5 Operating System (297)

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

5.1 Context Switch (3)

5.1.1 Context Switch: GATE1999-2.12 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

5.1.2 Context Switch: GATE2000-1.20, ISRO2008-47 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 operating-system easy isro2008 context-switch

5.1.3 Context Switch: GATE2011-6, UGCNET-June2013-III-62 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 operating-system context-switch easy ugcnetjune2013iii

5.2 Deadlock Prevention Avoidance Detection (2)

5.2.1 Deadlock Prevention Avoidance Detection: GATE2018-24 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 operating-system deadlock-prevention-avoidance-detection easy numerical-answers

5.2.2 Deadlock Prevention Avoidance Detection: GATE2018-39 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 operating-system deadlock-prevention-avoidance-detection normal

5.3 Disk Scheduling (12)

5.3.1 Disk Scheduling: GATE1989-4-xii https://gateoverflow.in/88222

Provide short answers to the following questions:

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

5.3.2 Disk Scheduling: GATE1990-9b 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

5.3.3 Disk Scheduling: GATE1995-20 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

5.3.4 Disk Scheduling: GATE1999-1.10 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

5.3.5 Disk Scheduling: GATE2004-12 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 operating-system disk-scheduling normal

5.3.6 Disk Scheduling: GATE2004-IT-62
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 Fist Serve)?

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

gate2004-it operating-system disk-scheduling normal

5.3.7 Disk Scheduling: GATE2007-IT-82
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

5.3.8 Disk Scheduling: GATE2007-IT-83
https://gateoverflow.in/3535

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.

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

Hint : We have 2048 tracks so, maximum swing (seek length) can be 2047
Which corresponds to a seek length of 211−1 in the 11th service

gate2007-it operating-system disk-scheduling normal

5.3.9 Disk Scheduling: GATE2009-31
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 operating-system disk-scheduling normal

5.3.10 Disk Scheduling: GATE2014-1-19
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.

gate2014-1 operating-system disk-scheduling numerical-answers normal

5.3.11 Disk Scheduling: GATE2015-1-30
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.

gate2015-1 operating-system disk-scheduling normal numerical-answers

5.3.12 Disk Scheduling: GATE2016-1-48 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__________.

gate2016-1 operating-system disk-scheduling normal numerical-answers

5.4 Disks (32)

5.4.1 Disks: GATE1990-7-c 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.

descriptive operating-system disks gate1990

5.4.2 Disks: GATE1993-6.7 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

5.4.3 Disks: GATE1993-7.8 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

5.4.4 Disks: GATE1995-14 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

5.4.5 Disks: GATE1996-23 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

5.4.6 Disks: GATE1997-74 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

5.4.7 Disks: GATE1998-17 https://gateoverflow.in/1731

Calculate the total time required to read 35 sectors on a 2-sided floppy disk. Assume that each track has 8 sectors and the track-to-track step time is 8 milliseconds. The first sector to be read is sector 3 on track 10. Assume that the diskette is soft sectored and the controller has a 1-sector buffer. The diskette spins at 300 RPM and initially, the head is on track 10.

gate1998 operating-system disks normal numerical-answers

5.4.8 Disks: GATE1998-2-9 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

5.4.9 Disks: GATE1998-25-a 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

5.4.10 Disks: GATE1998-25b 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

5.4.11 Disks: GATE1999-2-18, ISRO2008-46 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

5.4.12 Disks: GATE2001-1.22 https://gateoverflow.in/715

Which of the following requires a device driver?

A disk driver is software which enables communication between internal hard disk (or drive) and computer.

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

gate2001 operating-system disks easy

5.4.13 Disks: GATE2001-20 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 operating-system disks normal descriptive

5.4.14 Disks: GATE2001-8 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 operating-system disks normal descriptive

5.4.15 Disks: GATE2003-25, ISRO2009-12 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 operating-system disks normal isro2009

5.4.16 Disks: GATE2004-49

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

Disk block address= 4B, so
No. of address per block= 1024/4=256

gate2004 operating-system disks normal

5.4.17 Disks: GATE2005-21

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 operating-system disks easy

5.4.18 Disks: GATE2005-IT-63

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

5.4.19 Disks: GATE2005-IT-81-a

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

5.4.20 Disks: GATE2005-IT-81-b

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

5.4.21 Disks: GATE2007-11, ISRO2009-36, ISRO2016-21

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 operating-system disks normal isro2016

5.4.22 Disks: GATE2007-IT-44, ISRO2015-34 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

5.4.23 Disks: GATE2008-32 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 operating-system disks normal

5.4.24 Disks: GATE2009-51 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 operating-system disks normal

5.4.25 Disks: GATE2009-52 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 operating-system disks normal

5.4.26 Disks: GATE2011-44 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 operating-system disks normal

5.4.27 Disks: GATE2012-41 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 operating-system disks normal

5.4.28 Disks: GATE2013-29 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 <cylinder no., surface no., sector no.> . A file of size 42797 KB is stored in the disk and the starting disk location of the file is $< 1200, 9, 40 >$. 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 operating-system disks normal

5.4.29 Disks: GATE2014-2-20 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 _________.

gate2014-2 operating-system disks numerical-answers normal file-system

5.4.30 Disks: GATE2015-1-48 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__________________

gate2015-1 operating-system disks normal numerical-answers

5.4.31 Disks: GATE2015-2-49 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 _____

gate2015-2 operating-system disks normal numerical-answers

5.4.32 Disks: GATE2018-53 https://gateoverflow.in/204128

Consider a storage disk with 4 platters (numbered as $0, 1, 2$ and 3), 200 cylinders (numbered as $0, 1, \dots, 199$), and 256 sectors per track (numbered as $0, 1, \dots 255$). 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 operating-system disks numerical-answers

5.5 File System (4)

5.5.1 File System: GATE2004-IT-67 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

5.5.2 File System: GATE2008-20 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 file-system operating-system normal

5.5.3 File System: GATE2017-2-08 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

gate2017-2 operating-system file-system normal

5.5.4 File System: GATE2019-42 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

Disk block address= 4B=32 bit
so, no. of address/ disk block= 4KB/4=1024

gate2019 numerical-answers operating-system file-system

5.6 Fork (5)

5.6.1 Fork: GATE2004-IT-64 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

5.6.2 Fork: GATE2005-72 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 operating-system fork normal

5.6.3 Fork: GATE2008-66 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 operating-system fork normal

5.6.4 Fork: GATE2012-8 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 operating-system easy fork

5.6.5 Fork: GATE2019-17 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 numerical-answers operating-system fork

5.7 Inter Process Communication (1)

5.7.1 Inter Process Communication: GATE1997-3.7 https://gateoverflow.in/2238

I/O redirection

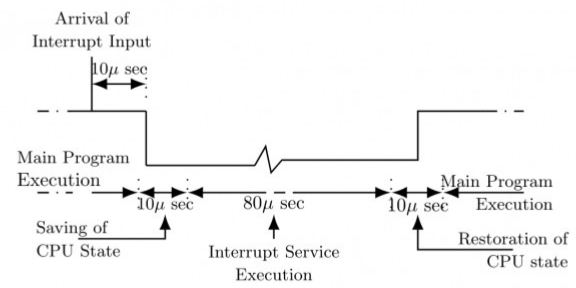
A. implies changing the name of a file
B. can be employed to use an existing file as input file for a program
C. implies connecting 2 programs through a pipe
D. None of the above

gate1997 operating-system normal inter-process-communication

5.8 Interrupts (8)

5.8.1 Interrupts: GATE1993-6.8 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

5.8.2 Interrupts: GATE1997-3.6 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

5.8.3 Interrupts: GATE1997-3.8 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

5.8.4 Interrupts: GATE1998-1.18 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

5.8.5 Interrupts: GATE1999-1.9 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

5.8.6 Interrupts: GATE2001-1.12 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 operating-system interrupts easy

5.8.7 Interrupts: GATE2011-11 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 operating-system interrupts normal

5.8.8 Interrupts: GATE2018-9 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 operating-system interrupts normal

5.9 Io Handling (6)

5.9.1 Io Handling: GATE1996-1.20, ISRO2008-56 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

5.9.2 Io Handling: GATE1998-1.29 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

5.9.3 Io Handling: GATE2004-IT-11, ISRO2011-33 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 $100 \ \mu s$ (including $20 \ \mu s$ 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

5.9.4 Io Handling: GATE2005-19 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 operating-system io-handling normal

5.9.5 Io Handling: GATE2005-20 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 operating-system io-handling normal

5.9.6 Io Handling: GATE2006-IT-8 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

5.10 Memory Management (7)

5.10.1 Memory Management: GATE1992-12-b 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

5.10.2 Memory Management: GATE1995-5 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

5.10.3 Memory Management: GATE1996-2.18 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

5.10.4 Memory Management: GATE2006-IT-56 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

5.10.5 Memory Management: GATE2007-IT-11 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

5.10.6 Memory Management: GATE2014-2-55 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 ________

gate2014-2 operating-system memory-management numerical-answers normal

5.10.7 Memory Management: GATE2015-2-30 https://gateoverflow.in/8145

Consider 6 memory partitions of sizes 200 KB, 400 KB, 600 KB, 500 KB, 300 $\text{KB}$and 250 KB, where KBrefers to kilobyte. These partitions need to be allotted to four processes of sizes 357 KB, 210 KB, 468 KB, 491 KBin 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

gate2015-2 operating-system memory-management easy

5.11 Os Protection (3)

5.11.1 Os Protection: GATE1999-1.11, UGCNET-Dec2015-II-44 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

5.11.2 Os Protection: GATE2001-1.13 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 operating-system normal os-protection

5.11.3 Os Protection: GATE2005-IT-19, UGCNET-June2012-III-57 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?

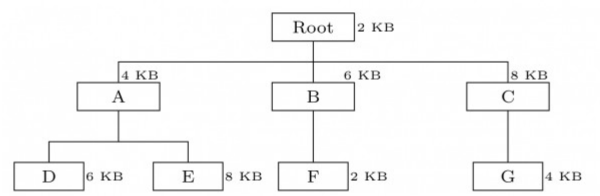
A. User mode
B. Kernel mode
C. Superuser mode
D. Privileged mode

gate2005-it operating-system os-protection normal ugcnetjune2012iii

5.12 Overlay (1)

5.12.1 Overlay: GATE1998-2.16 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 overlay

5.13 Page Replacement (29)

5.13.1 Page Replacement: GATE1993-21 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

5.13.2 Page Replacement: GATE1994-1.13 https://gateoverflow.in/2454

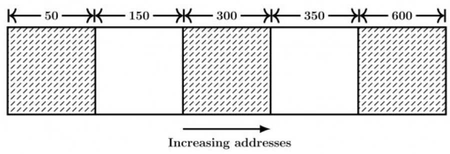
A memory page containing a heavily used variable that was initialized very early and is in constant use is removed then

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

5.13.3 Page Replacement: GATE1994-1.24 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

5.13.4 Page Replacement: GATE1995-1.8 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

5.13.5 Page Replacement: GATE1995-2.7 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

5.13.6 Page Replacement: GATE1997-3.10, ISRO2008-57, ISRO2015-64 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

5.13.7 Page Replacement: GATE1997-3.5 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

5.13.8 Page Replacement: GATE1997-3.9 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

5.13.9 Page Replacement: GATE2001-1.21 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 operating-system page-replacement normal

5.13.10 Page Replacement: GATE2002-1.23 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 operating-system page-replacement easy

5.13.11 Page Replacement: GATE2004-21, ISRO2007-44 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 operating-system virtual-memory page-replacement normal isro2007

5.13.12 Page Replacement: GATE2005-22, ISRO2015-36 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 operating-system page-replacement easy isro2015

5.13.13 Page Replacement: GATE2007-56 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 operating-system page-replacement normal

5.13.14 Page Replacement: GATE2007-82 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 operating-system page-replacement normal

5.13.15 Page Replacement: GATE2007-83 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 normal operating-system page-replacement

5.13.16 Page Replacement: GATE2007-IT-12 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 $\text{x} + 99$ 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

5.13.17 Page Replacement: GATE2007-IT-58 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

5.13.18 Page Replacement: GATE2008-IT-41 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

5.13.19 Page Replacement: GATE2009-9, ISRO2016-52 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 operating-system page-replacement normal isro2016

5.13.20 Page Replacement: GATE2010-24 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 operating-system page-replacement normal

5.13.21 Page Replacement: GATE2012-42 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 operating-system page-replacement normal

5.13.22 Page Replacement: GATE2014-1-33 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__________.

gate2014-1 operating-system page-replacement numerical-answers

5.13.23 Page Replacement: GATE2014-2-33 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 $1, 2, \ldots, 100$ 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

gate2014-2 operating-system page-replacement ambiguous

5.13.24 Page Replacement: GATE2014-3-20 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?

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

gate2014-3 operating-system page-replacement numerical-answers normal

5.13.25 Page Replacement: GATE2015-1-47 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

gate2015-1 operating-system page-replacement normal

5.13.26 Page Replacement: GATE2016-1-49 https://gateoverflow.in/39711

Consider a computer system with ten physical page frames. The system is provided with an access sequence $(a_1, a_2, \ldots, a_{20}, a_1, a_2, \ldots a_{20})$, 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_________.

gate2016-1 operating-system page-replacement normal numerical-answers

5.13.27 Page Replacement: GATE2016-2-20 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)

gate2016-2 operating-system page-replacement easy

5.13.28 Page Replacement: GATE2017-1-40 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 statement:

$S1$: *Random page replacement* algorithm (where a page chosen at random is replaced) suffers from Belady's anomaly.

$S2$: *LRU page replacement* algorithm suffers from Belady's anomaly.

Which of the following is CORRECT?

A. $S1$ is true, $S2$ is true
B. $S1$ is true, $S2$ is false
C. $S1$ is false, $S2$ is true
D. $S1$ is false, $S2$ is false

gate2017-1 page-replacement operating-system normal

5.13.29 Page Replacement: TIFR2013-B-14 https://gateoverflow.in/25794

Assume a demand paged memory system where ONLY THREE pages can reside in the memory at a time. The following sequence gives the order in which the program references the pages.

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

Assume that least frequently used page is replaced when necessary. If there is more than one least frequently used pages then the least recently used page among them is replaced. During the program's execution, how many times will the pages $1, 2, 3$ and 4 be brought to the memory?

A. $2, 2, 2, 2$ times, respectively
B. $1, 1, 1, 2$ times, respectively
C. $1, 1, 1, 1$ times, respectively
D. $2, 1, 2, 2$ times, respectively
E. None of the above

tifr2013 operating-system page-replacement

5.14 Precedence Graph (3)

5.14.1 Precedence Graph: GATE1989-11b 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.

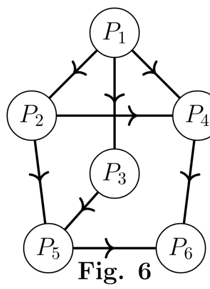


Fig. 6

gate1989 descriptive operating-system precedence-graph process-synchronization

5.14.2 Precedence Graph: GATE1991-01-xii 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

5.14.3 Precedence Graph: GATE1992-12-a https://gateoverflow.in/591

Draw the precedence graph for the concurrent program given below

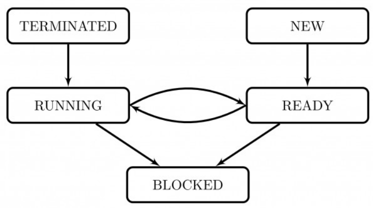
```
S1
parbegin
    begin
        S2:S4
    end;
    begin
        S3;
        parbegin
            S5;
            begin
                S6:S8
            end
       parend
   end;
   S7
parend;
S9
```

gate1992 operating-system normal concurrency precedence-graph

5.15 Process (4)

5.15.1 Process: GATE1996-1.18 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

5.15.2 Process: GATE2001-2.20 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 operating-system easy process

5.15.3 Process: GATE2002-2.21 https://gateoverflow.in/851

Which combination of the following features will suffice to characterize an OS as a multi-programmed OS?

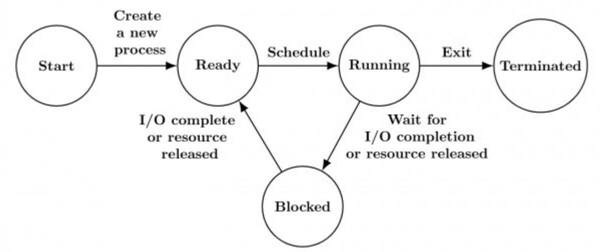
a. More than one program may be loaded into main memory at the same time for execution
b. If a program waits for certain events such as I/O, another program is immediately scheduled for execution
c. If the execution of a program terminates, another program is immediately scheduled for execution.

A. (a) B. (a) and (b) C. (a) and (c) D. (a), (b) and (c)

gate2002 operating-system normal process

5.15.4 Process: GATE2006-IT-13 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

5.16 Process Schedule (37)

5.16.1 Process Schedule: GATE1990-1-vi https://gateoverflow.in/83850

Fill in the blanks:

The highest-response ratio next scheduling policy favours ___________ jobs, but it also limits the waiting time of _________ jobs.

gate1990 operating-system process-schedule

5.16.2 Process Schedule: GATE1993-7.10 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-schedule normal

5.16.3 Process Schedule: GATE1995-1.15 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-schedule easy

5.16.4 Process Schedule: GATE1995-2.6 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-schedule normal

5.16.5 Process Schedule: GATE1996-2.20, ISRO2008-15 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-schedule normal isro2008

5.16.6 Process Schedule: GATE1998-2.17, UGCNET-Dec2012-III-43 https://gateoverflow.in/1690

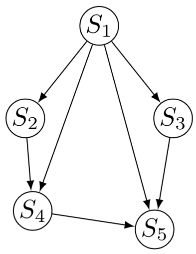
Consider n processes sharing the CPU in a round-robin fashion. Assuming that each process switch takes s seconds, what must be the quantum size q such that the overhead resulting from process switching is minimized but at the same time each process is guaranteed to get its turn at the CPU at least every 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-schedule normal ugcnetdec2012iii

5.16.7 Process Schedule: GATE1998-24 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-schedule descriptive

5.16.8 Process Schedule: GATE1998-7-b 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-schedule normal

5.16.9 Process Schedule: GATE2002-1.22 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 operating-system process-schedule easy

5.16.10 Process Schedule: GATE2003-77 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 operating-system process-schedule normal

5.16.11 Process Schedule: GATE2004-46

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 operating-system process-schedule normal

5.16.12 Process Schedule: GATE2005-IT-60

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-schedule normal

5.16.13 Process Schedule: GATE2006-06, ISRO2009-14

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 operating-system process-schedule normal isro2009

5.16.14 Process Schedule: GATE2006-64

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 operating-system process-schedule normal

5.16.15 Process Schedule: GATE2006-65 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 operating-system process-schedule normal

5.16.16 Process Schedule: GATE2006-IT-12 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-schedule normal

5.16.17 Process Schedule: GATE2006-IT-54 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-schedule normal

5.16.18 Process Schedule: GATE2007-16 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 operating-system process-schedule normal

5.16.19 Process Schedule: GATE2007-55 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 operating-system process-schedule normal

5.16.20 Process Schedule: GATE2007-IT-26 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-schedule normal

5.16.21 Process Schedule: GATE2008-IT-55 https://gateoverflow.in/3365

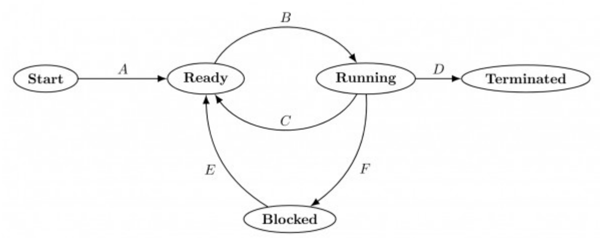
If the time-slice used in the round-robin scheduling policy is more than the maximum time required to execute any process, then the policy will

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-schedule easy

5.16.22 Process Schedule: GATE2009-32 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 operating-system process-schedule normal

5.16.23 Process Schedule: GATE2010-25 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 operating-system process-schedule easy

5.16.24 Process Schedule: GATE2011-35 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 operating-system process-schedule normal

5.16.25 Process Schedule: GATE2012-31 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 operating-system process-schedule normal

5.16.26 Process Schedule: GATE2013-10 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 operating-system process-schedule normal

5.16.27 Process Schedule: GATE2014-1-32 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 ____________________.

gate2014-1 operating-system process-schedule numerical-answers normal

5.16.28 Process Schedule: GATE2014-2-32 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 ___________.

gate2014-2 operating-system process-schedule numerical-answers normal

5.16.29 Process Schedule: GATE2014-3-32 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 ______.

gate2014-3 operating-system process-schedule numerical-answers normal

5.16.30 Process Schedule: GATE2015-1-46 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.

gate2015-1 operating-system process-schedule normal numerical-answers

5.16.31 Process Schedule: GATE2015-3-1 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

gate2015-3 operating-system process-schedule easy

5.16.32 Process Schedule: GATE2015-3-34 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

gate2015-3 operating-system process-schedule normal

5.16.33 Process Schedule: GATE2016-1-20 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

gate2016-1 operating-system process-schedule normal

5.16.34 Process Schedule: GATE2016-2-47 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.

gate2016-2 operating-system process-schedule normal numerical-answers

5.16.35 Process Schedule: GATE2017-1-24 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.

gate2017-1 operating-system process-schedule numerical-answers

5.16.36 Process Schedule: GATE2017-2-51 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 ______

gate2017-2 operating-system process-schedule numerical-answers

5.16.37 Process Schedule: GATE2019-41

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 numerical-answers operating-system process-schedule

5.17 Process Synchronization (59)

5.17.1 Process Synchronization: GATE1987-1-xvi

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

5.17.2 Process Synchronization: GATE1987-8a

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

5.17.3 Process Synchronization: GATE1988-10iib

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 Pi (i=0 or 1) where process Pj (j=1 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

5.17.4 Process Synchronization: GATE1990-2-iii

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

5.17.5 Process Synchronization: GATE1991-11,a

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

5.17.6 Process Synchronization: GATE1991-11,b

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;
```

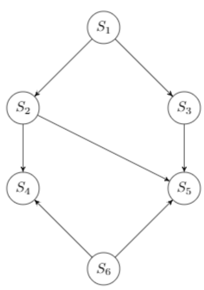
Is there a situation in which a waiting process can never enter the critical section? If so, explain and suggest modifications to the code to solve this problem

gate1991 process-synchronization normal operating-system

5.17.7 Process Synchronization: GATE1993-22

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

5.17.8 Process Synchronization: GATE1994-27

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

5.17.9 Process Synchronization: GATE1995-19 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

5.17.10 Process Synchronization: GATE1996-1.19, ISRO2008-61 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

5.17.11 Process Synchronization: GATE1996-2.19 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

5.17.12 Process Synchronization: GATE1996-21 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

5.17.13 Process Synchronization: GATE1997-6.8 https://gateoverflow.in/2264

Each Process P_i, i = 1....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

5.17.14 Process Synchronization: GATE1997-73 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

5.17.15 Process Synchronization: GATE1998-1.30 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

5.17.16 Process Synchronization: GATE1999-20-a 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

5.17.17 Process Synchronization: GATE1999-20-b 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

5.17.18 Process Synchronization: GATE2000-1.21 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 operating-system process-synchronization normal

5.17.19 Process Synchronization: GATE2000-20 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.

L1:

```
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);
}
```

L2:

```
Writer () {
    wait (mutex);
    if (▭) { _________ (3)
        signal (mutex);
        goto L2;
    }
    W=1;
    signal (mutex);
    ...../*do the write*/
    wait( mutex);
    W=0;
    signal (mutex);
}
```

b. Can the above solution lead to starvation of writers?

gate2000 operating-system process-synchronization normal descriptive

5.17.20 Process Synchronization: GATE2001-2.22 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 operating-system process-synchronization normal

5.17.21 Process Synchronization: GATE2002-18-a 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 operating-system process-synchronization normal descriptive

5.17.22 Process Synchronization: GATE2002-18-b 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 operating-system process-synchronization normal descriptive

5.17.23 Process Synchronization: GATE2002-20 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 operating-system process-synchronization normal descriptive

5.17.24 Process Synchronization: GATE2003-80 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) {          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 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 operating-system process-synchronization normal

5.17.25 Process Synchronization: GATE2003-81 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 operating-system process-synchronization normal

5.17.26 Process Synchronization: GATE2004-48 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 {
$L_1 : \ldots\ldots$	$L_3 : \ldots\ldots$
$L_2 : \ldots\ldots$	$L_4 : \ldots\ldots$
$\text{X} = \text{X} + 1;$	$\text{Y} = \text{Y} + 1;$
$Y = Y - 1;$	$X = Y - 1;$
$V(S_X);$	$V(S_Y);$
$V(S_Y);$	$V(S_X);$
}	}

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 operating-system process-synchronization normal

5.17.27 Process Synchronization: GATE2004-IT-65 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

5.17.28 Process Synchronization: GATE2005-IT-41 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

5.17.29 Process Synchronization: GATE2005-IT-42 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

5.17.30 Process Synchronization: GATE2006-61 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 operating-system process-synchronization normal

5.17.31 Process Synchronization: GATE2006-78 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 operating-system process-synchronization normal

5.17.32 Process Synchronization: GATE2006-79 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 operating-system process-synchronization normal

5.17.33 Process Synchronization: GATE2006-IT-55 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

5.17.34 Process Synchronization: GATE2007-58 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 */

<pre>while (true) {
    wants1 = true;
    while (wants2 == true);
    /* Critical Section */
    wants1 = false;
}
/* Remainder section */</pre>
	<pre>while (true) {
	    wants2 = true;
	    while (wants1 == true);
	    /* Critical Section */
	    wants2=false;
	}
	/* Remainder section */</pre>

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 operating-system process-synchronization normal

5.17.35 Process Synchronization: GATE2007-IT-10 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 flase (ii) is true
B. Both (i) and (ii) are false
C. (i) is true (ii) is flase
D. Both (i) and (ii) are true

gate2007-it operating-system process-synchronization normal

5.17.36 Process Synchronization: GATE2007-IT-56 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

5.17.37 Process Synchronization: GATE2008-IT-53 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

5.17.38 Process Synchronization: GATE2009-33 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 operating-system process-synchronization normal

5.17.39 Process Synchronization: GATE2010-23 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); Critical Section S1 = S2;	while (S1 != S2); Critical Section 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 operating-system process-synchronization normal

5.17.40 Process Synchronization: GATE2010-45 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 (S0);
print '0';
release (S1);
release (S2);
}
	wait (S1);
	release (S0);
		wait (S2);
		release (S0);

How many times will process $P0$ print '0'?

A. At least twice
B. Exactly twice
C. Exactly thrice
D. Exactly once

gate2010 operating-system process-synchronization normal

5.17.41 Process Synchronization: GATE2012-32 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 operating-system process-synchronization normal

5.17.42 Process Synchronization: GATE2013-34 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 operating-system process-synchronization normal

5.17.43 Process Synchronization: GATE2013-39 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);
```

D.
```
ExitX(R, S) {
V(R);
P(S);
```

`}`
`EntryY(R, S) {`
`V(S);`
`P(R);`
`}`
	`}`
	`EntryY(R, S) {`
	`V(S);`
	`P(R);`
	`}`

gate2013 operating-system process-synchronization normal

5.17.44 Process Synchronization: GATE2014-2-31 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.

gate2014-2 operating-system process-synchronization normal

5.17.45 Process Synchronization: GATE2015-1-9 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_______________________.

gate2015-1 operating-system process-synchronization normal numerical-answers

5.17.46 Process Synchronization: GATE2015-3-10 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

gate2015-3 operating-system process-synchronization normal

5.17.47 Process Synchronization: GATE2016-2-48 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.

gate2016-2 operating-system process-synchronization normal

5.17.48 Process Synchronization: GATE2017-1-27 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. $x = 1, y = 1$

gate2017-1 operating-system process-synchronization normal

5.17.49 Process Synchronization: GATE2018-40 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 operating-system process-synchronization normal

5.17.50 Process Synchronization: GATE2019-23 https://gateoverflow.in/302825

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

$P1$	$P2$	$P3$
:	:	:
:	:	:
$D = D + 20$	$D = D - 50$	$D = D + 10$
:	:	:
:	:	:

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

gate2019 numerical-answers operating-system process-synchronization

5.17.51 Process Synchronization: GATE2019-39 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 operating-system process-synchronization

5.17.52 Process Synchronization: TIFR2010-B-28 https://gateoverflow.in/18751

Consider the **concurrent** program:

```
x: 1;
    cobegin
        x:= x + 3 || x := x + x + 2
    coend
```

Reading and writing of variables is atomic, but the evaluation of an expression is not atomic.

The set of possible values of variable x at the end of the execution of the program is:

A. $\{4\}$
B. $\{8\}$
C. $\{4, 7, 10\}$
D. $\{4, 7, 8, 10\}$
E. $\{4, 7, 8\}$

tifr2010 process-synchronization

5.17.53 Process Synchronization: TIFR2010-B-32 https://gateoverflow.in/19244

Consider the following solution (expressed in Dijkstra's guarded command notation) to the mutual exclusion problem.

```
process P1 is
begin
    loop
    Non_critical_section;
    while not (Turn=1) do
        skip od;
        Critical_section_1;
        Turn=2;
    end loop
end
```

||

```
process P2 is
begin
    loop
    Non_critical_section;
    while not (turn=2) do
        skip od;
        Critical_section_2;
        Turn=1;
    end loop
end
```

Initially, Turn $= 1$, Assume that the two process run forever and that no process stays in its critical and non-critical section infinitely. A mutual exclusion program is correct if it satisfies the following requirements.

1. Only one process can be in a critical region at a time.
2. Program is a dead-lock free, i.e., if both processes are trying to enter the critical region then at least one of them does enter the critical region.
3. Program is starvation-free; i.e, a process trying to enter the critical region eventually manages to do so.

The above mutual exclusion solution.

A. Does not satisfy the requirement (1).
B. Satisfy the requirement (1) but does not satisfy the requirement (2).
C. Satisfies the requirements (1) and (2), but does not satisfies the requirement (3).
D. Satisfies the requirement (1) and (3), but does not satisfies the requirement (2).
E. Satisfies all the requirement (1), (2), and (3).

tifr2010 operating-system process-synchronization

5.17.54 Process Synchronization: TIFR2011-B-22 https://gateoverflow.in/20330

Consider the program

```
P:: x:=1; y:=1; z:=1; u:=0
```

And the program

```
Q:: x, y, z, u := 1, 1, 1, 1; u:= 0
```

Which of the following is true?

A. P and Q are equivalent for sequential processors.
B. P and Q are equivalent for all multi-processor models.
C. P and Q are equivalent for all multi-core machines.
D. P and Q are equivalent for all networks of computers.
E. None of the above

tifr2011 operating-system process-synchronization

5.17.55 Process Synchronization: TIFR2011-B-26 https://gateoverflow.in/20572

Consider the following two scenarios in the dining philosophers problem:

i. First a philosopher has to enter a room with the table that restricts the number of philosophers to four.
ii. There is no restriction on the number of philosophers entering the room.

Which of the following is true?

A. Deadlock is possible in (i) and (ii).
B. Deadlock is possible in (i).
C. Starvation is possible in (i).
D. Deadlock is not possible in (ii).
E. Starvation is not possible in (ii)

tifr2011 operating-system process-synchronization

5.17.56 Process Synchronization: TIFR2011-B-28 https://gateoverflow.in/20575

Consider a basic block:

```
x:= a[i]; a[j]:= y; z:= a[j]
```

optimized by removing common sub expression a[i] as follows:

```
x:= a[i]; z:= x; a[j]:= y.
```

Which of the following is true?

A. Both are equivalent.
B. The values computed by both are exactly the same.
C. Both give exactly the same values only if i is not equal to j.
D. They will be equivalent in concurrent programming languages with shared memory.
E. None of the above.

tifr2011 process-synchronization operating-system normal

5.17.57 Process Synchronization: TIFR2011-B-34 https://gateoverflow.in/20838

Consider the class of synchronization primitives. Which of the following is false?

a. Test and set primitives are as powerful as semaphores.
b. There are various synchronizations that can be implemented using an array of semaphores but not by binary semaphores.
c. Split binary semaphores and binary semaphores are equivalent.
d. All statements a - c are false.
e. Petri nets with and without inhibitor arcs have the same power.

tifr2011 operating-system process-synchronization

5.17.58 Process Synchronization: TIFR2012-B-9 https://gateoverflow.in/25109

Consider the concurrent program

```
x := 1;
cobegin
    x := x + x + 1 || x := x + 2
coend;
```

Reading and writing of a variable is atomic, but evaluation of an expression is not atomic. The set of possible values of variable x at the end of execution of the program is

A. $\{3\}$ B. $\{7\}$ C. $\{3, 5, 7\}$ D. $\{3, 7\}$ E. $\{3, 5\}$

tifr2012 process-synchronization operating-system

5.17.59 Process Synchronization: TIFR2015-B-14 https://gateoverflow.in/30077

Consider the following concurrent program (where statements separated by | | with-in cobegin-coend are executed concurrently).

```
x:=1
cobegin
   x:= x + 1 ||    x:= x + 1  ||    x:= x + 1
coend
```

Reading and writing of variables is atomic but evaluation of expressions is not atomic. The set of possible values of x at the end of execution of the program is

A. $\{4\}$
B. $\{2,3,4\}$
C. $\{2,4\}$
D. $\{2,3\}$
E. $\{2\}$

tifr2015 process-synchronization operating-system normal

5.18 Resource Allocation (26)

5.18.1 Resource Allocation: GATE1988-11 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

5.18.2 Resource Allocation: GATE1989-11a 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

5.18.3 Resource Allocation: GATE1992-02-xi https://gateoverflow.in/568

02. Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

(xi) 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

5.18.4 Resource Allocation: GATE1993-7.9, UGCNET-Dec2012-III-41 https://gateoverflow.in/2297

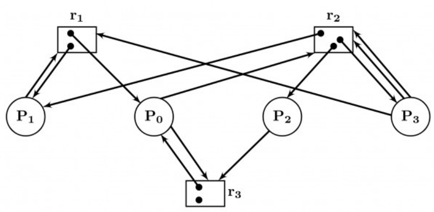
Consider a system having m resources of the same type. These resources are shared by 3 processes A,B, and C which have peak demands of 3, 4 and 6 respectively. For what value of m deadlock will not occur?

A. 7 B. 9 C. 10 D. 13 E. 15

gate1993 operating-system resource-allocation normal ugcnetdec2012iii

5.18.5 Resource Allocation: GATE1994-28 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

5.18.6 Resource Allocation: GATE1996-22 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

5.18.7 Resource Allocation: GATE1997-6.7 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

5.18.8 Resource Allocation: GATE1997-75 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

5.18.9 Resource Allocation: GATE1998-1.32 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

5.18.10 Resource Allocation: GATE2000-2.23 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 operating-system resource-allocation normal

5.18.11 Resource Allocation: GATE2001-19 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 operating-system resource-allocation normal descriptive

5.18.12 Resource Allocation: GATE2005-71 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 operating-system resource-allocation normal

5.18.13 Resource Allocation: GATE2005-IT-62 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

5.18.14 Resource Allocation: GATE2006-66

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 operating-system resource-allocation normal

5.18.15 Resource Allocation: GATE2007-57

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 operating-system resource-allocation normal

5.18.16 Resource Allocation: GATE2008-65

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 operating-system easy resource-allocation

5.18.17 Resource Allocation: GATE2008-IT-54

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

5.18.18 Resource Allocation: GATE2009-30

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 operating-system resource-allocation normal

5.18.19 Resource Allocation: GATE2010-46 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:

$\text{if}(i\%2 == 0)\{ \quad \text{if}(i < n) \text{ request } R_i; \quad \text{if}(i + 2 < n) \text{ request } R_{i+2}; \}\text{else}\{ \quad \text{if}(i < n) \text{ request } R_{n-i}; \quad \text{if}(i + 2 < n) \text{ requ}$

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 operating-system resource-allocation normal

5.18.20 Resource Allocation: GATE2013-16 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 operating-system resource-allocation normal

5.18.21 Resource Allocation: GATE2014-1-31 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.

gate2014-1 operating-system resource-allocation normal

5.18.22 Resource Allocation: GATE2014-3-31 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 _________.

gate2014-3 operating-system resource-allocation numerical-answers easy

5.18.23 Resource Allocation: GATE2015-2-23 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

gate2015-2 operating-system resource-allocation easy

5.18.24 Resource Allocation: GATE2015-3-52 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)

gate2015-3 operating-system resource-allocation normal

5.18.25 Resource Allocation: GATE2016-1-50 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

gate2016-1 operating-system resource-allocation difficult ambiguous

5.18.26 Resource Allocation: GATE2017-2-33 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

gate2017-2 operating-system resource-allocation normal

5.19 Runtime Environments (3)

5.19.1 Runtime Environments: GATE1991-02-iii 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

5.19.2 Runtime Environments: GATE1996-2.17 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

5.19.3 Runtime Environments: GATE2002-2.20 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 operating-system runtime-environments easy

5.20 Semaphore (7)

5.20.1 Semaphore: GATE1990-1-vii https://gateoverflow.in/83851

Fill in the blanks:

Semaphore operations are atomic because they are implemented within the OS ___Kernel___.

gate1990 operating-system semaphore process-synchronization

5.20.2 Semaphore: GATE1992-02,x, ISRO2015-35 https://gateoverflow.in/564

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

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 semaphore easy isro2015

5.20.3 Semaphore: GATE1998-1.31 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 semaphore easy

5.20.4 Semaphore: GATE2006-IT-57 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 semaphore normal

5.20.5 Semaphore: GATE2008-63 https://gateoverflow.in/486

The P and V operations on counting semaphores, where s is a counting semaphore, are defined as follows:

$$P(s) : \begin{array}{l} s = s - 1; \\ \text{If } s < 0 \text{ then wait;} \end{array}$$

$$V(s) : \begin{array}{l} s = s + 1; \\ \text{If } s \leq 0 \text{ then wake up process waiting on s;} \end{array}$$

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:

$$P(s) : \begin{array}{l} P_b(x_b); \\ s = s - 1; \\ \text{if } (s < 0) \\ \{ \\ \quad V_b(x_b); \\ \quad P_b(y_b); \\ \} \\ \text{ else } V_b(x_b); \end{array}$$

$$V(s) : \begin{array}{l} P_b(x_b); \\ s = s + 1; \\ \text{if } (s \leq 0) V_b(y_b); \\ V_b(x_b); \end{array}$$

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 operating-system normal semaphore

5.20.6 Semaphore: GATE2016-2-49 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 _______

gate2016-2 operating-system semaphore normal numerical-answers

5.20.7 Semaphore: TIFR2012-B-10 https://gateoverflow.in/25110

Consider the blocked-set semaphore where the signaling process awakens any one of the suspended process; i.e.,

Wait (S): If $S > 0$ then $S \leftarrow S - 1$, else suspend the execution of this process.

Signal (S): If there are processes that have been suspended on semaphore S, then wake any one of them, else $S \leftarrow S + 1$

Consider the following solution of mutual exclusion problem using blocked-set semaphores.

```
s := 1;
cobegin
P(1)   ||    P(2)     ||    .....    ||     P(N)
coend
```

Where the task body P(i) is

```
begin
while true do
begin
 < non critical section >
 Wait (S)
<critical section>
 Signal (S)
end
```

```
end
```

Here N is the number of concurrent processors. Which of the following is true?

A. The program fails to achieve mutual exclusion of critical regions.
B. The program achieves mutual exclusion, but starvation freedom is ensured only for $N \leq 2$
C. The program does not ensure mutual exclusion if $N \geq 3$
D. The program achieves mutual exclusion, but allows starvation for any $N \geq 2$
E. The program achieves mutual exclusion and starvation freedom for any $N \geq 1$

tifr2012 operating-system process-synchronization semaphore

5.21 Threads (7)

5.21.1 Threads: GATE2004-11 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 operating-system threads normal

5.21.2 Threads: GATE2004-IT-14 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

5.21.3 Threads: GATE2007-17 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 operating-system threads normal

5.21.4 Threads: GATE2011-16, UGCNET-June2013-III-65 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 operating-system threads normal ugcnetjune2013iii

5.21.5 Threads: GATE2014-1-20 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.

gate2014-1 operating-system threads normal

5.21.6 Threads: GATE2017-1-18 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

gate2017-1 operating-system threads

5.21.7 Threads: GATE2017-2-07 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

gate2017-2 operating-system threads

5.22 Virtual Memory (38)

5.22.1 Virtual Memory: GATE1989-2-iv 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

5.22.2 Virtual Memory: GATE1990-1-v https://gateoverflow.in/83833

Fill in the blanks:

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

5.22.3 Virtual Memory: GATE1990-7-b 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

5.22.4 Virtual Memory: GATE1991-03-xi https://gateoverflow.in/525

Choose the correct alternatives (more than one can be correct) and write the corresponding letters only:

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

5.22.5 Virtual Memory: GATE1994-1.21 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

5.22.6 Virtual Memory: GATE1995-1.7 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

5.22.7 Virtual Memory: GATE1995-2.16 https://gateoverflow.in/2628

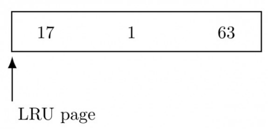
In a virtual memory system the address space specified by the address lines of the CPU must be _____ than the physical memory size and ____ than the secondary storage size.

A. smaller, smaller
B. smaller, larger
C. larger, smaller
D. larger, larger

gate1995 operating-system virtual-memory normal

5.22.8 Virtual Memory: GATE1996-7 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



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

5.22.9 Virtual Memory: GATE1998-2.18, UGCNET-June2012-III-48 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

5.22.10 Virtual Memory: GATE1999-19 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

5.22.11 Virtual Memory: GATE1999-2.10 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

5.22.12 Virtual Memory: GATE1999-2.11 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

5.22.13 Virtual Memory: GATE2000-2.22 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 operating-system easy virtual-memory

5.22.14 Virtual Memory: GATE2001-1.20 https://gateoverflow.in/713

Where does the swap space reside?

A. RAM B. Disk C. ROM D. On-chip cache

gate2001 operating-system easy virtual-memory

5.22.15 Virtual Memory: GATE2001-1.8 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 operating-system virtual-memory normal

5.22.16 Virtual Memory: GATE2001-2.21 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 operating-system virtual-memory normal

5.22.17 Virtual Memory: GATE2002-19 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. An 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 operating-system virtual-memory normal descriptive

5.22.18 Virtual Memory: GATE2003-26 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 operating-system virtual-memory normal

5.22.19 Virtual Memory: GATE2003-78 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 operating-system normal virtual-memory

5.22.20 Virtual Memory: GATE2003-79 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 operating-system normal virtual-memory

5.22.21 Virtual Memory: GATE2004-IT-66 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

5.22.22 Virtual Memory: GATE2006-62, ISRO2016-50 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 operating-system virtual-memory normal isro2016

5.22.23 Virtual Memory: GATE2006-63, UGCNET-June2012-III-45 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 operating-system virtual-memory normal ugcnetjune2012iii

5.22.24 Virtual Memory: GATE2008-67 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 operating-system virtual-memory normal

5.22.25 Virtual Memory: GATE2008-IT-16

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

5.22.26 Virtual Memory: GATE2008-IT-56

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

5.22.27 Virtual Memory: GATE2009-10

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 operating-system virtual-memory easy

5.22.28 Virtual Memory: GATE2009-34

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 operating-system virtual-memory easy

5.22.29 Virtual Memory: GATE2011-20, UGCNET-June2013-II-48

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 operating-system virtual-memory normal ugcnetjune2013ii

5.22.30 Virtual Memory: GATE2013-52

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 operating-system virtual-memory normal

5.22.31 Virtual Memory: GATE2013-53 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 normal operating-system virtual-memory

5.22.32 Virtual Memory: GATE2014-3-33 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 _________.

gate2014-3 operating-system virtual-memory numerical-answers normal

5.22.33 Virtual Memory: GATE2015-1-12 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________________.

gate2015-1 operating-system virtual-memory easy numerical-answers

5.22.34 Virtual Memory: GATE2015-2-25 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 ____.

gate2015-2 operating-system virtual-memory easy numerical-answers

5.22.35 Virtual Memory: GATE2015-2-47 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.

gate2015-2 operating-system virtual-memory normal numerical-answers

5.22.36 Virtual Memory: GATE2016-1-47 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.

gate2016-1 operating-system virtual-memory easy numerical-answers

5.22.37 Virtual Memory: GATE2018-10 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 operating-system virtual-memory normal

5.22.38 Virtual Memory: GATE2019-33

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 operating-system virtual-memory