

Hollow Man

Mutual Exclusion: Hardware Support互斥：硬件支持

本节需要大家理解硬件方案的实现方法和优缺点。

任务点

Roadmap

–Principles of Concurrency并发原理

→Mutual Exclusion: Hardware Support互斥：硬件支持

–Semaphores信号量

–Monitors管程

–Message Passing消息传递

–Readers/Writers Problem读者/写者问题

25

任务点

ch5-2 已完成

作答记录

1 【多选题】

使用如Exchange instruction等专用机器指令解决互斥问题的好处是（）。

(50.0分)

- A、 适用于单处理器或共享内存多处理器中的多进程互斥
- B、 实现让权等待
- C、 简单，易于验证
- D、 可支持多个临界区

正确答案： ACD      我的答案： ACD

目录

讨论

笔记

>

4.3 Multicore and Multithreading...

✔

4.4 Case Studies of threads

1

第5章 Concurrency: Mutual Ex. 

>

5.1 Principals of Concurrency...

3

5.2 Mutual Exclusion: Hardware Support...

1

5.3 Semaphores信号量

6

5.4 Monitors管程

1

5.5 Message Passing消息传递

2

5.6 Readers/Writers Problem...

2

5.7 Dining Philosophers Problem: Deadlock .

>

6.1 Principles of Deadlock死锁...

3

6.2 Deadlock prevention死锁...

2

6.3 Deadlock Avoidance死锁的...

3

6.4 Deadlock detection死锁的...

2

6.5 An Integrated deadlock str...

1

6.6 Dining Philosophers Probl...

1

6.7 Concurrency Mechanisms i...

1

6.8 习题课1

○

第7章 Memory Management内存管理. 

>

7.1 Basic requirements of Me...

1

7.2 Memory Partitioning内存...

3

7.3 Paging分页

1

7.4 Seamentation分段

1

2

【判断题】

多处理器允许并发进程交替执行，所以关中可以实现互斥（）。

(25.0分)

正确答案：×

我的答案：×

3

【判断题】

用专用机器指令解决互斥时，当一个进程离开临界区且多个进程等待时，可能饥饿

(25.0分)

正确答案：√

我的答案：√

上一页

下一页

目录

讨论

笔记

>

4.3 Multicore and Multithreadi...

✓

4.4 Case Studies of threads

1

第5章 Concurrency: Mutual Ex.

^

5.1 Principals of Concurrency...

3

5.2 Mutual Exclusion: Hardwar...

1

5.3 Semaphores信号量

6

5.4 Monitors管程

1

5.5 Message Passing消息传递

2

5.6 Readers/Writers Problem...

2

3 点击开启自动播放模式y: Deadlock .

^

6.1 Principles of Deadlock死锁...

3

6.2 Deadlock prevention死锁...

2

6.3 Deadlock Avoidance死锁的...

3

6.4 Deadlock detection死锁的...

2

6.5 An Integrated deadlock str...

1

6.6 Dining Philosophers Probl...

1

6.7 Concurrency Mechanisms i...

1

6.8 习题课1

○

第7章 Memory Management□.

7.1 Basic requirements of Me...

1

7.2 Memory Partitioning内存...

3

7.3 Paging分页

1

7.4 Seamentation分段

1