

A16C4_Chương 4 - Đồng bộ hóa tiến trình

1. Trong các giải pháp đồng bộ tiến trình sau, giải pháp nào giải quyết được vấn đề truy xuất độc quyền trên các máy tính có một hay nhiều bộ xử lý chia sẻ một vùng nhớ chung?

(1 Point)

☒

Monitor

☒

Semaphone

☐

Sleep and Wakeup

☐

Trao đổi thông điệp

2. Giải pháp Test&Set có giải quyết triệt để bài toán độc quyền truy xuất không khi hệ thống sử dụng nhiều CPU

(1 Point)

☒

Không

☐

Có

4. Phương pháp nhanh nhất để chia sẻ dữ liệu giữa các tiến trình

(1 Point)

☐

Socket

☐

Đường ống

☐

Truyền thông điệp

☒

Vùng nhớ chia sẻ

5. Tiến trình P1 chạy lệnh $x = x + 2$. Tiến trình P2 chạy lệnh $x = x + 5$. Giá trị $x = 1$ dùng chung cho hai tiến trình P1 và P2. Sử dụng nhóm giải pháp Sleep & Wakeup, giá trị x bằng bao nhiêu khi P1 và P2 thực thi xong

(1 Point)

☐

2

☐

6

☐

4

☒

8

☐

3

☐

7

7. Test-and-Setlock(boolean target) { boolean temp = target; target = TRUE; //thiết
lập giá trị mới = True để khóa return temp; //lấy giá trị cũ để kiểm tra
} Cài đặt tiến trình: while (TRUE) { while (Test-and-Setlock(lock)) { //wait
critical-section ();
lock = FALSE; lock=False(không khóa)
Noncritical-section (); } => Chọn phát biểu đúng

(1 Point)

- ☒ Test-and-Setlock(boolean target) là thao tác nguyên tử
- ☐ Vẫn có tình huống hai tiến trình có mặt đồng thời trong đoạn găng nếu sử dụng đoạn mã trên
- ☒ Đoạn mã trên giải quyết bài toán độc quyền truy xuất

8. Câu nào sau đây phát biểu không chính xác:

(1 Point)

- ☐ Tiến trình có thể trao đổi dữ liệu
- ☐ Tiến trình xử lý tín hiệu bằng cách gọi hàm xử lý tín hiệu
- ☐ Tiến trình xử lý tín hiệu theo cách riêng của nó
- ☒ Tiến trình có thể thông báo cho nhau về một sự kiện

9. Trong các giải pháp sau, giải pháp nào tiến trình đang chờ nhưng vẫn chiếm dụng CPU

(1 Point)

- ☒ Test & Set Lock
- ☒ Kiểm tra luân phiên
- ☒ Peterson
- ☐ Monitor
- ☐ Semaphore
- ☐ Sleep and Wakeup

10. Trong các giải pháp đồng bộ tiến trình sau, giải pháp nào vi phạm điều kiện "không có hai tiến trình cùng ở trong miền găng cùng lúc"

(1 Point)

- ☒ Sử dụng biến cờ hiệu
- ☐ Kiểm tra luân phiên
- ☐ Giải pháp Peterson
- ☐ Cấm ngắt
- ☐ Test and Set Lock

12. Nhận xét về cấm ngắt

(1 Point)

- ☐ Có thể cấm ngắt trên nhiều CPU trong hệ thống đa CPU
- ☐ Tiến trình được phép thực thi (chiếm dụng CPU) cho tới khi kết thúc tiến trình.
- ☒ Hệ điều hành không thể ngắt được tiến trình
- ☒ Người dùng không thể tắt được tiến trình trong tác vụ đang chạy

13. Điều kiện nào sau đây không cần thiết khi giải quyết bài toán đoạn găng

(1 Point)

- ☐ Có nhiều hơn 1 tiến trình muốn vào đoạn găng
- ☒ Phải giả thiết về tốc độ tiến trình và số lượng CPU
- ☐ Tiến trình bên ngoài găng không được ngăn cản các tiến trình khác vào đoạn găng
- ☐ Không có tiến trình chờ vô hạn để vào găng

14. Kỹ thuật nào sau đây là liên lạc trực tiếp giữa hai tiến trình

(1 Point)

- ☐ Socket
- ☐ Trao đổi thông điệp
- ☒ Đường ống (Pipe)
- ☐ Vùng nhớ chia sẻ

15. Khi giải quyết bài toán miền găng, điều kiện nào sau đây là không cần thiết:

(1 Point)

- ☐ Không có tiến trình nào phải chờ vô hạn để được vào miền găng
- ☐ Một tiến trình bên ngoài miền găng không được ngăn cản các tiến trình khác vào miền găng.
- ☐ Không có hai tiến trình nào trong miền găng cùng một lúc
- ☒ Phải giả thiết tốc độ các tiến trình cũng như về số lượng bộ xử lý

16. Với giải pháp Test&Set, nếu theo tác Test&SetLock không là thao tác nguyên tử thì

(1 Point)

- ☐ Không cài đặt được thuật toán
- ☐ Không ảnh hưởng gì đến giải pháp
- ☒ Có thể tồn tại nhiều tiến trình đồng thời trong đoạn găng
- ☐ Không cập nhật được biến lock

17. Khi lời gọi Sleep() được gọi bởi tiến trình P1 thì tiến trình P1 sẽ chuyển tới trạng thái nào?

(1 Point)

- ☐ Block
- ☐ Terminated
- ☐ Running
- ☐ New
- ☐ Ready
- ☒ Waitting

18. Thế nào là một thao tác nguyên tử trong giải thuật Test & Set Lock?

(1 Point)

- ☐ Mỗi một lệnh trong hàm Test&SetLock là một thao tác nguyên tử.
- ☒ Không được phép ngắt khi hàm Test&SetLock đang chạy.
- ☐ Thao tác rất nhỏ, nhỏ như nguyên tử.

19. Chọn phát biểu đúng về nhóm giải pháp sleep-wakeup

(1 Point)

- ☐ Khi chưa đủ điều kiện vào đoạn găng, tiến trình ở trạng thái ready
- ☐ Mọi phát biểu đều đúng
- ☐ Khi chưa đủ điều kiện vào đoạn găng, tiến trình chuyển sang ready
- ☒ Tiến trình đang sleep cần một tiến trình khác đánh thức.
- ☒ Khi chưa đủ điều kiện vào đoạn găng, tiến trình chuyển sang trạng thái Waiting

20. Đây là các giải pháp trong nhóm giải pháp busy-waiting?

(1 Point)

- ☒ Kiểm tra luân phiên
- ☒ Peterson
- ☒ Biến cờ hiệu
- ☐ Monitor

21. Phương pháp nhanh nhất để trao đổi thông tin giữa các tiến trình:

(1 Point)

- ☐ Đường ống
- ☐ Trao đổi thông điệp
- ☒ Vùng nhớ chia sẻ
- ☐ Socket

22. Trong các giải pháp sau, giải pháp nào mà khi tiến trình kiểm tra khi thấy có tiến trình khác ở trong đoạn găng sẽ chuyển sang trạng thái chờ?

(1 Point)

- ☐ Giải pháp Peterson
- ☐ Test & Set Lock
- ☐ Kiểm tra luân phiên
- ☒ Semaphore
- ☒ Monitor
- ☒ Sleep and Wakeup

23. Tiến trình chỉ có trạng thái running khi ở trong đoạn găng?

(1 Point)

- ☐ Đúng
- ☒ Sai