ⓘ Students have either already taken or started taking this quiz, so take care when editing it. If you change any quiz questions in a significant way, you might want to consider re-grading students' quizzes who took the old version of the quiz.

Points 8 ✓ Published ⋮

Details Questions

☑ Show question details

Question 1 pts

To ensure cache coherence means to:

ıswer

○ ensure that all processors in a multiprocessor system have the same view of memory

○ store data, that is frequently accessed, in a cache

○ to write data to main memory immediately after it is changed in the cache

○ ensure that a single cache is used by all processors in a multi-processor system

Question 1 pts

Which of the following is a way to ensure cache coherence?

ıswer

○ Bus snooping

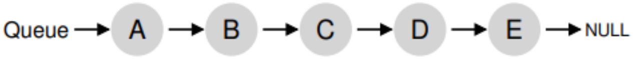
○ Cache affinity

○ Locking

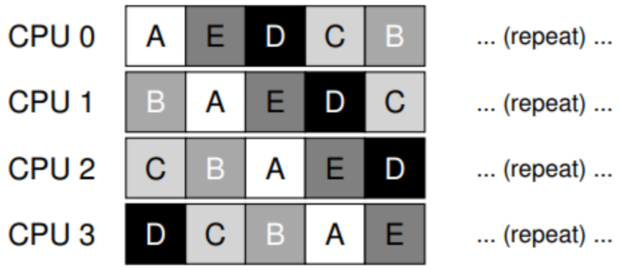
○ single-queue multi-processor scheduling

Question 1 pts

Suppose a single-queue multi-processor scheduler (SQMS) has five jobs in its queue:



to be scheduled for four processors. The scheduler ends up running the jobs as follows:



CPU 0	A	E	D	C	B	... (repeat) ...
CPU 1	B	A	E	D	C	... (repeat) ...
CPU 2	C	B	A	E	D	... (repeat) ...
CPU 3	D	C	B	A	E	... (repeat) ...

What is the problem with this SQMS scheduler?

ıswer

○ The scheduler is not taking into account cache affinity in its scheduling decisions

○ The scheduler is running all jobs in round robin without prioritizing shorter jobs

○ The scheduler is creating a "load imbalance"

○ The scheduler has good throughput but a bad response time

Question 1 pts

Which of the following is NOT true about a multi-queue multi-processor scheduler (MQMS)?

ıswer

○ MQMS always performs better than SQMS (single-queue multi-processor scheduler)

○ MQMS avoids the problem of locking and cache contention

○ MQMS is inherently able to prioritize cache affinity

○ MQMS can sometimes lead to load imbalance

Question 1 pts

Which of the following is NOT true about the address space abstraction for memory?

ıswer

○ It ensures that data addresses used by the program are the same as the physical addresses of the data

○ It allows multiple programs to run concurrently without interfering with each other's memory

○ It allows the operating system to manage memory more efficiently

○ It makes it easier for programmers to write code

Question 1 pts

For the following code segment:

```
#include <stdio.h>
#include <stdlin.h>

int g = 10;

int main(int argc, char** argv)
{
    int x = 3;
    int* p = (int *)malloc(sizeof(int));
    return 0;
}
```

Which of the following is true?

ıswer

○ (1) code segment location: main, (2) data segment location: &g, (3) location of heap: p, (4) location of stack: &x

○ (1) code segment location: &g, (2) data segment location: main, (3) location of heap: p, (4) location of stack: &x

○ (1) code segment location: main, (2) data segment location: &g, (3) location of heap: &x, (4) location of stack: p

○ (1) code segment location: &g, (2) data segment location: main, (3) location of heap: &x, (4) location of stack: p

Question

1 pts

Which of the following is NOT a valid use case for the pipe() system call?

ıswer

- ○ To create a communication channel between two unrelated processes
- ○ To communicate between two related processes
- ○ To implement a producer-consumer pattern
- ○ To implement a command pipeline

Question

1 pts

Which of the following is the correct way to read data from a named pipe with the name /tmp/myfifo?

ıswer

- ○ read("/tmp/myfifo", buffer, sizeof(buffer))
- ○ open("/tmp/myfifo", buffer, sizeof(buffer))
- ○ open("/tmp/myfifo", O_RDONLY)
- ○ read("/tmp/myfifo", O_RDONLY)

+ New question + New question group Find questions

☐ Notify users this quiz has changed

Cancel Save