



hđh
47 Questions

NAME : _____

CLASS : _____

DATE : _____

1. Đoạn mã nào trong các tiến trình có thể gây ra lỗi khi được thực thi đồng thời?

☐ A Critical Section.

☐ B Remainder Section.

☐ C Entry Section.

☐ D Exit Section

2. Đồng bộ hoá (Process Synchronization) là công việc cần phải áp dụng cho loại tiến trình nào?

☐ A Tiến trình hệ thống (System process).

☐ B Tiến trình cộng tác (Cooperating process).

☐ C Tiến trình người dùng (User process).

☐ D Tiến trình độc lập (Independent process)

3. Đoạn mã nào được sử dụng để kiểm soát quá trình đồng bộ?

☐ A Entry section.

☐ B Program code.

☐ C Critical section.

☐ D Remainder section.

4. Đoạn mã nào có thể chạy cùng lúc mà không gây ra sai sót dữ liệu?

☐ A Critical section.

☐ B Remainder section.

☐ C Program code

☐ D Entry section.

5. Biến số đơn nguyên (atomic variable) là gì?

☐ A Các thao tác lên biến số này tuần tự được thực thi trong CPU.

☐ B Biến số chỉ có ý nghĩa địa phương, sử dụng nội bộ trong tiểu trình.

☐ C Biến số chỉ chứa duy nhất một kiểu dữ liệu được định nghĩa trước.

☐ D Các thao tác lên biến số này được song song thực hiện trong CPU

6. Giải thuật Peterson sử dụng các biến số điều khiển nào để giải quyết bài toán đồng bộ giữa hai tiến trình?

☐ A boolean flag[2];

☐ B choosing[i] = true và int number[i];

☐ C int sync = 2

☐ D boolean flag[2] và int turn;

7. Một tiến trình Px thực hiện thao tác signal() trên một biến số Semaphore n thì có tác dụng gì?

☐ A n++ và sau đó nếu n <= 0 thì wake_up() tiến trình Px.

☐ B n++ và sau đó nếu n <= 0 thì wake_up() tiến trình đang bị blocked.

☐ C n++ và sau đó nếu n > 0 thì wake_up() một tiến trình đang bị blocked.

☐ D n++ và sau đó nếu n > 0 thì wake_up() tiến trình Px.

8. Giải thuật / Phương pháp nào sau đây chỉ có thể giải quyết đồng bộ không nhiều hơn 2 tiến trình?

☐ A Giải thuật Peterson.

☐ B Giải thuật Banker.

☐ C Phương pháp Semaphore.

☐ D Phương pháp Hàng rào bộ nhớ.

9. Một tiến trình Px thực hiện thao tác wait() trên một biến số Semaphore n thì có tác dụng gì?

☐ A n-- và sau đó nếu n < 0 thì block() tiến trình Px.

☐ B n-- và sau đó nếu n >= 0 thì block() tiến trình Px.

☐ C n-- và sau đó nếu n <= 0 thì block() các tiến trình khác Px.

☐ D n++ và sau đó nếu n <= 0 thì block() tiến trình Px.

10. Tình trạng cạnh tranh (Race condition) là gì?

☐ A Tiến trình không cho phép các tiến trình khác tác động lên biến số của nó, và dẫn đến việc đồng bộ thất bại.

☐ B Khi nhiều hơn một tiến trình thao tác lên dữ liệu chia sẻ, kết quả cuối cùng phụ thuộc vào thứ tự thực thi của các thao tác đó.

☐ C Các lệnh cấp thấp (là mã máy) được thực thi đồng thời trong một chu kỳ lệnh của CPU làm sai sót dữ liệu

☐ D Người sử dụng yêu cầu chạy 02 tiến trình có tranh chấp dữ liệu, gây nên hiện tượng tắc nghẽn cho hệ thống.

11. Kỹ thuật đồng bộ sử dụng Semaphore giải quyết được vấn đề gì mà giải thuật Peterson chưa làm được?
- ☐ A Progress (Tính tiến triển). ☐ B Bounded-Waiting (Chờ vô hạn định).
- ☐ C Busy-waiting (Chờ đợi bận rộn). ☐ D Mutual Exclusion (Loại trừ tương hỗ)
12. Yêu cầu về tính sống còn (liveness) của các giải pháp đồng bộ đảm bảo điều gì cho hệ thống?
- ☐ A Sự chờ đợi bận rộn (Busy waiting) không xuất hiện với mọi tiến trình ☐ B Hệ thống đang xử lý các tiến trình có hiệu năng khai thác cao
- ☐ C Các tiến trình luôn tiến triển, tài nguyên không cạn kiệt. ☐ D Dữ liệu luôn được đồng bộ và không có sai sót khi cập nhật.
13. Mục đích của việc sử dụng Semaphore là gì?
- ☐ A Trị số của Semaphore cho biết số tiến trình tối đa được vào hệ thống. ☐ B Thông tin của Semaphore phục vụ cho bài toán đồng bộ tiến trình.
- ☐ C Semaphore là tín hiệu ngắt gửi cho hệ điều hành khi cần đồng bộ tiến trình. ☐ D Trị số của Semaphore cho biết process nào đang được thực thi.
14. Phương pháp Hàng rào bộ nhớ (Memory Barrier) được hiện thực ra sao?
- ☐ A Các tiến trình được cấp các bản sao vùng nhớ chia sẻ để thao tác cập nhật. ☐ B Việc cập nhật vùng nhớ chia sẻ được quyết định bởi tiến trình cấp phát hàng rào.
- ☐ C Các câu lệnh thay đổi biến số chia sẻ cần được nhìn thấy bởi mọi tiến trình khác. ☐ D Các vùng nhớ chia sẻ cần được nhìn thấy bởi tất cả tiến trình đang đồng bộ.
15. "Critical Section" mô tả đoạn mã như thế nào trong một tiến trình?
- ☐ A Đoạn mã có yêu cầu nhập xuất dữ liệu từ thiết bị ngoại vi. ☐ B Đoạn mã có yêu cầu tính toán và sử dụng toàn bộ CPU.
- ☐ C Đoạn mã hệ điều hành tự thêm vào trong tiến trình ☐ D Đoạn mã có chứa những thao tác lên biến dùng chung.

16. "Entry / Exit Section" là đoạn mã gì?

- | | | | |
|----------------------------|---|----------------------------|--|
| <input type="checkbox"/> A | Đoạn mã có chứa những thao tác lên biến dùng chung. | <input type="checkbox"/> B | Đoạn mã hệ điều hành thêm vào trước và sau đoạn mã nguy cơ (Critical section). |
| <input type="checkbox"/> C | Đoạn mã có chứa lệnh can thiệp vào hoạt động của hệ điều hành | <input type="checkbox"/> D | Đoạn mã có yêu cầu tính toán và sử dụng toàn bộ CPU. |

17. Semaphore được hiện thực như thế nào?

- | | | | |
|----------------------------|---|----------------------------|---|
| <input type="checkbox"/> A | Biến số nguyên hoặc nhị phân, kèm theo 2 thao tác wait() và signal(). | <input type="checkbox"/> B | Biến số nguyên hoặc nhị phân, kèm theo 2 thao tác block() và wake_up(). |
| <input type="checkbox"/> C | Mảng các số nguyên hoặc nhị phân, kèm theo 2 thao tác wait() và signal(). | <input type="checkbox"/> D | Mảng các số nguyên hoặc nhị phân, kèm theo 2 thao tác block() và wake_up(). |

18. Bài toán "Bộ đệm giới hạn" (Bounded Buffer) đề cập đến vấn đề chính yếu gì?

- | | | | |
|----------------------------|---|----------------------------|---|
| <input type="checkbox"/> A | Gửi và nhận gói tin qua bộ nhớ chia sẻ có kích thước nhất định. | <input type="checkbox"/> B | Khoá chặn truy cập chỉ của một vài tiến trình đang thực thi. |
| <input type="checkbox"/> C | Tranh chấp tài nguyên giữa nhiều tiến trình trong lúc thực thi | <input type="checkbox"/> D | Bảo mật thông tin khi gửi và nhận thông điệp giữa các tiến trình. |

19. Bài toán "Bộ ghi – Bộ đọc" (Writers and Readers) đề cập đến vấn đề chính yếu gì?

- | | | | |
|----------------------------|--|----------------------------|--|
| <input type="checkbox"/> A | Phân phối dữ liệu từ nhiều tiến trình nguồn đến nhiều tiến trình đích. | <input type="checkbox"/> B | Bảo mật thông tin khi chia sẻ thông tin giữa các tiến trình đang thực thi. |
| <input type="checkbox"/> C | Dữ liệu chia sẻ mà chỉ một vài tiến trình mới có nhu cầu cập nhật dữ liệu. | <input type="checkbox"/> D | Gửi và nhận gói tin qua bộ nhớ chia sẻ có kích thước nhất định. |

20. Bài toán "Triết gia ăn tối" (Dining Philosophers) đề cập đến vấn đề chính yếu gì?

- | | | | |
|----------------------------|--|----------------------------|--|
| <input type="checkbox"/> A | Bảo mật thông tin chia sẻ thông tin giữa nhiều tiến trình với nhau. | <input type="checkbox"/> B | Tranh chấp các tài nguyên chia sẻ riêng biệt giữa từng cặp tiến trình |
| <input type="checkbox"/> C | Chia sẻ tài nguyên thành nhiều thực thể để đáp ứng cho nhiều tiến trình. | <input type="checkbox"/> D | Hiệu suất sử dụng tài nguyên trong hệ thống chạy song song nhiều tiến trình. |

21. Bài toán "Bộ đệm giới hạn" (Bounded Buffer) có thể giải quyết bằng bao nhiêu biến số semaphore?
- ☐ A Chỉ cần 2 biến: full và empty. ☐ B 1 mảng sem[5].
- ☐ C Duy nhất biến số n. ☐ D 3 biến: mutex, full và empty.
22. Bài toán "Bộ ghi – Bộ đọc" (Writers and Readers) có đặc trưng gì?
- ☐ A Hệ thống chỉ có một bộ đọc và rất nhiều bộ ghi. ☐ B Hệ thống chỉ có một bộ ghi và rất nhiều bộ đọc.
- ☐ C Các bộ đọc mới có thể cập nhật dữ liệu chia sẻ. ☐ D Tất cả bộ đọc và bộ ghi cần xếp hàng để thực thi.
23. Bài toán "Triết gia ăn tối" (Dining Philosophers) nếu sử dụng semaphore thì chúng được khởi tạo như thế nào?
- ☐ A semaphore chopstick[5], tất cả phần tử gán bằng 1. ☐ B semaphore chopstick, khởi tạo giá trị 5.
- ☐ C semaphore chopstick[5], tất cả phần tử gán bằng 1. ☐ D semaphore chopstick[5], các phần tử gán lần lượt từ 1 đến 5.
24. API POSIX cung cấp nhiều công cụ đồng bộ, nhưng không bao gồm công cụ nào sau đây?
- ☐ A Biến số semaphore ☐ B Dispatcher objects.
- ☐ C Biến số điều kiện (condition variable). ☐ D Khoá mutex lock
25. Bài toán "Bộ ghi – Bộ đọc" (Writers and Readers) các biến số được khởi tạo như thế nào?
- ☐ A semaphore rw_mutex = 0, mutex = 1; int read_count = 0; ☐ B semaphore rw_mutex = 1, mutex = 2; int read_count = 0; s
- ☐ C semaphore rw_mutex = 1, mutex = 1; int read_count = 2; ☐ D semaphore rw_mutex = 1, mutex = 1; int read_count = 0;

26. Bài toán "Triết gia ăn tối" (Dining Philosophers) có thể giải quyết bằng phương pháp nào để tránh bị tắc nghẽn (deadlock)?

- ☐ A Các khoá mutex_lock áp dụng cho từng vùng tranh chấp.
- ☐ B Giải thuật Peterson với các vòng lặp kiểm tra while().
- ☐ C Các biến số semaphore với các lệnh wait() và signal().
- ☐ D Bộ quan sát (Monitor) với các lệnh test().

27. Bài toán "Bộ ghi – Bộ đọc" (Writers and Readers) có biến thể thứ 2, nó khác gì với biến thể đầu tiên?

- ☐ A Các bộ đọc có thể thực thi song song mà không sai sót dữ liệu.
- ☐ B Nếu một bộ đọc mới đến, nó sẽ được thực thi sớm nhất có thể.
- ☐ C Số lượng bộ đọc và bộ ghi bị giới hạn để tránh cạn kiệt tài nguyên.
- ☐ D Nếu một bộ ghi mới đến, nó sẽ được thực thi sớm nhất có thể.

28. Cho hai tiến trình P1 và P2 quyền tác động lên biến semaphore chia sẻ S và Q (đều có khởi tạo = 1). Các lệnh sau đây lần lượt được thực thi, hệ thống sẽ diễn tiến như thế nào?

Time	P1	P2
t = 1	wait(S)	
t = 2		wait(Q)
t = 3	wait(Q)	
t = 4		wait(S)
t = 5	signal(S)	
t = 6		signal(Q)
t = 7	signal(Q)	
t = 8		signal(S)

- ☐ A Hệ thống sẽ đảm bảo P2 hoàn tất trước P1.
- ☐ B Hệ thống sẽ đảm bảo P1 hoàn tất trước P2.
- ☐ C Hệ thống sẽ chạy hết tất cả lệnh đã nêu.
- ☐ D Hệ thống sẽ rơi vào trạng thái Deadlock.

29. Trong đó biến mutex là biến toàn cục dùng chung (shared variable). Phát biểu nào sau đây là đúng với hệ thống nêu trên?

Question 18	
1/3 pts	
Chọn đúng một câu 2 yêu trình chương trình	
P1:	P2:
(các lệnh khác)	(các lệnh khác)
wait(mutex);	wait(mutex);
critical section	critical section
signal(mutex);	signal(mutex);
(các lệnh khác)	(các lệnh khác)

- ☐ A Với khởi tạo mutex = 0; chỉ có 1 tiến trình được vào critical section.
- ☐ B Với khởi tạo mutex = 1; P2 phải gửi tín hiệu đến P1 để xin vào critical section.
- ☐ C Với khởi tạo mutex = 2; P2 chắc chắn sẽ vào critical section trước.
- ☐ D Với khởi tạo mutex = 1; chỉ có 1 tiến trình được vào critical section.

30.

Question 19
0 / 3 pts
Cho đoạn mã của 2 tiến trình P1 và P2 như sau:

P1: (các lệnh khác) <code>signal(mutex);</code> <code>func_1();</code> (các lệnh khác)	P2: (các lệnh khác) <code>wait(mutex);</code> <code>func_2();</code> (các lệnh khác)
---	---

Trong đó biến mutex là biến toàn cục dùng chung (Shared variable) Chọn phát biểu đúng.

- | | | | |
|----------------------------|--|----------------------------|--|
| <input type="checkbox"/> A | Để đảm bảo hàm func_2() chạy trước func_1(), khởi tạo mutex = 2. | <input type="checkbox"/> B | Để đảm bảo hàm func_1() chạy trước func_2(), khởi tạo mutex = 1. |
| <input type="checkbox"/> C | Để đảm bảo hàm func_1() chạy trước func_2(), khởi tạo mutex = 0. | <input type="checkbox"/> D | Để đảm bảo hàm func_2() chạy trước func_1(), khởi tạo mutex = 0. |

31. Deadlock là trạng thái như thế nào?

- | | | | |
|----------------------------|--|----------------------------|--|
| <input type="checkbox"/> A | Mọi tiến trình đều đang yêu cầu sử dụng CPU. | <input type="checkbox"/> B | Các tiến trình không thể tiếp tục được thực thi. |
| <input type="checkbox"/> C | Hệ thống không thể tiếp tục hoạt động | <input type="checkbox"/> D | Hệ điều hành đóng băng các tiến trình. |

32. Đồ thị RAG của một hệ thống mô tả cho n tiến trình và m tài nguyên thì có bao nhiêu đỉnh (lực lượng của tập đỉnh V)?

- | | | | |
|----------------------------|----------------------|----------------------------|---------|
| <input type="checkbox"/> A | tối thiểu là $n + m$ | <input type="checkbox"/> B | n |
| <input type="checkbox"/> C | $n + m$ | <input type="checkbox"/> D | $n * m$ |

33. Mục tiêu của giải thuật “Nhà băng” (Banker) là gì?

- | | | | |
|----------------------------|---|----------------------------|---|
| <input type="checkbox"/> A | Chỉ ra một thứ tự thực thi các tiến trình sau khi hệ thống bị deadlock. | <input type="checkbox"/> B | Chỉ ra một thứ tự thực thi của các tiến trình sao cho hệ thống luôn an toàn |
| <input type="checkbox"/> C | Tìm ra thứ tự nạp vào hệ thống các chương trình mà người dùng yêu cầu. | <input type="checkbox"/> D | Tìm ra những chuỗi không an toàn trong hệ thống để phòng trừ. |

34. Trạng thái mà một hệ thống máy tính có các tiến trình vẫn hoạt động nhưng thời gian đáp ứng rất lâu là gì?

- | | | | |
|----------------------------|--------------|----------------------------|------------|
| <input type="checkbox"/> A | Low resource | <input type="checkbox"/> B | Unsafe |
| <input type="checkbox"/> C | Deadlock | <input type="checkbox"/> D | Starvation |

35. Quan hệ giữa “an toàn” và “deadlock” được diễn đạt như thế nào?

- ☐ A Hệ thống sẽ bị deadlock khi nó có trạng thái không an toàn
- ☐ B Hệ thống chỉ có thể bị deadlock khi nó có trạng thái không an toàn.
- ☐ C Hệ thống vẫn có thể bị deadlock khi nó đang an toàn.
- ☐ D An toàn và deadlock là 2 khái niệm cùng chỉ 1 trạng thái

36. Thứ tự của quy trình yêu cầu cấp phát tài nguyên là:

- ☐ A use – request – release.
- ☐ B request – use – release
- ☐ C release – use – request.
- ☐ D release – request – use.

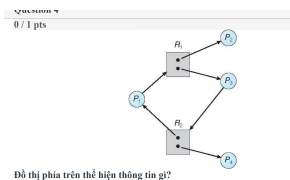
37. Yếu tố nào sau đây không phải là một đặc trưng của Deadlock?

- ☐ A Giữ và chờ (Hold and wait).
- ☐ B Hệ thống thiếu thốn tài nguyên (Starvation)
- ☐ C Loại trừ tương hỗ (Mutual Exclusion).
- ☐ D Không thể chiếm lại tài nguyên (No preemption).

38. Phát biểu nào sau đây SAI về đồ thị cấp phát tài nguyên

- ☐ A Tài nguyên có thể có nhiều thực thể
- ☐ B Tập đỉnh V gồm có 2 loại là tiến trình và tài nguyên
- ☐ C Đồ thị có chu trình thì hệ thống bị deadlock
- ☐ D Đồ thị không có chu trình thì hệ thống không bị deadlock.

39.



- ☐ A Tiến trình P1 đã sẵn sàng vào CPU.
- ☐ B Hệ thống có 4 tiến trình và 2 loại thực thể.
- ☐ C Tiến trình P1 đã có đủ tài nguyên nó cần.
- ☐ D Hệ thống có 4 tiến trình và 2 loại tài nguyên.

40.

Question 8
0/2 pts
Cho hệ thống có 5 tiến trình và 4 loại tài nguyên: A, B, C và D. Áp dụng giải thuật Banker (Nhả hàng).

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	2	0	0	1	2	1	5	2	0	
P1	1	0	0	1	1	7	5	2				
P2	1	5	4	2	5	5	5	5				
P3	0	6	3	2	0	6	5	3				
P4	0	0	3	4	0	0	5	0				

Hệ thống này có bao nhiêu đơn vị tài nguyên mỗi loại?

A

3 thực thể A, 14 thực thể B, 12 thực thể C và 12 thực thể D.

B

3 thực thể A, 22 thực thể B, 21 thực thể C và 16 thực thể D.

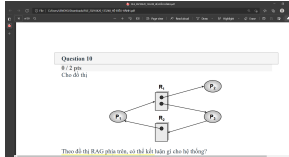
C

2 thực thể A, 9 thực thể B, 10 thực thể C và 12 thực thể D.

D

2 thực thể A, 14 thực thể B, 10 thực thể C và 12 thực thể D.

41.



A

P1 chỉ có thể chạy tiếp khi P2 kết thúc

B

P3 đang chiếm giữ một thực thể của tài nguyên R2

C

Hệ thống đã bị deadlock.

D

Chuỗi an toàn: P2 – P3 – P1

42. Khi hệ thống xảy ra deadlock, hệ điều hành phải chọn một tiến trình (nạn nhân) để kết thúc. Tính chất nào sau đây sẽ KHÔNG được quan tâm?

A

Tiến trình nạn nhân cần bao nhiêu tài nguyên để có thể chạy tiếp.

B

Thời gian mà tiến trình nạn nhân đã vận hành và tiếp tục cần để chạy.

C

Trạng thái deadlock của hệ thống là do tiến trình nào gây ra.

D

Tiến trình nạn nhân là độc lập (interactive) hay theo bó (batch).

43. Cho tập cạnh E của một đồ thị RAG như sau : $E = \{(P1,R1),(R1,P2),(P2,R2),(P3,R1),(R2,P3)\}$ chọn phát biểu đúng :

A

RAG trên không bị deadlock..

B

RAG trên là RAG vòng.

C

RAG trên không bị vòng.

D

RAG trên bị deadlock.

44. Một hệ thống có n tiến trình và m loại tài nguyên và đang ở trạng thái Deadlock. Lúc đó, nếu chạy giải thuật “Phát hiện deadlock” (Detection Algorithm) thì độ phức tạp là

A

 $O(n)$

B

 $O(1)$

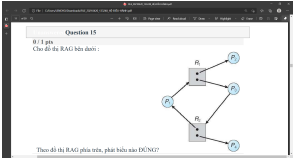
C

 $O(m * n)$

D

 $O(m * n * n)$

45.



A

P2 đang yêu cầu một thực thể của tài nguyên R1.

B

P1 đang chiếm giữ một thực thể của tài nguyên R2.

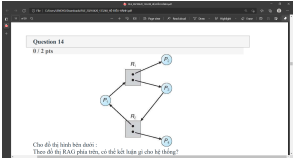
C

P3 đang yêu cầu thực thể đã bị P4 chiếm giữ.

D

P1 đang chiếm giữ toàn bộ tài nguyên R1.

46.



A

Hệ thống đã bị tắc nghẽn.

B

Chuỗi an toàn: P4 – P1 – P2 – P3.

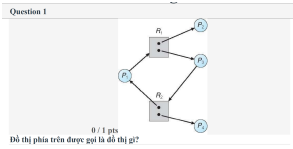
C

P1 chỉ có thể chạy tiếp khi P2 hoặc P3 kết thúc.

D

Deadlock đang xảy ra với P1 và P3.

47.



A

Request Allowed Graph

B

Request and Allocation

C

Resource Allocation Graph.

D

Resource and Process.

Answer Key

1. a	2. b	3. a	4. b
5. a	6. d	7. b	8. a
9. a	10. b	11. c	12. c
13. b	14. c	15. d	16. b
17. a	18. a	19. c	20. b
21. d	22. c	23. a, c	24. b
25. d	26. d	27. d	28. d
29. d	30. c	31. b	32. c
33. b	34. d	35. b	36. b
37. b	38. c	39. d	40. a
41. a	42. c	43. b	44. d
45. b	46. c	47. c	