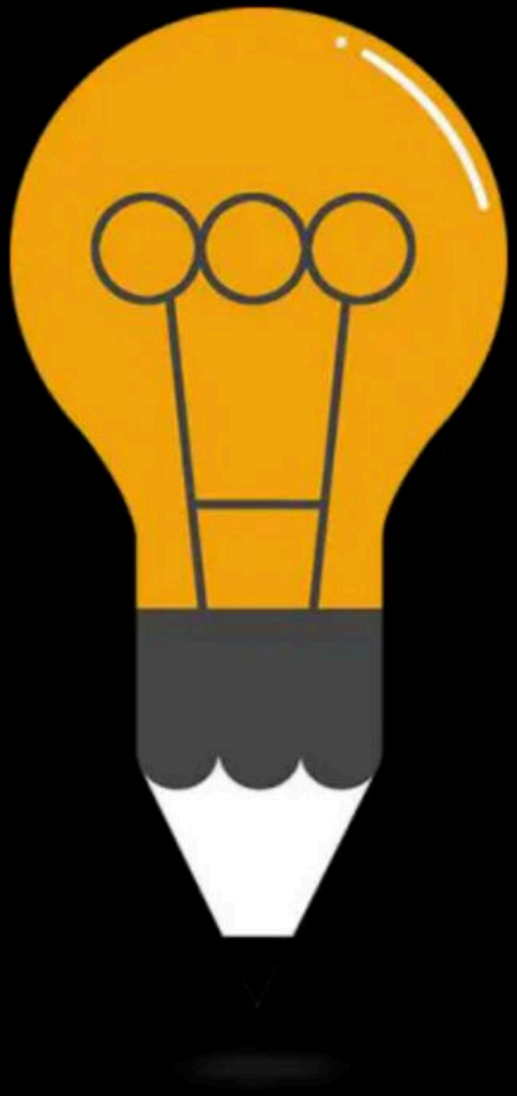
unacademy

Doubt Clearing Session & Page Replacement Algorithm

Comprehensive Course on Operating System for GATE - 2024/25

Vishvadeep Gothi • Lesson 32 • May 25, 2023



Operating System
Doubts &
Page Replacement

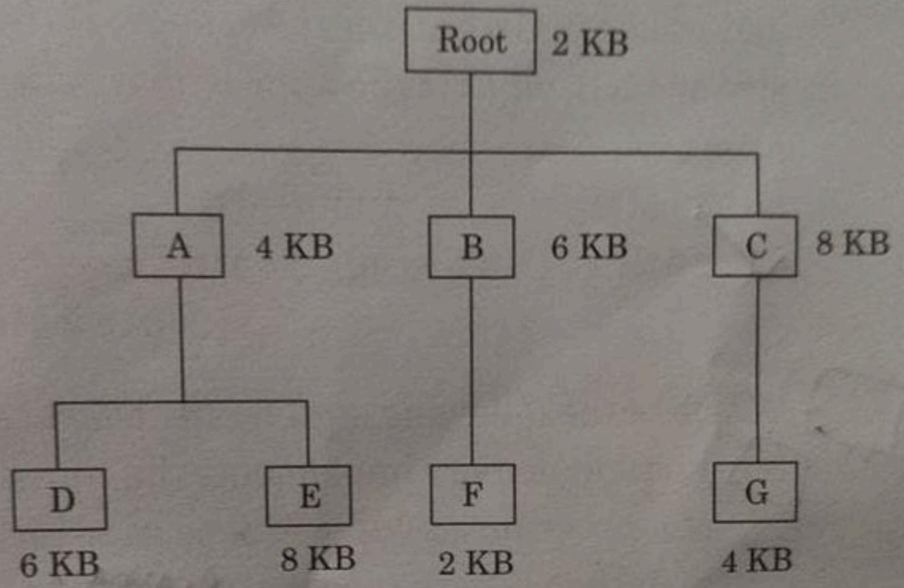
By: Vishvadeep Gothi

unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt

4.18 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?

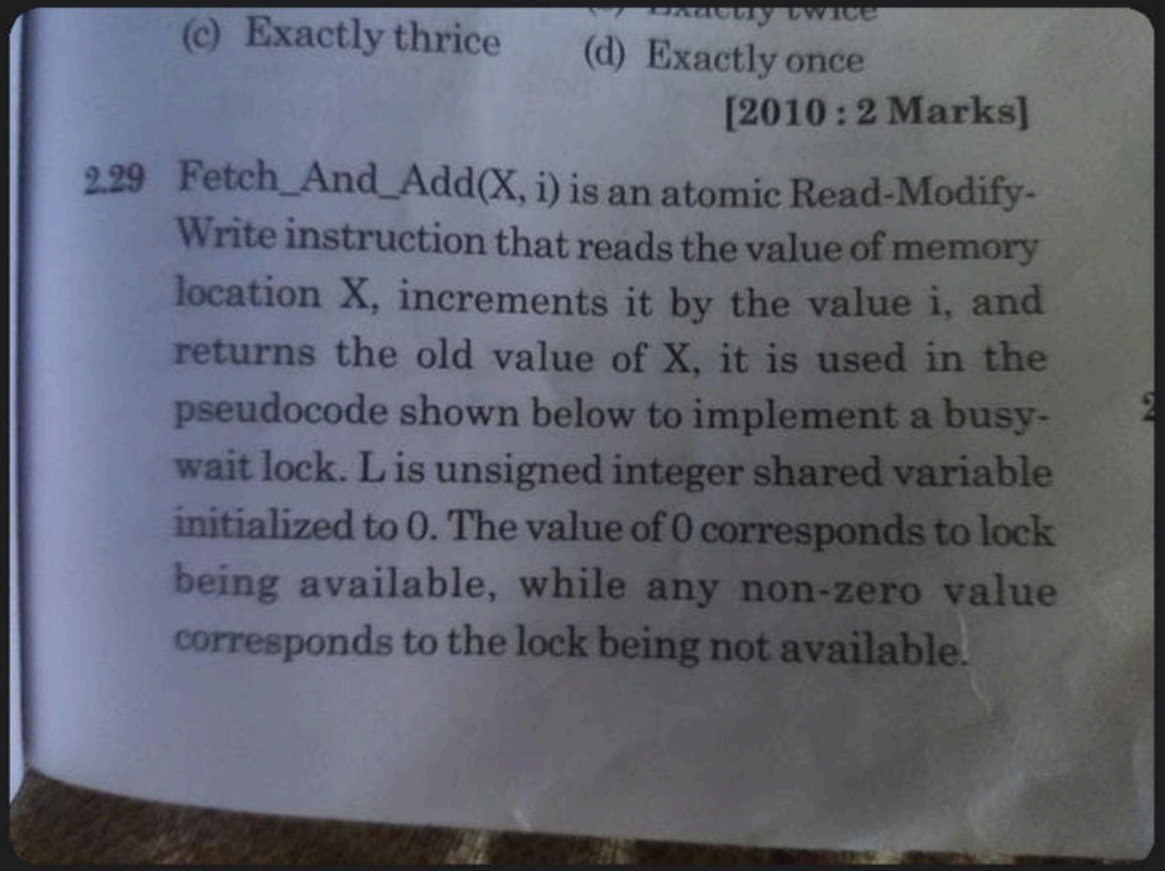
(a) 12 KB
(b) 14 KB
(c) 10 KB
(d) 8 KB

[1998 : 2 Marks]

unacademy

▲ 1 • Asked by Shreyas

Sr i think option A should also be correct



(c) Exactly thrice (d) Exactly once

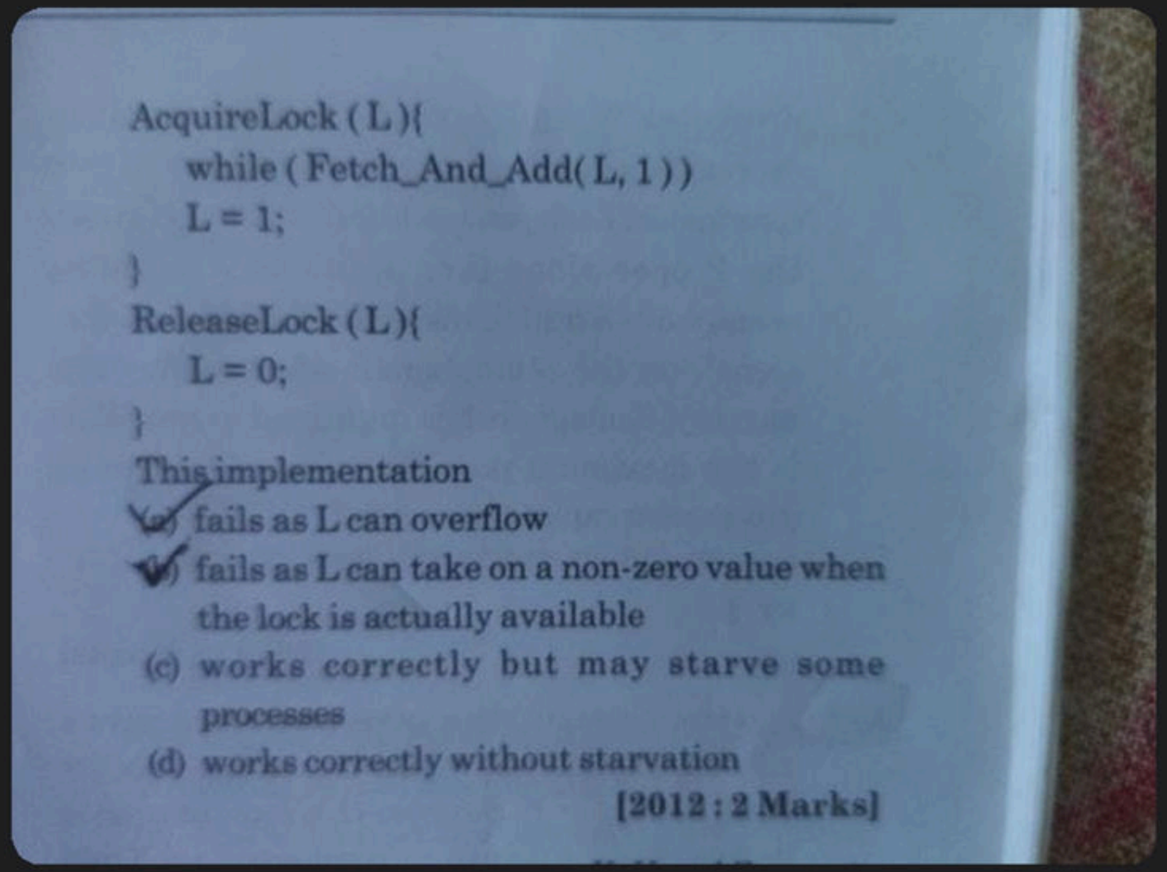
[2010 : 2 Marks]

2.29 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 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.

unacademy

▲ 2 • Asked by Shreyas

Please help me with this doubt



```
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

[2012 : 2 Marks]

unacademy

▲ 1 • Asked by Rishabh

Please help me with this doubt

Is TestAndSet() is stored in ISA of every CPU ?

Where is mode bit and Semaphore variables are stored ?

When some resources are preempted from a process to avoid deadlock, the process is rolled back to a stable state before the preemption of the resources or after the preemption of the resource ?

unacademy

▲ 1 • Asked by Indraneel

Sir I have completed Operating System course from you from youtube and right now I am about to complete COA. But should I study OS from plus again becasue i am not confident in it

unacademy

▲ 1 • Asked by Rishabh

Convert the Infix Expression to Postfix Expression in the order it will be evaluated in the computer

b/e - g*a -> d = c = f

LHS of assignment should not have expressions

unacademy

▲ 1 • Asked by Shreyas

Sir oltion A is given

4.39 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

TLB, PTBR

unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt

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

if $T(P_r) < T(P_h)$ then
kill P_r
else
wait

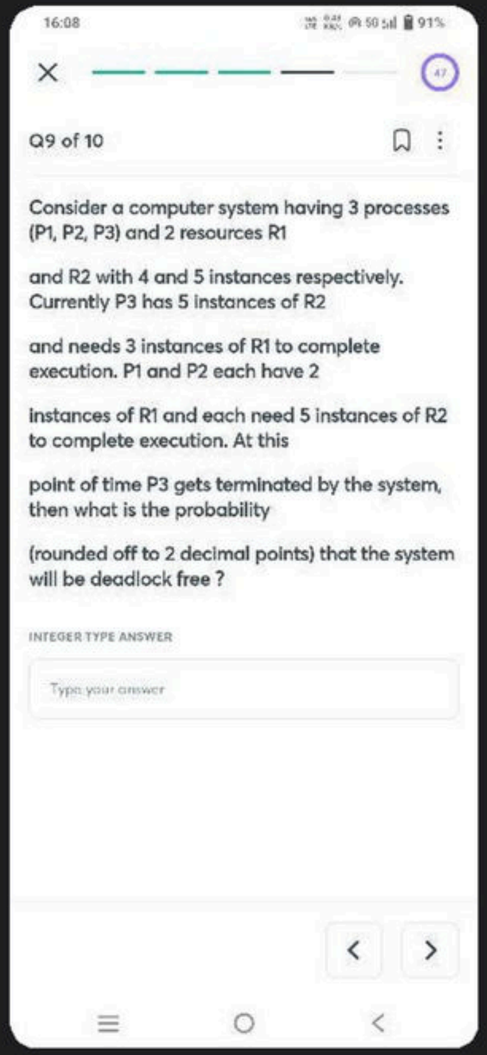
Which one of the following is TRUE?

(a) The scheme is deadlock-free, but not starvation-free
(b) The scheme is not deadlock-free, but starvation-free
(c) The scheme is neither deadlock-free nor starvation-free
(d) The scheme is both deadlock-free and starvation-free

[2004 : 2 Marks]

▲ 1 • Asked by Piyush

Please help me with this doubt



unacademy

▲ 1 • Asked by Piyush

Please help me with this doubt


Q2 of 10

What will be the output of the following code?

```
Int main()
{
int i=1;
fork();
i++;
printf("%d",i);
fork();
i++;
printf("%d",i);
fork();
i++;
printf("%d",i);
}
```

SELECT ONE OPTION

A 2 2 3 3 3 3 4 4 4 4 4 4 4 4

B 2 3 3 2 3 4 4 4 3 4 4 4 4 4

C 2 3 4 2 3 4 4 3 4 4 4 3 4 4

D All of the above three options are correct

unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt

4.40 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.

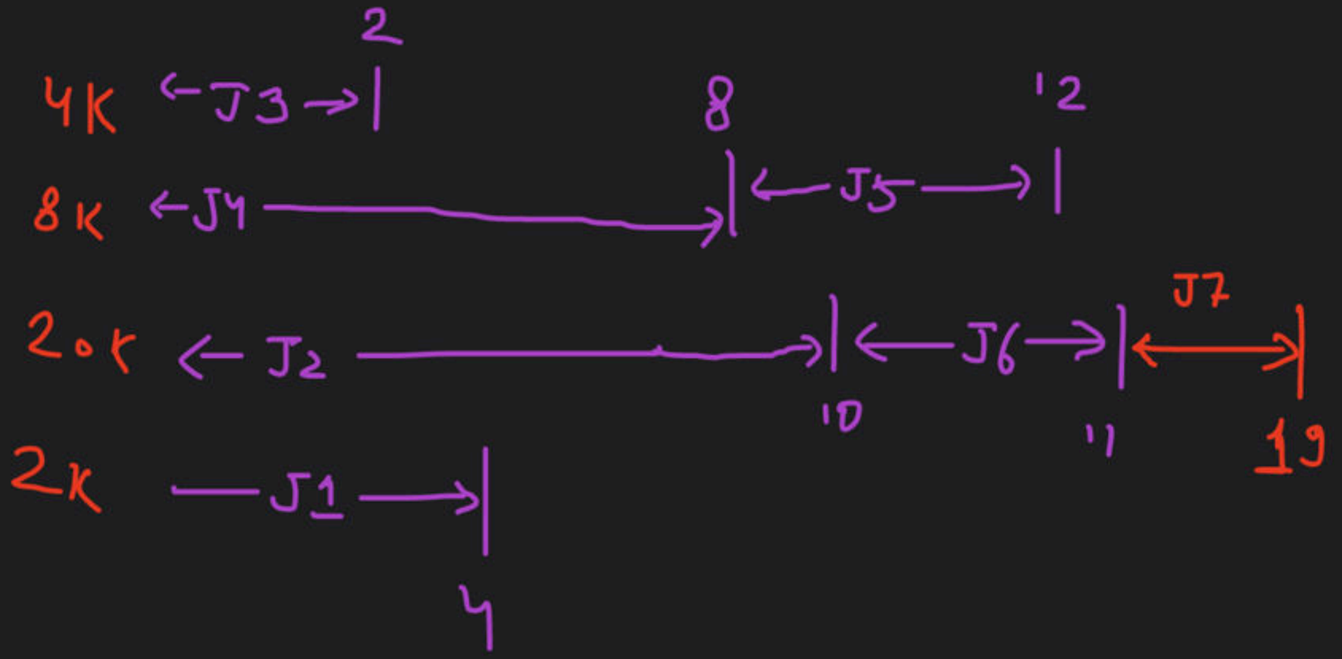
Request No.	Request sizes	Usage Time
J1	2k	4
J2	14k	10
J3	3k	2
J4	6k	8
J5	6k	4
J6	10k	1
J7	7k	8
J8	20k	6

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

(a) 16 (b) 19

(c) 20 (d) 37

[2007 : 1 Mark]



unacademy

▲ 1 • Asked by Jai

sir what if 3 and 4 statement remove

Consider the below code with P() be wait operation and V() be signal operation

```
Code :  function (int p)
    {

          thinking ( ) ;
          P(lock);          .....(i)
          P(fork[p]);
          P(fork [(p+1)modulus5]);
          V(lock);          .....(ii)
          eating ( );
          P(lock);          .....(iii)
          V(fork[p]);
          V(fork(p+1)modulus5);
          V(lock)           .....(iv)

}
```

The above code can be considered as one of the answers to fix the dining philosopher classical problem. lock is a binary semaphore with an initial value be 1.

Which of the following statement is correct?

unacademy

▲ 1 • Asked by Soham

B v crrct hoga na sir?

5.2.3 Deadlock Prevention Avoidance Detection: GATE CSE 2021 Set 2 | Question: 43 top ▸ https://gateoverflow.in/357497

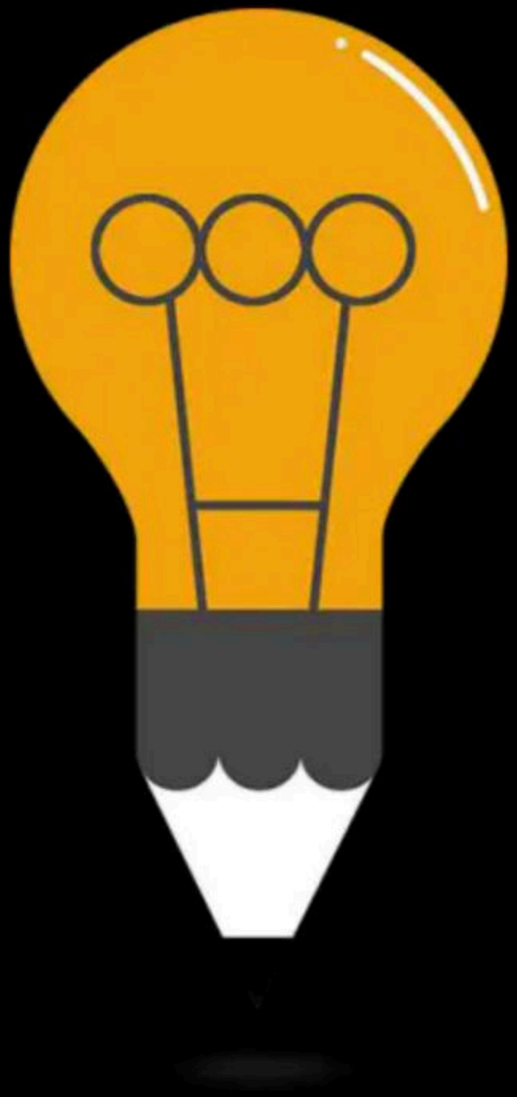
Consider a computer system with multiple shared resource types, with one instance per resource type. Each instance can be owned by only one process at a time. Owning and freeing of resources are done by holding a global lock (L). The following scheme is used to own a resource instance:

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

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

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

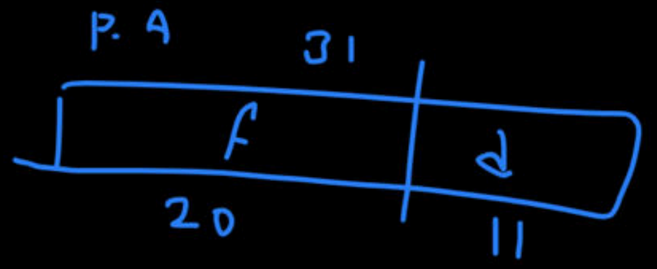
unacademy



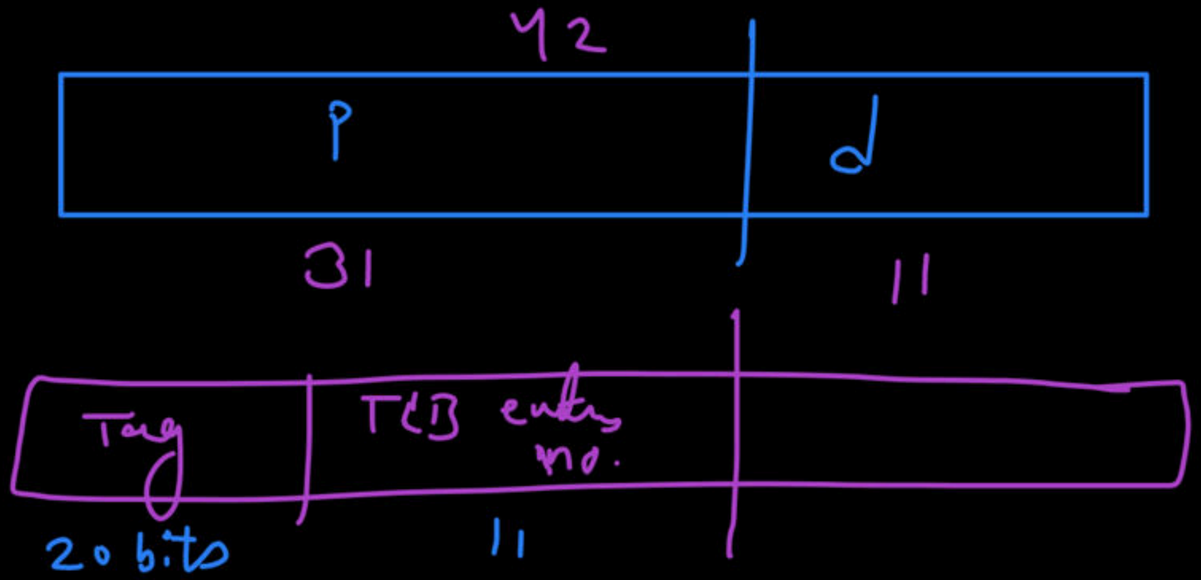
Operating System
DPP

By: Vishvadeep Gothi

Question 6



A computer system implements a 42- bit virtual address, 2GB physical address space, page size of 2KB, and an 8KB look-aside buffer (TLB) organized as direct mapped. Each page table entry contains a valid bit, a dirty bit and 2 protection bits along with the translation. The minimum length of the TLB tag in bits is _____?



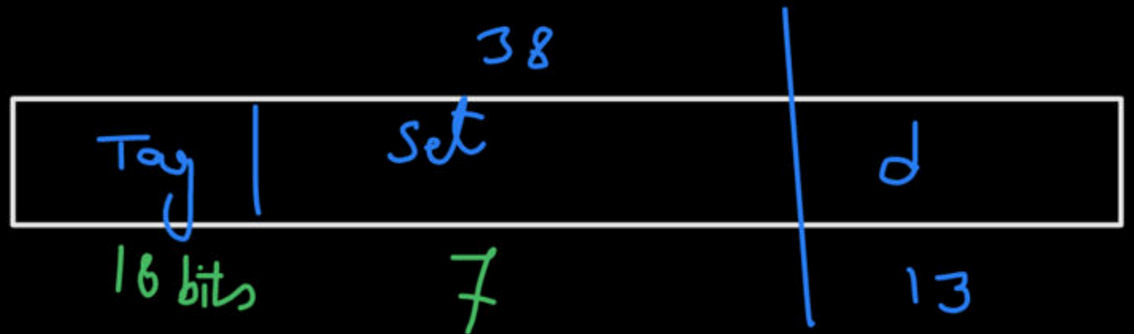
no. of entries in TLB $= \frac{8KB}{4B} = 2K$

1 entry size = 20 + 1 + 1 + 2 bits
= 24 bits = 3 bytes

Question 7

Ans = 18

A computer system implements a 38 bit virtual address, page size of 8 kilobytes, and a 512-entry translation look-aside buffer (TLB) organized into four way set associative manner. The minimum length of the TLB tag in bits is _____?

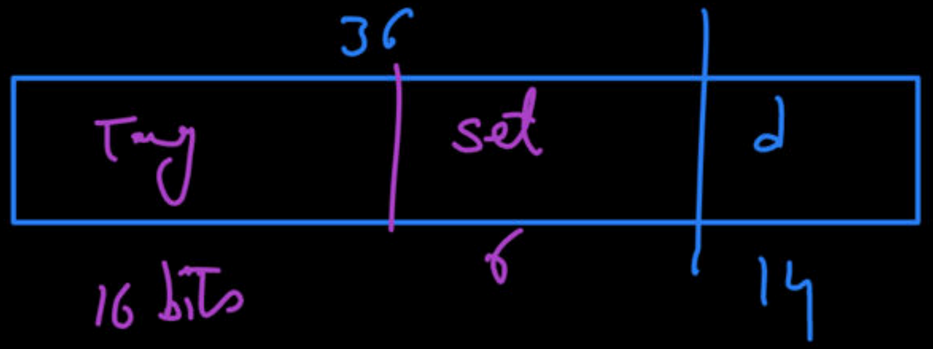


$$\text{no. of sets} = \frac{512}{4} = 128$$

=

Question 8

A Computer system implements a 36-bit virtual address, page size of 16 KBytes and a 256 - entry translation look-aside buffer (TLB) organized into 64 sets each having four ways. Assume that the TLB tag does not store any process id. The minimum length of the TLB tag in bits is _____.



Question 1

When a program tries to access a page that is mapped in address space but not loaded in physical memory, then _____?

a) segmentation fault occurs

b) fatal error occurs

c) page fault occurs

d) no error occurs

Question 2

Effective access time is directly proportional to _____

a) page-fault rate

b) hit ratio

c) memory access time

d) none of the mentioned

Question 3

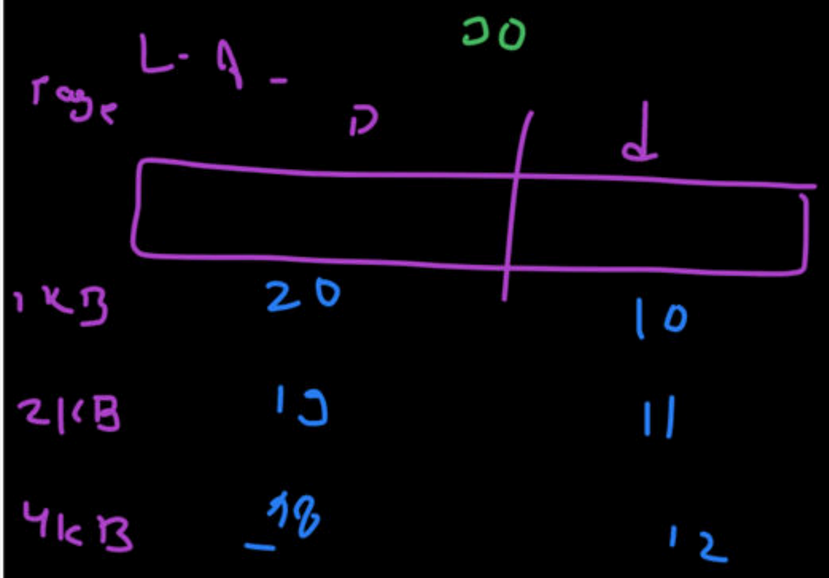
It is advantageous for the page size to be large because:

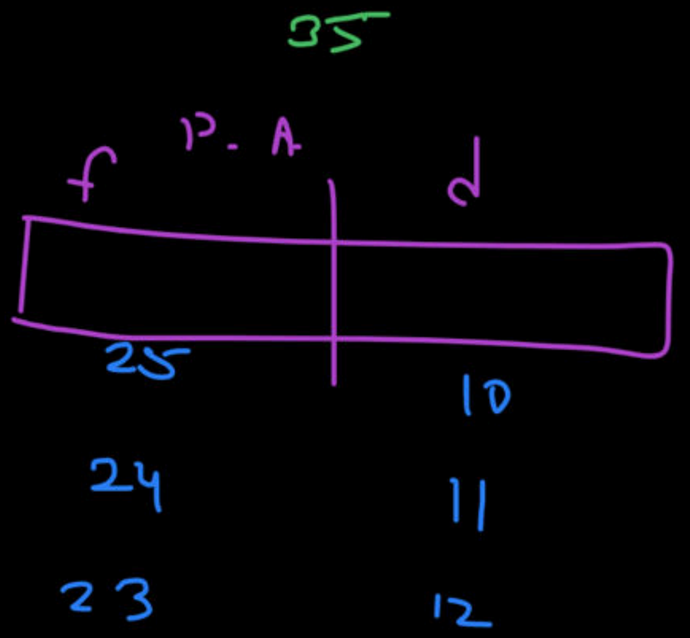
(A) Less unreferenced data will be loaded into memory

(B) Virtual address will be smaller

(C) Page table will be smaller

(D) Large program can be run





PT size

$2^{26} * 25$ bits

$2^{19} * 24$ bits

$2^{18} * 23$ bits

Question 4

It is advantageous for the page size to be small because:

(A) Less unreferenced data will be loaded into memory

(B) Virtual address will be smaller

(C) Page table will be smaller

(D) Large program can be run

Question 5

A demand paged memory environment has physical memory access time of 50 microseconds and page fault service time of 10 milliseconds. If the page fault rate is 5% then the effective memory access time is _____ microseconds?

$= 0.95 * 2 * 50\ \mu sec + 0.05 * (50\ \mu sec + 10000\ \mu sec)$

$= 597.5\ \mu sec$

Question 6

A demand paged memory environment has physical memory access time of 50 microseconds and page fault service time of 5000 microseconds if the replaced page is not dirty. The page fault service time of 100 milliseconds if a dirty page is replaced. Assume that among all pages which are getting replaced, only 2% are dirty, and 95% page hit ratio then the effective memory access time is _____ microseconds?

$$EMAT = 0.95 * 2 * 50\ \mu sec + 0.05\left[50 + 0.98 * 5000 + 0.02 * 100000\right]$$

$$= 442.5\ \mu sec$$

Page Replacement Policies

1. First In First Out (FIFO)
2. Optimal Policy
3. Least Recently Used (LRU)
4. Least Frequently Used (LFU)
5. Most Frequently Used (MFU)
6. Last In First Out (LIFO)
7. Second Chance

First In First Out (FIFO)

Assume:

- Number of frames = 3 (All empty initially)
- Page reference sequence: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

First In First Out (FIFO)

Assume:

- Number of frames = 4 (All empty initially)
- Page reference sequence: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

Belady's Anomaly

First In First Out (FIFO)

Advantages

1. Simple and easy to implement.
2. Low overhead.

Disadvantages:

1. Poor performance.
2. Doesn't consider the frequency of use or last used time, simply replaces the oldest page.
3. Suffers from Belady's Anomaly

Optimal Policy

Assume:

- Number of frames = 3 (All empty initially)
- Page reference sequence: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

Optimal Policy

Advantages

1. Easy to Implement
2. Simple data structures are used
3. Highly efficient

Disadvantages:

1. Requires future knowledge of the program
2. Time-consuming

Least Recently Used (LRU)

Assume:

- Number of frames = 3 (All empty initially)
- Page reference sequence: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

Least Recently Used (LRU)

Advantages

1. Efficient.
2. Doesn't suffer from Belady’s Anomaly

Disadvantages:

1. Complex Implementation
2. Expensive
3. Requires hardware support

Question

- Number of frames = 4 (All empty initially)
- Page reference sequence: 5, 7, 0, 1, 7, 6, 7, 2, 1, 6, 7, 6, 1
- Number of page faults for optimal and LRU policies?

Question

Consider the following page references:

2, 3, 4, 5, 6, 4, 5, 2, 7, 8, 9, 8, 9, 8, 9, 1, 6, 5, 6, 5, 3

Using optimal policy and 4 frames. Memory access time is 2ms and page fault service time is 40ms. The effective memory access time is?

Counting Algorithms

- Counting algorithms look at the number of occurrences of a particular page and use this as the criterion for replacement.

- Such counting algorithms includes:
  - LFU (Least Frequently Used)
  - MFU (Most Frequently Used)

Least Frequently Used (LFU)

Assume:

- Number of frames = 3 (All empty initially)
- Page reference sequence: 1 2 0 3 0 4 2 3 0 3 2

Most Frequently Used (MFU)

Assume:

- Number of frames = 3 (All empty initially)
- Page reference sequence: 1 2 0 3 0 4 2 3 0 3 2

Last In First Out (LIFO)

Assume:

- Number of frames = 3 (All empty initially)
- Page reference sequence: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

Question GATE-2016

Consider a computer system with ten physical page frames. The system is provided with an access sequence a_1, a_2, …, a_{20}, a_1, a_2, …, a_{20} where each a_i 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 _____

Question GATE-2014

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

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

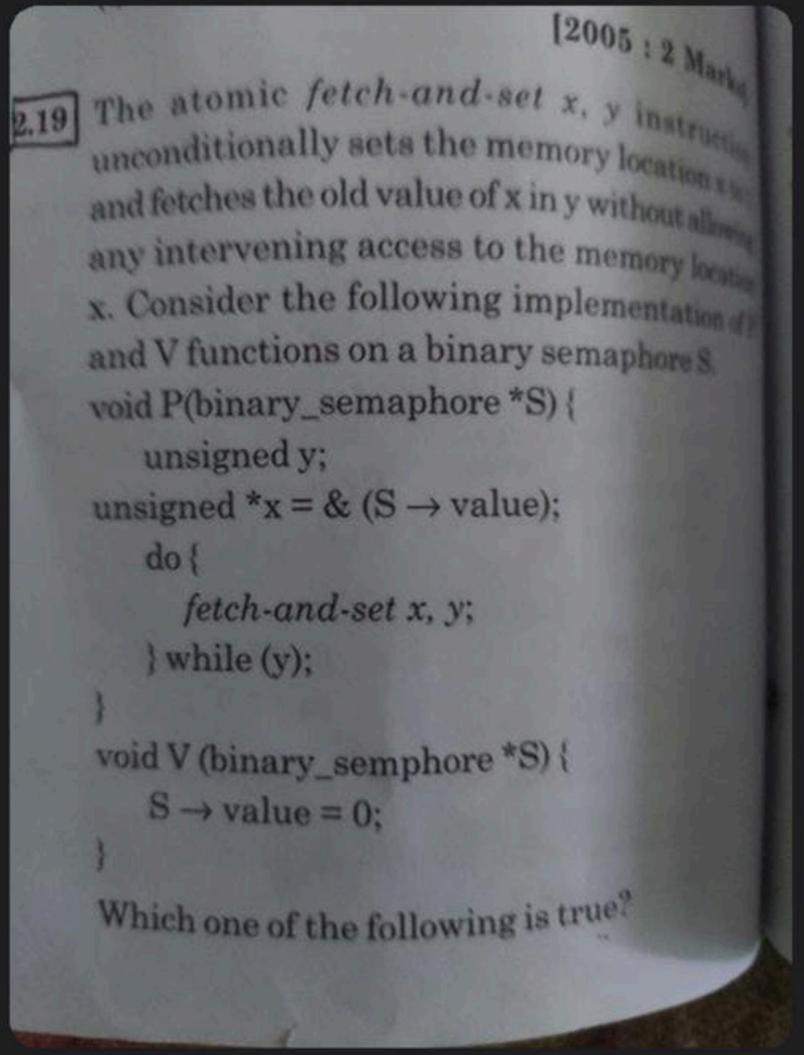
unacademy

Happy Learning.!

unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt



[2005 : 2 Marks]

2.19 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.

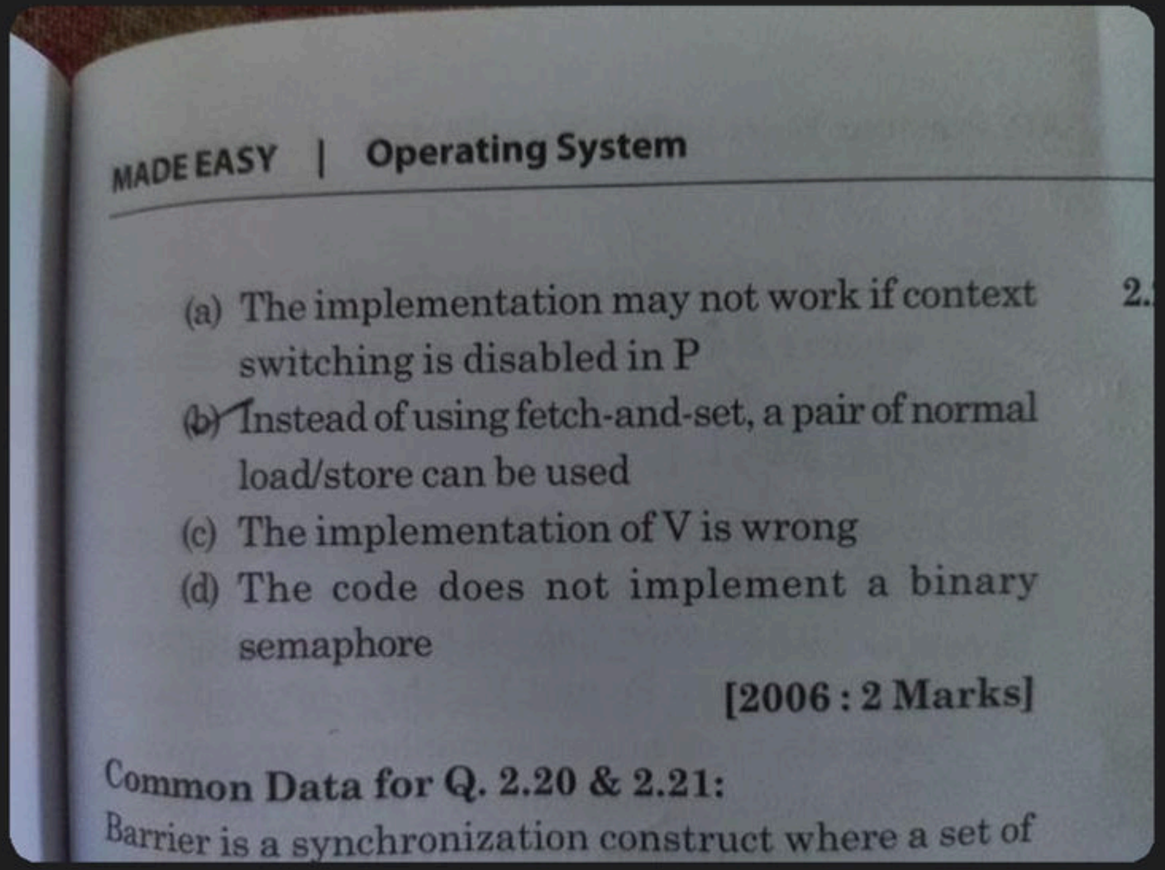
```
void P(binary_semaphore *S) {
    unsigned y;
unsigned *x = & (S → value);
    do {
        fetch-and-set x, y;
    } while (y);
}
void V (binary_semphore *S) {
    S → value = 0;
}
```

Which one of the following is true?

unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt



(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

[2006 : 2 Marks]

Common Data for Q. 2.20 & 2.21:

Barrier is a synchronization construct where a set of