 2021 Set 1 | Question: 46 top https://gateoverflow.in/357405

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);
```

```
        wait(S);
        counter++;
        signal(S);
        signal(S);
}
```

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 the 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

gate2021-cse-set1 multiple-selects operating-system process-synchronization semaphores

Answer

5.19.8 Semaphores: GATE IT 2006 | Question: 57 top

https://gateoverflow.in/3601

The wait and signal operations of a monitor are implemented using semaphores as follows. In the following,

- x is a condition variable,
- mutex is a semaphore initialized to 1,
- x_sem is a semaphore initialized to 0,
- x_count is the number of processes waiting on semaphore x_sem, initially 0,
- next is a semaphore initialized to 0,
- next_count is the number of processes waiting on semaphore next, initially 0.

The body of each procedure that is visible outside the monitor is replaced with the following:

```
P(mutex);
...
body of procedure
...
if (next_count > 0)
    V(next);
else
    V(mutex);
```

Each occurrence of x.wait is replaced with the following:

```
x_count = x_count + 1;
if (next_count > 0)
    V(next);
else
    V(mutex);
------------------------------------------------------------ E1;
x_count = x_count - 1;
```

Each occurrence of x.signal is replaced with the following:

```
if (x_count > 0)
{
    next_count = next_count + 1;
    ------------------- E2;
    P(next);
    next_count = next_count - 1;
}
```

For correct implementation of the monitor, statements $E1$ and $E2$ are, respectively,

A. $P(x\_sem), V(next)$
B. $V(next), P(x\_sem)$
C. $P(next), V(x\_sem)$
D. $P(x\_sem), V(x\_sem)$

gate2006-it operating-system process-synchronization semaphores normal

Answer

Answers: Semaphores

5.19.1 Semaphores: GATE CSE 1990 | Question: 1-vii top https://gateoverflow.in/83851

✔ The concept of semaphores is used for synchronization.
Semaphore is an integer with a difference. Well, actually a few differences.
You set the value of the integer when you create it, but can never access the value directly after that; you must use one of the semaphore functions to adjust it, and you cannot ask for the current value.
There are semaphore functions to increment or decrement the value of the integer by one.
Decrementing is a (possibly) blocking function. If the resulting semaphore value is negative, the calling thread or process is blocked, and cannot continue until some other thread or process increments it.
Incrementing the semaphore when it is negative causes one (and only one) of the threads blocked by this semaphore to become unblocked and runnable.

Therefore, all semaphore operations are atomic. Implemented in kernel,

24 votes -- Neeraj7375 (1.1k points)

5.19.2 Semaphores: GATE CSE 1992 | Question: 02,x, ISRO2015-35 top https://gateoverflow.in/564

✔ The answer is option **B.**

Currently semaphore is 7 so, after 20 P(wait) operation it will come to -13 then for 15 V(signal) operation the value comes to 2.

32 votes -- sanjeev_zerocode (295 points)

5.19.3 Semaphores: GATE CSE 1998 | Question: 1.31 top https://gateoverflow.in/1668

✔ Answer is option **(B)**

Initially semaphore is 10, then 6 down operations are performed means $(10-6=4)$ and 4 up operations means $(4+4=8)$

So, at last option **(B)** 8 is correct.

29 votes -- Kalpna Bhargav (2.5k points)

5.19.4 Semaphores: GATE CSE 2008 | Question: 63 top https://gateoverflow.in/486

✔ Answer is (**C**) .

Reasoning :-

First let me explain what is counting semaphore & How it works. Counting semaphore gives count, i.e. no of processes that can be in Critical section at same time. Here value of S denotes that count. So suppose $S=3$, we need to be able to have 3 processes in Critical section at max. Also when counting semaphore S has negative value we need to have Absolute value of S as no of processes waiting for critical section.

(A) & (B) are out of option, because Xb must be 1, otherwise our counting semaphore will get blocked without doing anything. Now consider options (C) & (D).

Option (D) :-

$Yb=1, Xb=1$

Assume that initial value of $S=2$. (At max 2 processes must be in Critical Section.)

We have 4 processes, $P1, P2, P3 \& P4$.

$P1$ enters critical section , It calls $P(s), S=S-1=1$. As $S>1$, we do not call $Pb(Yb)$.

$P2$ enters critical section , It calls $P(s), S=S-1=0$. As $S>0$ we do not call $Pb(Yb)$.

Now $P3$ comes, it should be blocked but when it calls $P(s), S=S-1=0-1=-1$ As $S<0$,Now we do call $Pb(Yb)$. Still $P3$ enters into critical section & We do not get blocked as Yb's Initial value was 1.

This violates property of counting semaphore. S is now -1, & No process is waiting. Also we are allowing 1 more process than what counting semaphore permits.

If Yb would have been 0, $P3$ would have been blocked here & So Answer is (**C**).

$Pb(yb)$;

👍 132 votes -- Akash Kanase (36k points)

5.19.5 Semaphores: GATE CSE 2016 Set 2 | Question: 49 top https://gateoverflow.in/39576

✔ Answer: (7). Take any sequence of $20P$ and $12V$ operations, atleast one process will always remain blocked.

👍 29 votes -- Ashish Deshmukh (1.3k points)

5.19.6 Semaphores: GATE CSE 2020 | Question: 34 top https://gateoverflow.in/333197

✔ Answer: A. It ensures that no process executes CODE SECTION Q before every process has finished CODE SECTION P.

Explanation
In short, semaphore 'a' controls mutually exclusive execution of statement count+=1 and semaphore 'b' controls entry to CODE SECTION Q when all the process have executed CODE SECTION P. As checked by given condition if(count==n) signal(b); the semaphore 'b' is initialized to 0 and only increments when this condition is TRUE. (Side fact, processes do not enter the CODE SECTION Q in mutual exclusion, the moment **all** have executed CODE SECTION P, process will enter CODE SECTION Q in any order.)

Detailed explanation:-
Consider this situation as the processes need to execute three stages- Section P, then the given code and finally Section Q.

It is evident that semaphores do not control Section P hence, **There is no restriction in execution of P.**

Now, we are given 2 semaphores 'a' and 'b' initialized to '1' and '0' respectively.

Take an example of 3 processes (hence n=3, count=0(initially)) and lets say first of them has finished executing Section P and enters the given code. It does following changes:-
1. will execute wait(a) hence making semaphore a=0
2. increment the count from 0 to 1 (first time)
3. **If(count==n) evaluates FALSE and hence signal(b) is not executed.** So semaphore b remains 0
4. signal(a) hence making semaphore a=1
5. wait(b) But since semaphore b is already 0, **The process will be in blocked/waiting state.**
First out of the three processes is unable to enter the CODE SECTION Q !

Now say second process completes CODE SECTION P and starts executing the given code. It can be concluded that it will follow the same sequence (5 steps) as mentioned above and status of variables will be:- count = 2 (still count<n), semaphore a=1, semaphore b=0 (no change)

Finally the last process finishes execution of CODE SECTION P.
It will follow same steps 1 and 2 making semaphore a=0 and **count = 3**
3. if(count==n) evalueates TRUE! and hence signal(b) is executed marking semaphore b = 1 FOR THE FIRST TIME.
4 and 5 will be executed the same way.

Now the moment this last process signaled b, the previously blocked process will be able to execute wait(b) and the very next moment execute signal(b) to allow other blocked/waiting process to proceed.
This way all the processes enter CODE SECTION Q after executing CODE SECTION P.

👍 17 votes -- dhruvhacks (609 points)

5.19.7 Semaphores: GATE CSE 2021 Set 1 | Question: 46 top https://gateoverflow.in/357405

✔ **Correct Options: A,B,D**

The given code allows up to 2 threads to be in the critical section as the initial value of semaphore is 5 and 2 wait operations are necessary to enter the critical section ($\lceil 5/2 \rceil = 2$).

In the critical section the increment operation is not atomic. So, multiple threads entering the critical section simultaneously can cause race condition.

A. Assume that the 5 threads execute sequentially with no interleaving then after each thread ends the counter value increments by 1. Hence after 5 threads finish, counter value will be incremented 5 times from 0 to 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**

4 votes -- Cringe is my middle name... (885 points)

5.19.8 Semaphores: GATE IT 2006 | Question: 57 top https://gateoverflow.in/3601

✔

- x_count is the number of processes waiting on semaphore x_sem, initially 0,

x_count is incremented and decremented in x.wait, which shows that in between them wait(x_sem) must happen which is P(x_sem). Correspondingly V(x_sem) must happen in x.signal. So, D choice.

What is a monitor?

References

18 votes -- Arjun Suresh (332k points)

5.20 System Call (1) top

5.20.1 System Call: GATE CSE 2021 Set 1 | Question: 14 top https://gateoverflow.in/357438

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. exit
B. malloc
C. sleep
D. strlen

gate2021-cse-set1 multiple-selects operating-system system-call

Answer

Answers: System Call

5.20.1 System Call: GATE CSE 2021 Set 1 | Question: 14 top https://gateoverflow.in/357438

✔ System calls are used to get some service from operating system which generally requires some higher level of privilege.

This question uses two important words “always” and “standard C library functions”.
Let’s check options

1. **exit**- This is a function defined in standard C library and it **always** invokes system call every time, flushes the streams, and terminates the caller.
2. **malloc** – This is a function defined in standard C library and it ***does not always*** invoke the system call. When a process is created, certain amount of heap memory is already allocated to it, when required to expand or shrink that memory, it internally uses sbrk/brk system call on Unix/Linux. i.e., **not every malloc** call needs a system call but if the current allocated size is not enough, it’ll do a system call to get more memory.
3. **sleep**- This is not even standard C library function, it is a POSIX standard C library function. Unix and Windows uses different header files for it. Now as question has said the *following standard C library* function, let’s consider it as that way. Yes it **always** invokes the system call .
4. **strlen** – This is a function defined in standard C library and **doesn’t** require any system call to perform its function of calculating the string length.

Answer : A,C

4 votes -- Persistent (89 points)

5.21 Threads (8) top

5.21.1 Threads: GATE CSE 2004 | Question: 11 top

https://gateoverflow.in/1008

Consider the following statements with respect to user-level threads and kernel-supported threads

I. context switch is faster with kernel-supported threads
II. for user-level threads, a system call can block the entire process
III. Kernel supported threads can be scheduled independently
IV. User level threads are transparent to the kernel

Which of the above statements are true?

A. (II), (III) and (IV) only
B. (II) and (III) only
C. (I) and (III) only
D. (I) and (II) only

gate2004-cse operating-system threads normal

Answer

5.21.2 Threads: GATE CSE 2007 | Question: 17 top

https://gateoverflow.in/1215

Consider the following statements about user level threads and kernel level threads. Which one of the following statements is FALSE?

A. Context switch time is longer for kernel level threads than for user level threads.
B. User level threads do not need any hardware support.
C. Related kernel level threads can be scheduled on different processors in a multi-processor system.
D. Blocking one kernel level thread blocks all related threads.

gate2007-cse operating-system threads normal

Answer

5.21.3 Threads: GATE CSE 2011 | Question: 16, UGCNET-June2013-III: 65 top

https://gateoverflow.in/2118

A thread is usually defined as a light weight process because an Operating System (OS) maintains smaller data structure for a thread than for a process. In relation to this, which of the following statement is correct?

A. OS maintains only scheduling and accounting information for each thread
B. OS maintains only CPU registers for each thread
C. OS does not maintain virtual memory state for each thread
D. OS does not maintain a separate stack for each thread

gate2011-cse operating-system threads normal ugcnetjune2013iii

Answer

5.21.4 Threads: GATE CSE 2014 Set 1 | Question: 20 top

https://gateoverflow.in/1787

Which one of the following is **FALSE**?

A. User level threads are not scheduled by the kernel.
B. When a user level thread is blocked, all other threads of its process are blocked.
C. Context switching between user level threads is faster than context switching between kernel level threads.
D. Kernel level threads cannot share the code segment.

gate2014-cse-set1 operating-system threads normal

Answer

5.21.5 Threads: GATE CSE 2017 Set 1 | Question: 18 top

https://gateoverflow.in/118298

Threads of a process share

A. global variables but not heap
B. heap but not global variables
C. neither global variables nor heap
D. both heap and global variables

gate2017-cse-set1 operating-system threads

Answer

5.21.6 Threads: GATE CSE 2017 Set 2 | Question: 07 top

https://gateoverflow.in/118240

Which of the following is/are shared by all the threads in a process?

I. Program counter
II. Stack
III. Address space
IV. Registers

A. (I) and (II) only
B. (III) only
C. (IV) only
D. (III) and (IV) only

gate2017-cse-set2 operating-system threads

Answer

5.21.7 Threads: GATE CSE 2021 Set 2 | Question: 42 top

https://gateoverflow.in/357498

Consider the following multi-threaded code segment (in a mix of C and pseudo-code), invoked by two processes P_1 and P_2, and each of the processes spawns two threads T_1 and T_2:

```
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. Both P_1 and P_2 will print the value of x as 2.
B. At least of P_1 and P_2 will print the value of x as 4.
C. At least one of the threads will print the value of y as 2.
D. Both T_1 and T_2, in both the processes, will print the value of y as 1.

gate2021-cse-set2 multiple-selects operating-system threads

Answer

5.21.8 Threads: GATE IT 2004 | Question: 14 top

https://gateoverflow.in/3655

Which one of the following is NOT shared by the threads of the same process ?

A. Stack
B. Address Space
C. File Descriptor Table
D. Message Queue

gate2004-it operating-system easy threads

Answer

Answers: Threads

5.21.1 Threads: GATE CSE 2004 | Question: 11

https://gateoverflow.in/1008

✔ Answer: (**A**)

I. User level thread switching is faster than kernel level switching. So, (I) is false.
II. is true.
III. is true.
IV. User level threads are transparent to the kernel
In case of Computing transparent means functioning without being aware. In our case user level threads are functioning without kernel being aware about them. So (IV) is actually correct.

61 votes

-- Akash Kanase (36k points)

User level threads can switch almost as fast as a procedure call. Kernel supported threads switch much slower. So, I is false.

II, III and IV are TRUE. So A.

"The kernel knows nothing about user-level threads and manages them as if they were single-threaded processes"

Ref: http://stackoverflow.com/questions/15983872/difference-between-user-level-and-kernel-supported-threads

References

32 votes

-- Arjun Suresh (332k points)

5.21.2 Threads: GATE CSE 2007 | Question: 17

https://gateoverflow.in/1215

✔ Answer: (**D**)

A. Context switch time is longer for kernel level threads than for user level threads. – This is True, as Kernel level threads are managed by OS and Kernel maintains lot of data structures. There are many overheads involved in Kernel level thread management, which are not present in User level thread management !

B. User level threads do not need any hardware support.– This is true, as User level threads are implemented by Libraries programmably, Kernel does not sees them.

C. Related kernel level threads can be scheduled on different processors in a multi-processor system.– This is true.

D. Blocking one kernel level thread blocks all related threads. – This is false. If it had been user Level threads this would have been true, (In One to one, or many to one model !) Kernel level threads are independent.

50 votes

-- Akash Kanase (36k points)

5.21.3 Threads: GATE CSE 2011 | Question: 16, UGCNET-June2013-III: 65

https://gateoverflow.in/2118

✔ Answer to this question is (**C**).

Many of you would not agree at first So here I explain it how.

OS , on per thread basis, maintains ONLY TWO things : CPU Register state and Stack space. It does not maintain anything else for individual thread. Code segment and Global variables are shared. Even TLB and Page Tables are also shared since they belong to same process.

A. option (A) would have been correct if 'ONLY' word were not there. It NOT only maintains register state BUT stack space also.

B. is obviously FALSE
C. is TRUE as it says that OS does not maintain VIRTUAL Memory state for individual thread which isTRUE
D. This is also FALSE.

83 votes -- Sandeep_Uniyal (6.5k points)

5.21.4 Threads: GATE CSE 2014 Set 1 | Question: 20 top https://gateoverflow.in/1787

✔ **(D)** is the answer. Threads can share the Code segments. They have only separate Registers and stack.

User level threads are scheduled by the thread library and kernel knows nothing about it. So, **A** is TRUE.

When a user level thread is blocked, all other threads of its process are blocked. So, **B** is TRUE. (With a multi-threaded kernel, user level threads can make non-blocking system calls without getting blocked. But in this option, it is explicitly said 'a thread is blocked'.)

Context switching between user level threads is faster as they actually have no context-switch- nothing is saved and restored while for kernel level thread, Registers, PC and SP must be saved and restored. So, **C** also TRUE.

Reference: http://www.cs.cornell.edu/courses/cs4410/2008fa/homework/hw1_soln.pdf

References

49 votes -- Sandeep_Uniyal (6.5k points)

5.21.5 Threads: GATE CSE 2017 Set 1 | Question: 18 top https://gateoverflow.in/118298

✔ A thread shares with other threads a process's (to which it belongs to) :

- Code section
- Data section (static + heap)
- Address Space
- Permissions
- Other resources (e.g. files)

Therefore, (**D**) is the answer.

52 votes -- Kantikumar (3.4k points)

5.21.6 Threads: GATE CSE 2017 Set 2 | Question: 07 top https://gateoverflow.in/118240

✔ Thread is light weight process, and every thread have its own, stack, register, and PC (one of the register in CPU contain address of next instruction to be executed), so only address space that is shared by all thread for a single process.
So, option (**B**) is correct answer.

35 votes -- 2018 (5.5k points)

5.21.7 Threads: GATE CSE 2021 Set 2 | Question: 42 top https://gateoverflow.in/357498

✔ Each process has its own address space.

1. P_1 :
   Two threads T_{11}, T_{12} are created in main.

   Both execute foo function and threads don't wait for each other. Due to explicit locking mechanism here mutual exclusion is there and hence no race condition inside foo().

   y being thread local, both the threads will print the value of y as 1.

   Due to the wait in main, the print(x) will happen only after both the threads finish. So, x will have become 2.

   PS: Even if x was not assigned 0 explicitly in C all global and static variables are initialized to 0 value.

2. P_2 :

Same thing happens here as P_1 as this is a different process. For sharing data among different processes mechanisms like shared memory, files, sockets etc must be used.

So, the correct answer here is A and D.

- Suppose wait is removed from the main(). Then the possible x values can be $0, 1, 2$ as the main thread as well as the two created threads can execute in any order.
- Suppose locking mechanism is removed from foo() and assignments are not atomic. (If increment is atomic here, then locking is not required). Then race condition can happen and so one of the increments can overwrite the other. So, in main, x value printed can be either 1 or 2.
- Now suppose we had just one process which does a fork() inside main before creating the threads. How the answer should change?

👍 6 votes

-- Arjun Suresh (332k points)

5.21.8 Threads: GATE IT 2004 | Question: 14 top

https://gateoverflow.in/3655

✔ Stack is not shared

👍 29 votes

-- Sankaranarayanan P.N (8.5k points)

5.22 Virtual Memory (39) top

5.22.1 Virtual Memory: GATE CSE 1989 | Question: 2-iv top

https://gateoverflow.in/87081

Match the pairs in the following:

(A)	Virtual memory	(p)	Temporal Locality
(B)	Shared memory	(q)	Spatial Locality
(C)	Look-ahead buffer	(r)	Address Translation
(D)	Look-aside buffer	(s)	Mutual Exclusion

match-the-following gate1989 operating-system virtual-memory

Answer

5.22.2 Virtual Memory: GATE CSE 1990 | Question: 1-v top

https://gateoverflow.in/83833

Under paged memory management scheme, simple lock and key memory protection arrangement may still be required if the __________ processors do not have address mapping hardware.

gate1990 operating-system virtual-memory fill-in-the-blanks

Answer

5.22.3 Virtual Memory: GATE CSE 1990 | Question: 7-b top

https://gateoverflow.in/85404

In a two-level virtual memory, the memory access time for main memory, $t_M = 10^{-8}$ sec, and the memory access time for the secondary memory, $t_D = 10^{-3}$ sec. What must be the hit ratio, H such that the access efficiency is within 80 percent of its maximum value?

gate1990 descriptive operating-system virtual-memory

Answer

5.22.4 Virtual Memory: GATE CSE 1991 | Question: 03-xi top

https://gateoverflow.in/525

Indicate all the false statements from the statements given below:

A. The amount of virtual memory available is limited by the availability of the secondary memory
B. Any implementation of a critical section requires the use of an indivisible machine- instruction ,such as test-and-set.

C. The use of monitors ensure that no dead-locks will be caused .
D. The LRU page-replacement policy may cause thrashing for some type of programs.
E. The best fit techniques for memory allocation ensures that memory will never be fragmented.

gate1991 operating-system virtual-memory normal multiple-selects

Answer

5.22.5 Virtual Memory: GATE CSE 1994 | Question: 1.21 top

https://gateoverflow.in/2464

Which one of the following statements is true?

A. Macro definitions cannot appear within other macro definitions in assembly language programs
B. Overlaying is used to run a program which is longer than the address space of a computer
C. Virtual memory can be used to accommodate a program which is longer than the address space of a computer
D. It is not possible to write interrupt service routines in a high level language

gate1994 operating-system normal virtual-memory

Answer

5.22.6 Virtual Memory: GATE CSE 1995 | Question: 1.7 top

https://gateoverflow.in/2594

In a paged segmented scheme of memory management, the segment table itself must have a page table because

A. The segment table is often too large to fit in one page
B. Each segment is spread over a number of pages
C. Segment tables point to page tables and not to the physical locations of the segment
D. The processor's description base register points to a page table

gate1995 operating-system virtual-memory normal

Answer

5.22.7 Virtual Memory: GATE CSE 1995 | Question: 2.16 top

https://gateoverflow.in/2628

In a virtual memory system the address space specified by the address lines of the CPU must be _____ than the physical memory size and ____ than the secondary storage size.

A. smaller, smaller
B. smaller, larger
C. larger, smaller
D. larger, larger

gate1995 operating-system virtual-memory normal

Answer

5.22.8 Virtual Memory: GATE CSE 1996 | Question: 7 top

https://gateoverflow.in/2759

A demand paged virtual memory system uses 16 bit virtual address, page size of 256 bytes, and has 1 Kbyte of main memory. LRU page replacement is implemented using the list, whose current status (page number is decimal) is

17	1	63

LRU page

For each hexadecimal address in the address sequence given below,

$00FF, 010D, 10FF, 11B0$

indicate

i. the new status of the list
ii. page faults, if any, and
iii. page replacements, if any.

gate1996 operating-system virtual-memory normal descriptive

Answer

5.22.9 Virtual Memory: GATE CSE 1998 | Question: 2.18, UGCNET-June2012-III: 48 top

https://gateoverflow.in/1691

If an instruction takes i microseconds and a page fault takes an additional j microseconds, the effective instruction time if on the average a page fault occurs every k instruction is:

A. $i + \frac{j}{k}$
B. $i + (j \times k)$
C. $\frac{i+j}{k}$
D. $(i + j) \times k$

gate1998 operating-system virtual-memory easy ugcnetjune2012iii

Answer

5.22.10 Virtual Memory: GATE CSE 1999 | Question: 19 top

https://gateoverflow.in/1518

A certain computer system has the segmented paging architecture for virtual memory. The memory is byte addressable. Both virtual and physical address spaces contain 2^{16} bytes each. The virtual address space is divided into 8 non-overlapping equal size segments. The memory management unit (MMU) has a hardware segment table, each entry of which contains the physical address of the page table for the segment. Page tables are stored in the main memory and consists of 2 byte page table entries.

a. What is the minimum page size in bytes so that the page table for a segment requires at most one page to store it? Assume that the page size can only be a power of 2.

b. Now suppose that the pages size is 512 bytes. It is proposed to provide a TLB (Transaction look-aside buffer) for speeding up address translation. The proposed TLB will be capable of storing page table entries for 16 recently referenced virtual pages, in a fast cache that will use the direct mapping scheme. What is the number of tag bits that will need to be associated with each cache entry?

c. Assume that each page table entry contains (besides other information) 1 valid bit, 3 bits for page protection and 1 dirty bit. How many bits are available in page table entry for storing the aging information for the page? Assume that the page size is 512 bytes.

gate1999 operating-system virtual-memory normal descriptive

Answer

5.22.11 Virtual Memory: GATE CSE 1999 | Question: 2.10 top

https://gateoverflow.in/1488

A multi-user, multi-processing operating system cannot be implemented on hardware that does not support

A. Address translation
B. DMA for disk transfer
C. At least two modes of CPU execution (privileged and non-privileged)
D. Demand paging

gate1999 operating-system normal virtual-memory

Answer

5.22.12 Virtual Memory: GATE CSE 1999 | Question: 2.11 top https://gateoverflow.in/1489

Which of the following is/are advantage(s) of virtual memory?

A. Faster access to memory on an average.
B. Processes can be given protected address spaces.
C. Linker can assign addresses independent of where the program will be loaded in physical memory.
D. Program larger than the physical memory size can be run.

gate1999 operating-system virtual-memory easy

Answer

5.22.13 Virtual Memory: GATE CSE 2000 | Question: 2.22 top https://gateoverflow.in/669

Suppose the time to service a page fault is on the average 10 milliseconds, while a memory access takes 1 microsecond. Then a 99.99% hit ratio results in average memory access time of

A. 1.9999 milliseconds
B. 1 millisecond
C. 9.999 microseconds
D. 1.9999 microseconds

gate2000-cse operating-system easy virtual-memory

Answer

5.22.14 Virtual Memory: GATE CSE 2001 | Question: 1.20 top https://gateoverflow.in/713

Where does the swap space reside?

A. RAM
B. Disk
C. ROM
D. On-chip cache

gate2001-cse operating-system easy virtual-memory

Answer

5.22.15 Virtual Memory: GATE CSE 2001 | Question: 1.8 top https://gateoverflow.in/701

Which of the following statements is false?

A. Virtual memory implements the translation of a program's address space into physical memory address space
B. Virtual memory allows each program to exceed the size of the primary memory
C. Virtual memory increases the degree of multiprogramming
D. Virtual memory reduces the context switching overhead

gate2001-cse operating-system virtual-memory normal

Answer

5.22.16 Virtual Memory: GATE CSE 2001 | Question: 2.21 top https://gateoverflow.in/739

Consider a machine with 64 MB physical memory and a 32-bit virtual address space. If the page size s 4 KB, what is the approximate size of the page table?

A. 16 MB
B. 8 MB
C. 2 MB
D. 24 MB

gate2001-cse operating-system virtual-memory normal

Answer

5.22.17 Virtual Memory: GATE CSE 2002 | Question: 19 top

https://gateoverflow.in/872

A computer uses $32 - bit$ virtual address, and $32 - bit$ physical address. The physical memory is byte addressable, and the page size is 4 Kbytes. It is decided to use two level page tables to translate from virtual address to physical address. Equal number of bits should be used for indexing first level and second level page table, and the size of each table entry is 4 bytes.

A. Give a diagram showing how a virtual address would be translated to a physical address.
B. What is the number of page table entries that can be contained in each page?
C. How many bits are available for storing protection and other information in each page table entry?

gate2002-cse operating-system virtual-memory normal descriptive

Answer

5.22.18 Virtual Memory: GATE CSE 2003 | Question: 26 top

https://gateoverflow.in/916

In a system with 32 bit virtual addresses and 1 KB page size, use of one-level page tables for virtual to physical address translation is not practical because of

A. the large amount of internal fragmentation
B. the large amount of external fragmentation
C. the large memory overhead in maintaining page tables
D. the large computation overhead in the translation process

gate2003-cse operating-system virtual-memory normal

Answer

5.22.19 Virtual Memory: GATE CSE 2003 | Question: 78 top

https://gateoverflow.in/788

A processor uses $2 - level$ page tables for virtual to physical address translation. Page tables for both levels are stored in the main memory. Virtual and physical addresses are both 32 bits wide. The memory is byte addressable. For virtual to physical address translation, the 10 most significant bits of the virtual address are used as index into the first level page table while the next 10 bits are used as index into the second level page table. The 12 least significant bits of the virtual address are used as offset within the page. Assume that the page table entries in both levels of page tables are 4 bytes wide. Further, the processor has a translation look-aside buffer (TLB), with a hit rate of 96%. The TLB caches recently used virtual page numbers and the corresponding physical page numbers. The processor also has a physically addressed cache with a hit rate of 90%. Main memory access time is 10 ns, cache access time is 1 ns, and TLB access time is also 1 ns.

Assuming that no page faults occur, the average time taken to access a virtual address is approximately (to the nearest 0.5 ns)

A. 1.5 ns
B. 2 ns
C. 3 ns
D. 4 ns

gate2003-cse operating-system normal virtual-memory

Answer

5.22.20 Virtual Memory: GATE CSE 2003 | Question: 79 top

https://gateoverflow.in/43578

A processor uses 2-level page tables for virtual to physical address translation. Page tables for both levels are stored in the main memory. Virtual and physical addresses are both 32 bits wide. The memory is byte addressable. For virtual to physical address translation, the 10 most significant bits of the virtual address are used as index into the first level page table while the next 10 bits are used as index into the second level page table. The 12 least significant bits of the virtual address are used as offset within the page. Assume that the page table entries in both levels of page tables are 4 bytes wide. Further, the processor has a translation look-aside buffer (TLB), with a hit rate of 96%. The TLB caches recently used virtual page numbers and the corresponding physical page numbers. The processor also has a physically addressed cache with a hit rate of 90%. Main memory access time is 10 ns, cache access time is 1 ns, and TLB access time is also 1 ns.

Suppose a process has only the following pages in its virtual address space: two contiguous code pages starting at virtual address $0x00000000$, two contiguous data pages starting at virtual address $0x00400000$, and a stack page starting at virtual address $0xFFFFF000$. The amount of memory required for storing the page tables of this process is

A. 8 KB
B. 12 KB
C. 16 KB
D. 20 KB

gate2003-cse operating-system normal virtual-memory

Answer

5.22.21 Virtual Memory: GATE CSE 2006 | Question: 62, ISRO2016-50 top

https://gateoverflow.in/1840

A CPU generates 32-bit virtual addresses. The page size is 4 KB. The processor has a translation look-aside buffer (TLB) which can hold a total of 128 page table entries and is 4-way set associative. The minimum size of the TLB tag is:

A. 11 bits
B. 13 bits
C. 15 bits
D. 20 bits

gate2006-cse operating-system virtual-memory normal isro2016

Answer

5.22.22 Virtual Memory: GATE CSE 2006 | Question: 63, UGCNET-June2012-III: 45 top

https://gateoverflow.in/1841

A computer system supports 32-bit virtual addresses as well as 32-bit physical addresses. Since the virtual address space is of the same size as the physical address space, the operating system designers decide to get rid of the virtual memory entirely. Which one of the following is true?

A. Efficient implementation of multi-user support is no longer possible
B. The processor cache organization can be made more efficient now
C. Hardware support for memory management is no longer needed
D. CPU scheduling can be made more efficient now

gate2006-cse operating-system virtual-memory normal ugcnetjune2012iii

Answer

5.22.23 Virtual Memory: GATE CSE 2008 | Question: 67 top

https://gateoverflow.in/490

A processor uses 36 bit physical address and 32 bit virtual addresses, with a page frame size of 4 Kbytes. Each page table entry is of size 4 bytes. A three level page table is used for virtual to physical address translation, where the virtual address is used as follows:

- Bits $30-31$ are used to index into the first level page table.
- Bits $21-29$ are used to index into the 2nd level page table.
- Bits $12-20$ are used to index into the 3rd level page table.
- Bits $0-11$ are used as offset within the page.

The number of bits required for addressing the next level page table(or page frame) in the page table entry of the first, second and third level page tables are respectively

A. 20,20,20
B. 24,24,24
C. 24,24,20
D. 25,25,24

gate2008-cse operating-system virtual-memory normal

Answer

5.22.24 Virtual Memory: GATE CSE 2009 | Question: 10 top

https://gateoverflow.in/1302

The essential content(s) in each entry of a page table is / are

A. Virtual page number

B. Page frame number

C. Both virtual page number and page frame number

D. Access right information

gate2009-cse operating-system virtual-memory easy

Answer

5.22.25 Virtual Memory: GATE CSE 2009 | Question: 34 top

https://gateoverflow.in/1320

A multilevel page table is preferred in comparison to a single level page table for translating virtual address to physical address because

A. It reduces the memory access time to read or write a memory location.

B. It helps to reduce the size of page table needed to implement the virtual address space of a process

C. It is required by the translation lookaside buffer.

D. It helps to reduce the number of page faults in page replacement algorithms.

gate2009-cse operating-system virtual-memory easy

Answer

5.22.26 Virtual Memory: GATE CSE 2011 | Question: 20, UGCNET-June2013-II: 48 top

https://gateoverflow.in/2122

Let the page fault service time be 10 milliseconds(ms) in a computer with average memory access time being 20 nanoseconds (ns). If one page fault is generated every 10^6 memory accesses, what is the effective access time for memory?

A. 21 ns
B. 30 ns
C. 23 ns
D. 35 ns

gate2011-cse operating-system virtual-memory normal ugcnetjune2013ii

Answer

5.22.27 Virtual Memory: GATE CSE 2013 | Question: 52 top

https://gateoverflow.in/379

A computer uses $46-bit$ virtual address, $32-bit$ physical address, and a three–level paged page table organization. The page table base register stores the base address of the first-level table $(T1)$, which occupies exactly one page. Each entry of $T1$ stores the base address of a page of the second-level table $(T2)$. Each entry of $T2$ stores the base address of a page of the third-level table $(T3)$. Each entry of $T3$ stores a page table entry (PTE). The PTE is 32 bits in size. The processor used in the computer has a 1 MB 16 way set associative virtually indexed physically tagged cache. The cache block size is 64 bytes.

What is the size of a page in KB in this computer?

A. 2
B. 4
C. 8
D. 16

gate2013-cse operating-system virtual-memory normal

Answer

5.22.28 Virtual Memory: GATE CSE 2013 | Question: 53 top

https://gateoverflow.in/43294

A computer uses $46-bit$ virtual address, $32-bit$ physical address, and a three–level paged page table organization. The page table base register stores the base address of the first-level table $(T1)$, which occupies exactly one page. Each entry of $T1$ stores the base address of a page of the second-level table $(T2)$. Each entry of $T2$ stores the base address of a page of the third-level table $(T3)$. Each entry of $T3$ stores a page table entry (PTE). The PTE is 32 bits in size. The processor used in the computer has a 1 MB 16 way set associative virtually indexed physically tagged cache. The cache block size is 64 bytes.

What is the minimum number of page colours needed to guarantee that no two synonyms map to different sets in the processor cache of this computer?

A. 2
B. 4
C. 8
D. 16

gate2013-cse normal operating-system virtual-memory

Answer

5.22.29 Virtual Memory: GATE CSE 2014 Set 3 | Question: 33 top

https://gateoverflow.in/2067

Consider a paging hardware with a TLB. Assume that the entire page table and all the pages are in the physical memory. It takes 10 milliseconds to search the TLB and 80 milliseconds to access the physical memory. If the TLB hit ratio is 0.6, the effective memory access time (in milliseconds) is _________.

gate2014-cse-set3 operating-system virtual-memory numerical-answers normal

Answer

5.22.30 Virtual Memory: GATE CSE 2015 Set 1 | Question: 12 top

https://gateoverflow.in/8186

Consider a system with byte-addressable memory, $32-bit$ logical addresses, 4 $kilobyte$ page size and page table entries of 4 $bytes$ each. The size of the page table in the system in $megabytes$ is________________.

gate2015-cse-set1 operating-system virtual-memory easy numerical-answers

Answer

5.22.31 Virtual Memory: GATE CSE 2015 Set 2 | Question: 25 top

https://gateoverflow.in/8120

A computer system implements a $40-bit$ virtual address, page size of 8 kilobytes, and a $128-entry$ translation look-aside buffer (TLB) organized into 32 sets each having 4 ways. Assume that the TLB tag does not store any process id. The minimum length of the TLB tag in bits is ____.

gate2015-cse-set2 operating-system virtual-memory easy numerical-answers

Answer

5.22.32 Virtual Memory: GATE CSE 2015 Set 2 | Question: 47 top

https://gateoverflow.in/8247

A computer system implements 8 kilobyte pages and a $32-bit$ physical address space. Each page table entry contains a valid bit, a dirty bit, three permission bits, and the translation. If the maximum size of the page table of a process is 24 megabytes, the length of the virtual address supported by the system is _______ bits.

gate2015-cse-set2 operating-system virtual-memory normal numerical-answers

Answer

5.22.33 Virtual Memory: GATE CSE 2016 Set 1 | Question: 47 top

https://gateoverflow.in/39690

Consider a computer system with 40-bit virtual addressing and page size of sixteen kilobytes. If the computer system has a one-level page table per process and each page table entry requires 48 bits, then the size of the per-process page table is __________ megabytes.

gate2016-cse-set1 operating-system virtual-memory easy numerical-answers

Answer

5.22.34 Virtual Memory: GATE CSE 2018 | Question: 10 top

https://gateoverflow.in/204084

Consider a process executing on an operating system that uses demand paging. The average time for a memory access in the system 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 a 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. $(X - M)/D - X)$

gate2018-cse operating-system virtual-memory normal

Answer

5.22.35 Virtual Memory: GATE CSE 2019 | Question: 33 top

https://gateoverflow.in/302815

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 addressible. 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}

gate2019-cse operating-system virtual-memory

Answer

5.22.36 Virtual Memory: GATE CSE 2020 | Question: 53 top

https://gateoverflow.in/333178

Consider a paging system that uses 1-level page table residing in main memory and a TLB for address translation. Each main memory access takes 100 ns and 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 the required page is read from disk. TLB update time is negligible. The average memory access time in ns (round off to 1 decimal places) is ___________

gate2020-cse numerical-answers operating-system virtual-memory

Answer

5.22.37 Virtual Memory: GATE IT 2004 | Question: 66 top

https://gateoverflow.in/3709

In a virtual memory system, size of the virtual address is 32-bit, size of the physical address is 30-bit, page size is 4 Kbyte and size of each page table entry is 32-bit. The main memory is byte addressable. Which one of the following is the maximum number of bits that can be used for storing protection and other information in each page table entry?

A. 2
B. 10
C. 12
D. 14

gate2004-it operating-system virtual-memory normal

Answer

5.22.38 Virtual Memory: GATE IT 2008 | Question: 16 top

https://gateoverflow.in/3276

A paging scheme uses a Translation Look-aside Buffer (TLB). A TLB-access takes 10 ns and the main memory access takes 50 ns. What is the effective access time(in ns) if the TLB hit ratio is 90% and there is no page-fault?

A. 54
B. 60
C. 65
D. 75

gate2008-it operating-system virtual-memory normal

Answer

5.22.39 Virtual Memory: GATE IT 2008 | Question: 56 top

https://gateoverflow.in/3366

Match the following flag bits used in the context of virtual memory management on the left side with the different purposes on the right side of the table below.

Name of the bit		Purpose	
I.	Dirty	a.	Page initialization
II.	R/W	b.	Write-back policy
III.	Reference	c.	Page protection
IV.	Valid	d.	Page replacement policy

A. I-d, II-a, III-b, IV-c
B. I-b, II-c, III-a, IV-d
C. I-c, II-d, III-a, IV-b
D. I-b, II-c, III-d, IV-a

gate2008-it operating-system virtual-memory easy

Answer

Answers: Virtual Memory

5.22.1 Virtual Memory: GATE CSE 1989 | Question: 2-iv top

https://gateoverflow.in/87081

✔

(A)	Virtual memory	(r)	Address Translation
(B)	Shared memory	(s)	Mutual Exclusion
(C)	Look-ahead buffer	(q)	Spatial Locality
(D)	Look-aside buffer	(p)	Temporal Locality

https://gateoverflow.in/3304/difference-between-translation-buffer-translation-buffer

References

👍 24 votes

-- Prashant Singh (47.2k points)

5.22.2 Virtual Memory: GATE CSE 1990 | Question: 1-v top

https://gateoverflow.in/83833

i/o processors because processor will issue address for device controller and if there is no translation hardware then it ain't gonna be peachy.

👍 14 votes

-- ashish gusai (523 points)

5.22.3 Virtual Memory: GATE CSE 1990 | Question: 7-b top

https://gateoverflow.in/85404

✔ In 2 level virtual memory, for every memory access, we need 2 page table access (TLB is missing in the question) and 1 memory access for data. In the question TLB is not mentioned (old architecture). So, best case memory access time

$= 3 \times 10^{-8}\ s$.

We are given

$$3 \times 10^{-8} = 0.8 \left[\underbrace{3 \times 10^{-8}}_{\text{2 for Page Tables and 1 Mem}} + (1-h) \times 10^{-3} \right]$$

(For above, the main memory access time and page table access times are included for all memory accesses -- hence h is not multiplied with 3×10^{-8})

$$\implies 0.6 \times 10^{-8} = 0.8 \times 10^{-3} - 0.8h \times 10^{-3} \implies h = \frac{8\times10^{-4} - 6\times10^{-9}}{8\times10^{-4}} = 1 - 0.75 \times 10^{-5} \approx 99.99\%$$

46 votes

-- Arjun Suresh (332k points)

5.22.4 Virtual Memory: GATE CSE 1991 | Question: 03-xi top

https://gateoverflow.in/525

✔

A. True.
B. This is false. Example:- Peterson's solution is a purely software-based solution without the use of hardware.https://en.wikipedia.org/wiki/Peterson's_algorithm
C. False. Reference: https://en.wikipedia.org/wiki/Monitor_(synchronization)
D. True. This will happen if the page getting replaced is immediately referred to in the next cycle.
E. False. Memory can get fragmented with the best fit.

References

29 votes

-- Akash Kanase (36k points)

5.22.5 Virtual Memory: GATE CSE 1994 | Question: 1.21 top

https://gateoverflow.in/2464

✔

A. Is TRUE.
B. False. Overlaying is used to increase the address space usage when physical memory is limited on systems where virtual memory is absent. But it cannot increase the address space (logical) of a computer.
C. False. Like above is true for physical memory but here it is specified address space which should mean logical address space.
D. Is false. We can write in high level language just that the performance will be bad.

References

24 votes

-- Arjun Suresh (332k points)

5.22.6 Virtual Memory: GATE CSE 1995 | Question: 1.7 top

https://gateoverflow.in/2594

✔ Option (B) is true for segmented paging(segment size becomes large so paging done on each segment) which is different from paged segmentation(segment table size becomes large and paging done on segment table)

Here option (A) is true , as segment table are sometimes too large to keep in one pages. So, segment table divided into pages. Thus page table for each Segment Table pages are created.

For reference , read below :

https://stackoverflow.com/questions/16643180/differences-or-similarities-between-segmented-paging-and-paged-segmentation
Differences or similarities between Segmented paging and Paged segmentation scheme.

References

44 votes

-- Anurag Semwal (6.7k points)

5.22.7 Virtual Memory: GATE CSE 1995 | Question: 2.16 top https://gateoverflow.in/2628

Answer is (**C**).

Primary memory < virtual memory < secondary memory

We can extend VM upto the size of disk(secondary memory).

27 votes

-- jayendra (6.7k points)

5.22.8 Virtual Memory: GATE CSE 1996 | Question: 7 top https://gateoverflow.in/2759

Given that page size is 256 bytes (2^8) and Main memory (MM) is 1KB (2^{10}).

So total number of pages that can be accommodated in MM $= \frac{2^{10}}{2^8} = 4$.

So, essentially, there are 4 frames that can be used for paging (or page replacements).

The current sequence of pages in memory shows 3 pages $(17, 1, 63)$. So, there is 1 empty frame left. It also says that the least recently used page is 17.

Now, since page size given is 8 bits wide $(256\ B)$, and virtual memory is of 16 bit, we can say that 8 bits are used for offset. The given address sequence is hexadecimal can be divided accordingly:

Page Number in Hexadecimal	Offset	Page Number in Decimal
00	FF	0
01	0D	1
10	FF	16
11	B0	17

We only need the Page numbers, which can be represented in decimal as: $0, 1, 16, 17$.

Now, if we apply LRU algorithm to the existing frame with these incoming pages, we get the following states:

0	Miss	17	1	63	**0**
1	Hit	17	**1**	63	0
16	Miss	**16**	1	63	0
17	Miss	16	1	**17**	0

i. New status of the list is **16 1 17 0.**
ii. Number of page faults $= \mathbf{3}$.
iii. Page replacements are indicated above.

70 votes

-- Ashis Kumar Sahoo (699 points)

5.22.9 Virtual Memory: GATE CSE 1998 | Question: 2.18, UGCNET-June2012-III: 48 top https://gateoverflow.in/1691

Page fault rate $= \frac{1}{k}$

Page hit rate $= 1 - \frac{1}{k}$

Service time $= i$

Page fault service time $= i + j$

Effective memory access time,

$$= \frac{1}{k} \times (i + j) + \left(1 - \frac{1}{k}\right) \times i$$

$$= \frac{(i + j)}{k} + i - \frac{i}{k}$$

$$= \frac{i}{k} + \frac{j}{k} + i - \frac{i}{k}$$

$$= i + \frac{j}{k}$$

So, option (**A**) is correct.

48 votes

-- shashi shekhar (437 points)

5.22.10 Virtual Memory: GATE CSE 1999 | Question: 19 top

https://gateoverflow.in/1518

✔

a. Size of each segment $= \frac{2^{16}}{8} = 2^{13}$

   Let the size of page be 2^k bytes

   We need a page table entry for each page. For a segment of size 2^{13}, number of pages required will be

   2^{13-k} and so we need 2^{13-k} page table entries. Now, the size of these many entries must be less than or equal to the page size, for the page table of a segment to be requiring at most one page. So,

   $2^{13-k} \times 2 = 2^k$ (As a page table entry size is 2 bytes)

   $k = 7$ bits

   So, page size $= 2^7 = 128$ bytes

b. The TLB is placed after the segment table.

   Each segment will have $\frac{2^{13}}{2^9} = 2^4$ page table entries

   So, all page table entries of a segment will reside in the cache and segment number will differentiate between page table entry of each segment in the TLB cache.

   Total segments $= 8$

   Therefore 3 bits of tag is required

c. Number of Pages for a segment $= \frac{2^{16}}{2^9} = 2^7$

   Bits needed for page frame identification

   $= 7$ bits

   $+1$ valid bit

   $+3$ page protection bits

   $+1$ dirty bit

   $= 12$ bits needed for a page table entry

   Size of each page table entry $= 2$ bytes $= 16$ bits

   Number of bits left for aging $= 16 - 12 = 4$ bits

36 votes

-- Danish (3.4k points)

5.22.11 Virtual Memory: GATE CSE 1999 | Question: 2.10 top https://gateoverflow.in/1488

✔ Answer should be both (A) and (C) (Earlier GATE questions had multiple answers and marks were given only if all correct answers were selected).
Address translation is needed to provide memory protection so that a given process does not interfere with another. Otherwise we must fix the number of processors to some limit and divide the memory space among them -- which is not an "efficient" mechanism.

We also need at least 2 modes of execution to ensure user processes share resources properly and OS maintains control. This is not required for a single user OS like early version of MS-DOS.
Demand paging and DMA enhances the performances- not a strict necessity.
Ref: Hardware protection section in Galvin

49 votes -- Arjun Suresh (332k points)

5.22.12 Virtual Memory: GATE CSE 1999 | Question: 2.11 top https://gateoverflow.in/1489

✔ Virtual memory provides an interface through which processes access the physical memory. So,

A. Is false as direct access can never be slower.

B. Is true as without virtual memory it is difficult to give protected address space to processes as they will be accessing physical memory directly. No protection mechanism can be done inside the physical memory as processes are dynamic and number of processes changes from time to time.

C. Position independent can be produced even without virtual memory support.

D. This is one primary use of virtual memory. Virtual memory allows a process to run using a virtual address space and as and when memory space is required, pages are swapped in/out from the disk if physical memory gets full.

So, answer is (**B**) and (**D**).

46 votes -- Arjun Suresh (332k points)

5.22.13 Virtual Memory: GATE CSE 2000 | Question: 2.22 top https://gateoverflow.in/669

✔ Since nothing is told about page tables, we can assume page table access time is included in memory access time.

So, average memory access time

$= .9999 \times 1 + 0.0001 \times 10,000$
$= 0.9999 + 1$
$= 1.9999$ microseconds

Correct Answer: D

46 votes -- Arjun Suresh (332k points)

5.22.14 Virtual Memory: GATE CSE 2001 | Question: 1.20 top https://gateoverflow.in/713

✔ Option (B) is correct.

Swap space is the area on a hard disk which is part of the Virtual Memory of your machine, which is a combination of accessible physical memory (RAM) and the swap space. Swap space temporarily holds memory pages that are inactive. Swap space is used when your system decides that it needs physical memory for active processes and there is insufficient unused physical memory available. If the system happens to need more memory resources or space, inactive pages in physical memory are then moved to the swap space therefore freeing up that physical memory for other uses. Note that the access time for swap is slower therefore do not consider it to be a complete replacement for the physical memory. Swap space can be a dedicated swap partition (recommended), a swap file, or a combination of swap partitions and swap files.

57 votes -- Manoj Kumar (26.7k points)

5.22.15 Virtual Memory: GATE CSE 2001 | Question: 1.8 top https://gateoverflow.in/701

✔ (**D**) should be the answer.

(A) - MMU does this translation but MMU is part of VM (hardware).

(B), (C) - The main advantage of VM is the increased address space for programs, and independence of address space, which allows more degree of multiprogramming as well as option for process security.

(D) - VM requires switching of page tables (this is done very fast via switching of pointers) for the new process and thus it is theoretically slower than without VM. In anyway VM doesn't directly decrease the context switching overhead.

58 votes -- Arjun Suresh (332k points)

5.22.16 Virtual Memory: GATE CSE 2001 | Question: 2.21 top https://gateoverflow.in/739

✔ Number of pages $= 2^{32}/4KB = 2^{20}$ as we need to map every possible virtual address.

So, we need 2^{20} entries in the page table. Physical memory being $64\ MB$, a physical address must be 26 bits and a page (of size $4KB$) address needs $26-12 = 14$ address bits. So, each page table entry must be at least 14 bits.

So, total size of page table $= 2^{20} \times 14\ bits \approx 2\ MB$ (assuming PTE is 2 bytes)

Correct Answer: C

54 votes -- Arjun Suresh (332k points)

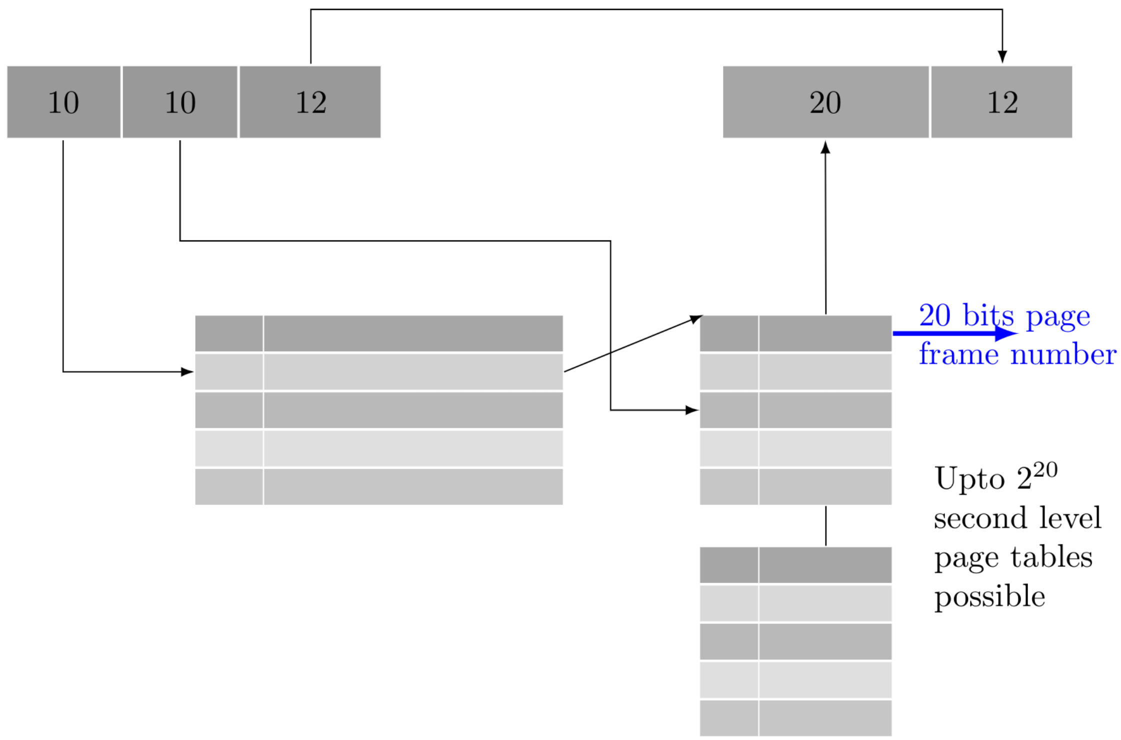
5.22.17 Virtual Memory: GATE CSE 2002 | Question: 19 top https://gateoverflow.in/872

✔

- $VA = 32\ bits$
- $PA = 32\ bits$
- Page size $= 4\ KB = 2^{12}B$
- PTE $= 4\ B$

Since page size is 4 KB we need $\lg 4K = 12$ bits as offset bits.

(A) It is given that equal number of bits should be used for indexing first level and second level page table. So, out of the remaining $32-12 = 20$ bits 10 bits each must be used for indexing into first level and second level page tables as follows:



(**B**) Since 10 bits are used for indexing to a page table, number of page table entries possible $= 2^{10} = 1024$. This is same for both first level as well as second level page tables.

(C)

Frame no $=$ 32 bit (Physical Address) -12 (Offset) $= 20$

No. of bits available for Storing Protection and other information in second level page table

$= 4 \times 8 - 20$

$= 32 - 20 = 12\ bits$

No. of bits in first level page table to address a second level page table is $\log_2$ of

$\frac{\text{Physical memory size}}{\text{\#Entries in a Second level page table} \times \text{PTE size}}$

$= \log_2 \left[\frac{2^{32}}{2^{10} \times 4}\right]$

$= \log_2 \left(2^{20}\right)$

$= 20\ bits.$

So here also, the no. of bits available for storing protection and other information $= 32 - 20 = 12\ bits.$

43 votes

-- Akash Kanase (36k points)

5.22.18 Virtual Memory: GATE CSE 2003 | Question: 26 top

https://gateoverflow.in/916

✔

A. Internal fragmentation exists only in the last level of paging.
B. There is no External fragmentation in the paging.
C. $\frac{2^{32}}{2^{10}} = 2^{22} = 4M$ entries in the page table which is very large. (**Answer**)
D. Not much relevant.

38 votes

-- Abhishek Singhal (233 points)

5.22.19 Virtual Memory: GATE CSE 2003 | Question: 78 top

https://gateoverflow.in/788

✔ 78. It's given cache is physically addressed. So, address translation is needed for all memory accesses. (I assume page table lookup happens after TLB is missed, and main memory lookup after cache is missed)

```
Average access time = Average address translation time + Average memory access time
= 1ns
(TLB is accessed for all accesses)
+ 2*10*0.04
(2 page tables accessed from main memory in case of TLB miss)
+ Average memory access time
= 1.8ns + Cache access time + Average main memory access time
= 1.8ns + 1 * 0.9 (90% cache hit)
+ 0.1 * (10+1) (main memory is accessed for cache misses only)
= 1.8ns + 0.9 + 1.1
= 3.8ns
```

We assumed that page table is in main memory and not cached. This is given in question also, though they do not explicitly say that page tables are not cached. But in practice this is common as given here. So, in such a system,

```
Average address translation time
= 1ns (TLB is accessed for all accesses)
+ 2*0.04 * [0.9 * 1 + 0.1 * 10]
(2 page tables accessed in case of TLB miss and they go through cache)
```

$= 1\ ns + 1.9 \times .08$

$= 1.152\ ns$

and average memory access time $= 1.152\ ns + 2\ ns = 3.152\ ns$

> If the same thing is repeated now probably you would get marks for both. 2003 is a long way back -- then page table caching never existed as given in the SE answers. Since it exists now, IIT profs will make this clear in question itself.

References

88 votes

-- gatecse (63.3k points)

5.22.20 Virtual Memory: GATE CSE 2003 | Question: 79 top

https://gateoverflow.in/43578

✔ First level page table is addressed using 10 bits and hence contains 2^{10} entries. Each entry is 4 bytes and hence this table requires 4 KB. Now, the process uses only 3 unique entries from this 1024 possible entries (two code pages starting from $0x00000000$ and two data pages starting from $0x00400000$ have same first 10 bits). Hence, there are only 3 second level page tables. Each of these second level page tables are also addressed using 10 bits and hence of size 4 KB. So,

```
total page table size of the process
= 4 KB + 3 * 4 KB
= 16 KB
```

Correct Answer: C

78 votes

-- Arjun Suresh (332k points)

5.22.21 Virtual Memory: GATE CSE 2006 | Question: 62, ISRO2016-50 top

https://gateoverflow.in/1840

✔ The page size of 4 KB. So, offset bits are 12 bits.

So, the remaining bits of virtual address, $32 - 12 = 20$ bits, will be used for indexing.

Number of sets $= 128/4 = 32(4\text{-way set}) \implies 5$ bits.

So, tag bits $= 20 - 5 = 15$ bits.

Correct option C.

50 votes

-- Vicky Bajoria (4.1k points)

5.22.22 Virtual Memory: GATE CSE 2006 | Question: 63, UGCNET-June2012-III: 45 top

https://gateoverflow.in/1841

✔ A is the best answer here.

Virtual memory provides

1. increased address space for processes
2. memory protection
3. relocation

So, when we don't need more address space, even if we get rid of virtual memory, we need hardware support for the other two. Without hardware support for memory protection and relocation, we can design a system (by either doing them in software or by partitioning the memory for different users) but those are highly inefficient mechanisms. i.e., there we have to divide the physical memory equally among all users and this limits the memory usage per user and also restricts the maximum number of users.

84 votes

-- Arjun Suresh (332k points)

5.22.23 Virtual Memory: GATE CSE 2008 | Question: 67 top

https://gateoverflow.in/490

✔ Physical address is 36 bits. So, number of bits to represent a page frame $= 36 - 12 = 24\ bits$ (12 offset bits as given in question to address 4 KB assuming byte addressing). So, each entry in a third level page table must have 24 bits for addressing the page frames.

A page in logical address space corresponds to a page frame in physical address space. So, in logical address space also we need 12 bits as offset bits. From the logical address which is of 32 bits, we are now left with $32 - 12 = 20\ bits$; these 20 $bits$ will be divided into three partitions (as given in the question) so that each partition represents 'which entry' in the i^{th} level page table we are referring to.

- An entry in level i page table determines 'which page table' at $(i+1)^{th}$ level is being referred.

Now, there is only 1 first level page table. But there can be many second level and third level page tables and "how many" of these exist depends on the physical memory capacity. (In actual the no. of such page tables depend on the memory usage of a given process, but for addressing we need to consider the worst case scenario). The simple formula for getting the number of page tables possible at a level is to divide the available physical memory size by the size of a given level page table.

$$\text{Number of third level page tables possible} = \frac{\text{Physical memory size}}{\text{Size of a third level page table}}$$
$$= \frac{2^{36}}{\text{Number of entries in a single third level page table} \times \text{ Size of an entry}}$$
$$= \frac{2^{36}}{2^9 \times 4} \because \text{ (bits 12-20 gives 9 bits)}$$
$$= \frac{2^{36}}{2^{11}}$$
$$= 2^{25}$$

PS: No. of third level page tables possible means the no. of distinct addresses a page table can have. At any given time, no. of page tables at level j is equal to the no. of entries in the level $j-1$, but here we are considering the **possible** page table addresses.

http://www.cs.utexas.edu/~lorenzo/corsi/cs372/06F/hw/3sol.html See Problem 3, second part solution - It clearly says that we should not assume that page tables are page aligned (page table size need not be same as page size unless told so in the question and different level page tables can have different sizes).
So, we need 25 bits in second level page table for addressing the third level page tables.

Similarly we need to find the no. of possible second level page tables and we need to address each of them in first level page table.

Now,

$$\text{Number of second level page tables possible} = \frac{\text{Physical memory size}}{\text{Size of a second level page table}}$$
$$= \frac{2^{36}}{\text{Number of entries in a single second level page table} \times \text{ Size of an entry}}$$
$$= \frac{2^{36}}{2^9 \times 4} \because \text{ (bits 21-29 gives 9 bits)}$$
$$= \frac{2^{36}}{2^{11}}$$
$$= 2^{25}$$

So, we need 25 bits for addressing the second level page tables as well.

So, answer is (**D**).

Video Explanation for Multi-level Paging: https://youtu.be/bArypfVmPb8

(Edit:-
There is nothing to edit for such awesome explanation but just adding one of my comment if it is useful - comment. However if anyone finds something to add (or correct) then feel free to do that in my comment.)

References

233 votes

-- Arjun Suresh (332k points)

5.22.24 Virtual Memory: GATE CSE 2009 | Question: 10 top

https://gateoverflow.in/1302

✔ It is (**B**).

The page table contains the page frame number essentially.

25 votes

-- Gate Keeda (15.9k points)

5.22.25 Virtual Memory: GATE CSE 2009 | Question: 34 top

https://gateoverflow.in/1320

✔ Option - > (**B**)

A. It reduces the memory access time to read or write a memory location. -> No This is false. Actually because of multi level paging we increase no of memory accesses.

B. It helps to reduce the size of page table needed to implement the virtual address space of a process -> This is **true**, In case of big virtual memory page, size of Page table can also be too huge to fit in single Page. So we do multi level paging.

C. It is required by the translation lookaside buffer.-> Examiner was not being enough creative here, This is false & There is no relation. This option is just given for no reason !

D. It helps to reduce the number of page faults in page replacement algorithms.-> This is false, we might increase no of page faults. (Due to second / thirrd level page not in memory here !) So this is false.

41 votes

-- Akash Kanase (36k points)

5.22.26 Virtual Memory: GATE CSE 2011 | Question: 20, UGCNET-June2013-II: 48 top

https://gateoverflow.in/2122

✔ Open slides 12-13 to check :

http://web.cs.ucla.edu/~ani/classes/cs111.08w/Notes/Lecture%2016.pdf

$$\begin{aligned}\text{EMAT} &= \frac{1}{10^6} \times 10\ ms + \left(1 - \frac{1}{10^6}\right) \times 20\ ns \\ &= 29.99998\ ns \\ &\approx 30\ ns\end{aligned}$$

Answer = **option B**

References

61 votes

-- Amar Vashishth (25.2k points)

5.22.27 Virtual Memory: GATE CSE 2013 | Question: 52 top

https://gateoverflow.in/379

✔ Let the page size be x.

Since virtual address is 46 bits, we have total number of pages $= \frac{2^{46}}{x}$

We should have an entry for each page in last level page table which here is $T3$. So,
Number of entries in $T3$ (sum of entries across all possible $T3$ tables) $= \frac{2^{46}}{x}$

Each entry takes 32 bits $= 4$ bytes. So, total size of $T3$ tables $= \frac{2^{46}}{x} \times 4 = \frac{2^{48}}{x}$ bytes

Now, no. of $T3$ tables will be Total size of $T3$ tables/page table size and for each of these page tables, we must have a $T2$ entry. Taking $T3$ size as page size, no. of entries across all $T2$ tables $= \frac{\frac{2^{48}}{x}}{x} = \frac{2^{48}}{x^2}$

Now, no. of $T2$ tables (assuming $T2$ size as page size) $= \frac{2^{48}}{x^2} \times 4$ bytes $= \frac{\frac{2^{50}}{x^2}}{x} = \frac{2^{50}}{x^3}$.

Now, for each of these page table, we must have an entry in $T1$.
So, number of entries in $T1 = \frac{2^{50}}{x^3}$

And size of $T1 = \frac{2^{50}}{x^3} \times 4 = \frac{2^{52}}{x^3}$

Given in question, size of $T1$ is page size which we took as x. So,

$$x = \frac{2^{52}}{x^3}$$
$$\implies x^4 = 2^{52}$$
$$\implies x = 2^{13}$$
$$\implies x = 8\text{ KB}$$

Correct Answer: C

144 votes — -- Arjun Suresh (332k points)

I already put it as comment, in case if one skipped it.

One other method to find page size-

We know that all levels page tables must be completely full except outermost, the outermost page table may occupy whole page or less. But in question, it is given that Outermost page table occupies whole page.

Now let page size is 2^p Bytes.

Given that PTE = 32 bits = 4 Bytes $= 2^2$ Bytes.

Number of entries in any page of any pagetable =page size/PTE $= \frac{2^p}{2^2} = 2^{p-2}$.

Therefore Logical address split is

p-2	p-2	p-2	p

logical address space is 46 bits as given. Hence. equation becomes,

$(p-2)+(p-2)+(p-2)+p = 46$

$\Rightarrow p = 13.$

Therefore, page size is 2^{13} Bytes = **8KB**.

References

137 votes — -- Sachin Mittal (15.8k points)

5.22.28 Virtual Memory: GATE CSE 2013 | Question: 53

https://gateoverflow.in/43294

✔ Let the page size be x.

Since virtual address is 46 bits, we have total number of pages $= \frac{2^{46}}{x}$

We should have an entry for each page in last level page table which here is $T3$. So,

Number of entries in $T3$ (sum of entries across all possible $T3$ tables) $= \frac{2^{46}}{x}$

Each entry takes 32 bits $=$ 4 bytes. So, total size of $T3$ tables $= \frac{2^{46}}{x} \times 4 = \frac{2^{48}}{x}$ bytes

Now, no. of $T3$ tables will be Total size of $T3$ tables/page table size and for each of these page tables, we must have a $T2$ entry. Taking $T3$ size as page size, no. of entries across all $T2$ tables

$= \frac{\frac{2^{48}}{x}}{x} = \frac{2^{48}}{x^2}$

Now, no. of T2 tables (assuming T2 size as pagesize) $= \frac{2^{48}}{x^2} \times 4$ bytes $= \frac{\frac{2^{50}}{x^2}}{x} = \frac{2^{50}}{x^3}$.

Now, for each of these page table, we must have an entry in $T1$. So, number of entries in $T1$

$= \frac{2^{50}}{x^3}$

And size of $T1 = \frac{2^{50}}{x^3} \times 4 = \frac{2^{52}}{x^3}$

Given in question, size of $T1$ is page size which we took as x. So,

$x = \frac{2^{52}}{x^3}$

$\implies x^4 = 2^{52}$

$\implies x = 2^{13} = 8KB$

Min. no. of page color bits = No. of set index bits + no. of offset bits − no. of page index bits (This ensures no synonym maps to different sets in the cache)

We have 1MB cache and 64B cache block size. So,

number of sets $= 1\text{MB}/(64 \text{ B} \times \text{Number of blocks in each set}) = 16K/16(16 \text{ way set associative}) = 1K = 2^{10}$.

So, we need 10 index bits. Now, each block being $64(2^6)$ bytes means we need 6 offset bits.

And we already found page size $= 8KB = 2^{13}$, so 13 bits to index a page

Thus, no. of page color bits $= 10 + 6 - 13 = 3$.

With 3 page color bits we need to have $2^3 = 8$ different page colors

More Explanation:

A synonym is a physical page having multiple virtual addresses referring to it. So, what we want is no two synonym virtual addresses to map to two different sets, which would mean a physical page could be in two different cache sets. This problem never occurs in a physically indexed cache as indexing happens via physical address bits and so one physical page can never go to two different sets in cache. In virtually indexed cache, we can avoid this problem by ensuring that the bits used for locating a cache block (index+offset) of the virtual and physical addresses are the same.

In our case we have 6 offset bits $+10$ bits for indexing. So, we want to make these 16 bits same for both physical and virtual address. One thing is that the page offset bits -13 bits for 8 KB page, is always the same for physical and virtual addresses as they are never translated. So, we don't need to make these 13 bits same. We have to only make the remaining $10 + 6 - 13 = 3$ bits same. Page coloring is a way to do this. Here, all the physical pages are colored and a physical page of one color is mapped to a virtual address by OS in such a way that a set in cache always gets pages of the same color. So, in order to make the 3 bits same, we take all combinations of it $(2^3 = 8)$ and colors the physical pages with 8 colors and a cache set always gets a page of one color only. (In page coloring, it is the job of OS to ensure that the 3 bits are the same).

http://ece.umd.edu/courses/enee646.F2007/Cekleov1.pdf

http://cseweb.ucsd.edu/classes/fa14/cse240A-a/pdf/08/CSE240A-MBT-L18-VirtualMemory.ppt.pdf

https://en.wikipedia.org/wiki/CPU_cache#Address_translation

https://en.wikipedia.org/wiki/Cache_coloring

Correct Answer: C

References

52 votes -- Arjun Suresh (332k points)

5.22.29 Virtual Memory: GATE CSE 2014 Set 3 | Question: 33 top https://gateoverflow.in/2067

✔ EMAT=TLB hit × (TLB access time + memory access time) + TLB miss(TLB access time + page table access time+memory access time)

$= 0.6(10 + 80) + 0.4(10 + 80 + 80)$

$= 54 + 68$

$= 122\ msec$

54 votes -- neha pawar (3.3k points)

5.22.30 Virtual Memory: GATE CSE 2015 Set 1 | Question: 12 top https://gateoverflow.in/8186

✔ total no of pages $= \frac{2^{32}}{2^{12}} = 2^{20}$

We need a PTE for each page and an entry is 4 bytes. So,
page table size $= 4 \times 2^{20} = 2^{22}B = \mathbf{4MB}$

39 votes

-- Anoop Sonkar (4.1k points)

5.22.31 Virtual Memory: GATE CSE 2015 Set 2 | Question: 25 top

https://gateoverflow.in/8120

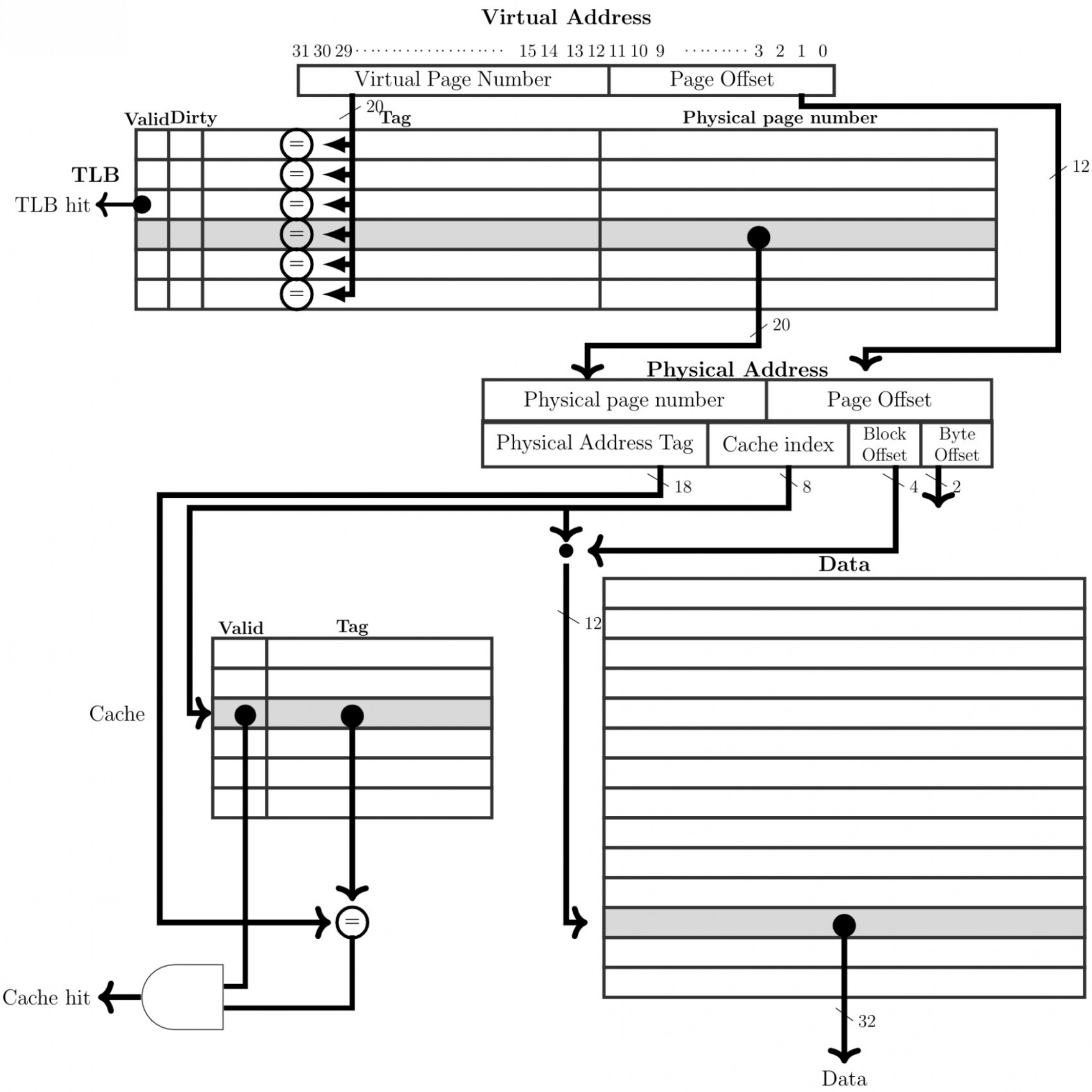
✔ Ans $40 - (5 + 13) = 22$ bits

TLB maps a virtual address to the physical address of the page. (The lower bits of page address (page offset bits) are not used in TLB as they are the same for virtual as well as physical addresses). Here, for 8 kB page size we require 13 page offset bits.

In TLB we have 32 sets and so virtual address space is divided into 32 using 5 set bits. (Associativity doesn't affect the set bits as they just adds extra slots in each set).

So, number of tag bits $= 40 - 5 - 13 = 22$

Following diagram shows how TLB and Cache works:



63 votes

-- Vikrant Singh (11.2k points)

5.22.32 Virtual Memory: GATE CSE 2015 Set 2 | Question: 47 top https://gateoverflow.in/8247

✔ $8\;KB$ pages means 13 offset bits.
For 32 bit physical address, $32-13=19$ page frame bits must be there in each PTE (Page Table Entry).
We also have 1 valid bit, 1 dirty bit and 3 permission bits.
So, total size of a PTE (Page Table Entry) $= 19+5=24$ bits $= 3$ bytes.

Given in question, maximum page table size $= 24\;MB$
Page table size $=$ No. of PTEs $\times$ size of an entry
So, no. of PTEs $= 24MB/3B = 8M$

Virtual address supported = No. of PTEs * Page size (As we need a PTE for each page and assuming single-level paging)
= $8M$*$8KB$
= 64GB= 2^{36} Bytes

So, length of virtual address supported = 36 bits (assuming byte addressing)

77 votes

-- Arjun Suresh (332k points)

5.22.33 Virtual Memory: GATE CSE 2016 Set 1 | Question: 47 top https://gateoverflow.in/39690

✔ $\text{No. of pages}(N) = 2^{26} = \text{No. of entries in Page Table}$
$\text{Page Table Entry Size}(E) = 6 \text{ bytes}$

$\text{So, Page Table Size} = n \times e = 2^{26} \times 6 \text{ bytes} = 384 \text{ MB}$

45 votes

-- G VENKATESWARLU (461 points)

5.22.34 Virtual Memory: GATE CSE 2018 | Question: 10 top https://gateoverflow.in/204084

✔ Let P be the page fault rate.

Average memory access time $= (1-$ page fault rate$)\times$ memory access time when no page fault $+$ Page fault rate $\times$ Memory access time when page fault.

$X = (1-P)M + P\,D$

$X = M + P(D-M)$

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

(B) is the answer.

47 votes

-- Hemant Parihar (11.9k points)

5.22.35 Virtual Memory: GATE CSE 2019 | Question: 33 top https://gateoverflow.in/302815

✔ TLB Entry: $\boxed{\text{Page Number}}\boxed{\text{Frame Number}}$

Memory is word addressable.

- $\text{Word size} = 4 \text{ Bytes}$
- $\text{Page size} = 8 \text{ KB} = 2^{11} \text{ words}$
- $\text{Virtual Memory size} = 2^{64} \text{ words}$
- $\text{Number of pages possible} = 2^{53}$
- $\text{Number of bits required for Page number} = 53 \text{ bits}$
- $\text{Number of bits required for Page offset} = 64-53 = 11 \text{ bits}$

At a time TLB contains $128 = 2^7$ distinct page numbers.

If a page number is found in TLB then there will be a hit for all the words (Word addresses) of that Page.

1 - page hit implies 2^{11} distinct virtual address hits.

So 2^7page hit implies $2^7 * 2^{11} = 2^8 * 2^{10} = 256 * 2^{10}$ virtual address hits

Option **B.** At most, $256 * 2^{10}$ distinct virtual addresses can be translated without any TLB miss.

58 votes -- Soumya Jain ($12.5k$ points)

5.22.36 Virtual Memory: GATE CSE 2020 | Question: 53 top

https://gateoverflow.in/333178

✔ Given,

1. Main Memory access time: 100 ns
2. TLB lookup time: 20 ns
3. Time to transfer one page to/from disk: 5000 ns
4. TLB hit ratio: 0.95
5. Page fault rate: 0.10
6. 20 % of page faults needs to be written back to disk

Hence, effective memory access time =

$0.95(20 + 100) + 0.05\{0.90(20 + 100 + 100) + 0.10[0.80(20 + 100 + 5000 + 100) + 0.20(20 + 100 + 5000 + 5000 + 100)]\}$
$= 155.0$ ns

Explanation:

If there is a TLB hit, you just need to access the memory. If there is a miss 1 TLB lookup was wasted,

1. You need to lookup the page table for the entry and then access the required location, requiring 2 memory accesses - Assuming No Page fault occurs.
2. If there is a page fault, Then 1 memory access was wasted (you can only know that the page is not present in memory by checking the corresponding page entry in the page table). 80 % of the time, you'll only be fetching a page from secondary storage which takes 5000 ns, 20% of the time, you'll need to write a dirty page back to disk and bring the page (which caused the page fault) back to main memory, requiring $5000 + 5000$ ns

62 votes -- Debasish Das ($1.5k$ points)

- For all memory accesses in a system with virtual memory we need Virtual Address to Physical Address translation and this goes through TLB.
- On TLB hit, we get the physical address.
- On TLB miss, we have to do page table access which always resides in physical memory (no page fault possible here).
- In the question it is given $1-$ level page table is used. So, TLB miss will need one physical memory access to get the physical address.
- Question mentions page fault rate as 10% and this should default to 10 page faults every 100 memory accesses. (Since TLB miss rate is 5% and for normal program run a TLB hit and page fault cannot happen for a memory access (can happen for invalid memory accesses), it is also possible to consider page fault rate as 10% of all TLB misses. See the last part of the answer for this.

In the question page transfer time is given. This is different from page fault service time which includes the page transfer time + the memory access time as once the page is filled, a new memory access is initiated.

So, $\text{Average Memory Access Time} = \text{Address Translation Time} + \text{Data Retrieval Time}$
$= \text{TLB access time} + \text{TLB Miss ratio} \times \text{Page Table Access time } + \text{Main memory access time} + \text{Page fault rate} \times (\text{Page}$
$= 20 + 0.05 \times 100 + 100 + 0.1 \times (5000 + 20 + 0.05 \times 100 + 100) + 0.1 \times 0.2 \times 5000$
$= 20 + 5 + 100 + 512.5 + 100$
$= 737.5 \ ns$

PS: If the question had given page fault service time also as 5000 answer will be
$20 + 0.05 \times 100 + 0.9 \times 100 + 0.1 \times 5000 + 0.1 \times 0.2 \times 5000 = 25 + 90 + 500 + 100 = 715 \ ns$

"Assume that the TLB hit ratio is 95%, page fault rate is 10%"

If this statement is changed to

"Assume that the TLB hit ratio is 95%, and **when TLB miss happens** page fault rate is 10%"

$\text{Average Memory Access Time} = \text{Address Translation Time} + \text{Data Retrieval Time}$
$= \text{TLB access time} + \text{TLB Miss ratio} \times \text{Page Table Access time } + \text{Main memory access time} + \text{Page fault rate} \times (\text{Page}$
$= 20 + 0.05 \times 100 + 100 + 0.05 \times 0.1 \times (5000 + 20 + 0.05 \times 100 + 100) + 0.05 \times 0.1 \times 0.2 \times 5000$
$= 20 + 5 + 100 + 25.625 + 5$
$= 155.625 \ ns$

If "memory access being restarted" is ignored for page fault, this will be

$$
\begin{aligned}
&= 20 + 0.05 \times 100 + 100 + 0.05 \times 0.1 \times (5000) + 0.05 \times 0.1 \times 0.2 \times 5000 \\
&= 20 + 5 + 100 + 25 + 5 \\
&= 155 \ ns
\end{aligned}
$$

Ideally the answer key should be $715 - 738$ due to the confusion in the meaning of page transfer time as most standard resources use page fault service time instead.

If we assume page fault rate is given "only when TLB miss happens" answer should be $155 - 155.7$

A previous year question where page fault rate "per instruction" is clearly mentioned in question: https://gateoverflow.in/318/gate2004-47. This GATE2020 question is VERY POORLY framed and must be challenged.

Another similar question where TLB miss is taken as per memory access is given below (See the equation used in 3-e)

https://gateoverflow.in/?qa=blob&qa_blobid=5047954265438465988

References

👍 21 votes -- Arjun Suresh (332k points)

5.22.37 Virtual Memory: GATE IT 2004 | Question: 66 top

https://gateoverflow.in/3709

✔ Answer is (**D**).

Page table entry must contain bits for representing frames and other bits for storing information like dirty bit,reference bit etc

No. of frames (no. of possible pages) $=$ Physical memory size/ Page size $= 2^{30}/2^{12} = 2^{18}$

$18 + x = 32$ (PT entry size=32 bit)

$x = 14$ bits

👍 37 votes -- neha pawar (3.3k points)

5.22.38 Virtual Memory: GATE IT 2008 | Question: 16 top

https://gateoverflow.in/3276

✔ Effective access time $=$ hit ratio $\times$ time during hit $+$ miss ratio $\times$ time during miss

In both cases TLB is accessed and assuming page table is accessed from memory only when TLB misses.

$= 0.9 \times (10 + 50) + 0.1 \times (10 + 50 + 50)$

$= 54 + 11 = 65$

Correct Answer: C

👍 41 votes -- Arjun Suresh (332k points)

5.22.39 Virtual Memory: GATE IT 2008 | Question: 56 top

https://gateoverflow.in/3366

✔ Option (**D**).

Dirty bit : The **dirty bit** is set when the processor writes to (modifies) this memory. The **bit** indicates that its associated block of memory has been modified and has not been saved to storage yet. **Dirty bits** are used by the CPU cache and in the page replacement algorithms of an operating system.

R/W bit : If the bit is set, the page is read/write. Otherwise when it is not set, the page is read-only.

Reference bit is used in a version of FIFO called second chance (SC) policy, in order to avoid replacement of heavily used page.. It is set to one when a page is used heavily and periodically set to 0. Since it is used in a version FIFO which is a page replacement policy, this bit is come under category of page replacement.

Valid bit is not used for page replacement. It is not used in any page replacement policy. It tells the page in the memory is valid or not. If it is valid it is directly used and if it is not then a fresh page is loaded. So, basically it is page initialization, because we are not replacing, it is initializing, we not knocking out somebody, we are filling empty space, so initialization, so option (**D**).

63 votes

-- Vicky Bajoria (4.1k points)

Answer Keys

5.1.1	C	5.1.2	B	5.1.3	C	5.2.1	2	5.2.2	A
5.2.3	A;B;C	5.2.4	A	5.3.1	N/A	5.3.2	N/A	5.3.3	N/A
5.3.4	B	5.3.5	D	5.3.6	B	5.3.7	3	5.3.8	10
5.3.9	346	5.3.10	B	5.3.11	C	5.3.12	B	5.3.13	C
5.4.1	N/A	5.4.2	N/A	5.4.3	B	5.4.4	N/A	5.4.5	N/A
5.4.6	9.006	5.4.7	D	5.4.8	N/A	5.4.9	N/A	5.4.10	D
5.4.11	D	5.4.12	A	5.4.13	800	5.4.14	A	5.4.15	C
5.4.16	B	5.4.17	A	5.4.18	B	5.4.19	C	5.4.20	C
5.4.21	B	5.4.22	B	5.4.23	D	5.4.24	99.55 : 99.65	5.4.25	14020
5.4.26	6.1 : 6.2	5.4.27	85	5.4.28	C	5.4.29	D	5.4.30	B
5.4.31	D	5.5.1	C	5.5.2	D	5.5.3	D	5.5.4	4.0 : 4.1
5.5.5	A;C	5.5.6	B	5.6.1	C	5.6.2	B	5.6.3	C
5.6.4	31	5.6.5	C	5.7.1	B	5.8.1	90.00	5.8.2	C
5.8.3	D	5.8.4	A	5.8.5	C	5.8.6	C	5.8.7	D
5.8.8	A	5.9.1	A	5.9.2	B	5.9.3	C	5.9.4	A
5.9.5	B	5.9.6	C	5.10.1	3.2	5.10.2	N/A	5.10.3	B
5.10.4	B	5.10.5	10000	5.10.6	A	5.10.7	C	5.10.8	C
5.10.9	B	5.11.1	A	5.11.2	D	5.11.3	B	5.12.1	N/A
5.12.2	B	5.12.3	B	5.12.4	C	5.12.5	C	5.12.6	A
5.12.7	B	5.12.8	C	5.12.9	C	5.12.10	B	5.12.11	A
5.12.12	C	5.12.13	B	5.12.14	A	5.12.15	C	5.12.16	A
5.12.17	A	5.12.18	B	5.12.19	7	5.12.20	D	5.12.21	6
5.12.22	A	5.12.23	1	5.12.24	D	5.12.25	B	5.12.26	A;C
5.12.27	4108 : 4108	5.12.28	C	5.12.29	A	5.12.30	B	5.13.1	N/A
5.13.2	N/A	5.13.3	N/A	5.14.1	B	5.14.2	C	5.14.3	B
5.14.4	B	5.15.1	N/A	5.15.2	N/A	5.15.3	N/A	5.15.4	C
5.15.5	B	5.15.6	A	5.15.7	D	5.15.8	A	5.15.9	N/A
5.15.10	19	5.15.11	B	5.15.12	D	5.15.13	A	5.15.14	B
5.15.15	A	5.15.16	B	5.15.17	A	5.15.18	B	5.15.19	C
5.15.20	D	5.15.21	A	5.15.22	C	5.15.23	B	5.15.24	7.2
5.15.25	1000	5.15.26	5.5	5.15.27	12	5.15.28	D	5.15.29	C

5.15.30	A	5.15.31	8.25	5.15.32	3	5.15.33	29	5.15.34	2
5.15.35	C	5.15.36	5.25:5.26	5.15.37	12 : 12	5.15.38	A;C;D	5.15.39	D
5.15.40	D	5.15.41	B	5.15.42	D	5.15.43	C	5.16.1	D
5.16.2	N/A	5.16.3	N/A	5.16.4	N/A	5.16.5	N/A	5.16.6	N/A
5.16.7	N/A	5.16.8	N/A	5.16.9	N/A	5.16.10	C	5.16.11	C
5.16.12	N/A	5.16.13	D	5.16.14	N/A	5.16.15	B	5.16.16	N/A
5.16.17	N/A	5.16.18	B	5.16.19	N/A	5.16.20	B	5.16.21	N/A
5.16.22	N/A	5.16.23	N/A	5.16.24	B	5.16.25	C	5.16.26	D
5.16.27	A	5.16.28	B	5.16.29	B	5.16.30	D	5.16.31	A
5.16.32	A	5.16.33	A	5.16.34	B	5.16.35	D	5.16.36	C
5.16.37	C	5.16.38	3	5.16.39	A	5.16.40	C	5.16.41	D
5.16.42	C	5.16.43	80	5.16.44	A	5.16.45	D	5.16.46	A
5.16.47	B	5.16.48	A	5.16.49	C	5.16.50	C	5.16.51	D
5.17.1	N/A	5.17.2	N/A	5.17.3	A	5.17.4	D;E	5.17.5	N/A
5.17.6	N/A	5.17.7	C	5.17.8	N/A	5.17.9	B	5.17.10	C
5.17.11	N/A	5.17.12	C	5.17.13	B	5.17.14	C	5.17.15	A
5.17.16	A	5.17.17	B	5.17.18	B	5.17.19	B	5.17.20	7
5.17.21	D	5.17.22	D	5.17.23	A	5.17.24	B	5.17.25	C
5.17.26	B	5.18.1	N/A	5.18.2	D	5.18.3	B	5.19.1	N/A
5.19.2	B	5.19.3	B	5.19.4	C	5.19.5	7	5.19.6	A
5.19.7	A;B;D	5.19.8	D	5.20.1	A;C	5.21.1	A	5.21.2	D
5.21.3	C	5.21.4	D	5.21.5	D	5.21.6	B	5.21.7	A;D
5.21.8	A	5.22.1	N/A	5.22.2	N/A	5.22.3	99.99	5.22.4	B;C;E
5.22.5	A	5.22.6	A	5.22.7	C	5.22.8	N/A	5.22.9	A
5.22.10	N/A	5.22.11	A;C	5.22.12	B;D	5.22.13	D	5.22.14	B
5.22.15	D	5.22.16	C	5.22.17	N/A	5.22.18	C	5.22.19	D
5.22.20	C	5.22.21	C	5.22.22	C	5.22.23	D	5.22.24	B
5.22.25	B	5.22.26	B	5.22.27	C	5.22.28	C	5.22.29	122
5.22.30	4	5.22.31	22	5.22.32	36	5.22.33	384	5.22.34	B
5.22.35	B	5.22.36	155:156	5.22.37	D	5.22.38	C	5.22.39	D

Since GATE Overflow (GO) started in August 2014, a lot of people have dedicated their time and effort in bringing this book now. Initiated by Omesh Pandita and Arjun Suresh as a Q/A platform for CSE students, Kathleen Bankson was instrumental in getting all previous year GATE questions here. Then experts like Praveen Saini, Happy Mittal, Sankaranarayanan P.N., Suraj Kumar etc. have contributed a lot to the answers in GATE Overflow. Pragy Agarwal even after topping GATE has continuously contributed to GATE Overflow with his knowledge as well as in making the contents beautiful with fine latex skills. We also have to thank the work done by our editors Jotheeswari Arasakumar, Lakshman Patel, Soujanya Reddy, Sabiha Banu, Misbah Ghaya, Ishrat Jahan, Nataliyah Ahmad, Kanza Bayad, Andrijana Milicevik and Pavan Singh who are continuously adding and keeping the contents in GATE Overflow neat and clean. There are also many toppers of GATE 2015-21 who are contributing a lot in GATE Overflow. The list of all the contributors can be found on GATE Overflow site but even that does not include the contributions of some like Arif Ali Anapparakkal for the book design, Arvind Devaraj and others for guidance and help, Aravind S.R. for fixing the page numbering issue, Marlyn Joseph for book printing support and Ajay K. Soni and Akshat Joshi for book design support. Our final proof reading and editing were done with the help of Soujanya Reddy, Lakshman Patel, Ajay K. Soni, Lakshman Patel, M. Chandrakiruthiga, Pooja Khatri, Shikha Mallick, Manoja Rajalakshmi Aravindakshan, Subarna Das and Sukanya Das. Special mention to Ajay K. Soni, Lakshman Patel, Aditya Ravishankar, Sweta Kumari Rawani, Gyanendra Singh, Sabiha Banu, Sukanya Das, Subarna Das and Pooja Khatri for handling all the difficult images in Tikz.

From the GATE Overflow family, we thank the contributions of Silpa V.S., Digvijay Pandey, Rahul Kumar Yadav and others for getting the GATECSE Last-Rank page maintained. Bikram Ballav is behind most of the free exams on GATE Overflow and our exam interface which is already used for 50,000+ exam takes is contributed by Arindam Sarkar. Pragy Agarwal is also behind the rank and score predictor tool, used by GATE Overflow which has 99-100% accuracy over the last 6 years. Special thanks to Sachin Mittal for making the How to Use GATE Overflow videos, Silpa V.S. for classifying the questions topic-wise for the book, Pooja Palod for making our first GATE schedule and Debashish Deka, Priti Sharma and Digvijay Pandey for GATE Overflow classroom contributions. We also thank all the GATE toppers who took time to write a review for GATE Overflow. Finally on behalf of GATE Overflow family and all GATE aspirants we thank the wonderful faculties of IISc., IITs and other premier institutes who are behind the creation of such wonderful questions.