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

Critical Section

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

Tiến trình cộng tác (Cooperating process)

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

Entry Section

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

Remainder section

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

Các thao tác lên biến số này tuần tự được thực thi 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?

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 dụnggì?

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

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

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

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

Giải thuật Peterson.

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

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 thao tác đó

11 .Kỹ thuật đồng bộ sử dụng Semaphore giải quyết được vấn đề gì mà giải thuật Petersonchưa làm được?

Busy-waiting (Chờ đợi bận rộn).

12 .Yêu cầu về tính sống còn (liveness) của các giải pháp đồng bộ đảm bảo điều gì cho hệthống?

Các tiến trình luôn tiến triển, tài nguyên không cạn kiệt.

13 .Mục đích của việc sử dụng Semaphore là gì?

Thông tin của Semaphore phục vụ cho bài toán đồng bộ tiến trình.

14 .Phương pháp Hàng rào bộ nhớ (Memory Barrier) được hiện thực ra sao?

Các câu lệnh thay đổi biến số chia sẻ cần được nhìn thấy bởi mọi tiến trình khác.

15 .“Critical Section” mô tả đoạn mã như thế nào trong một tiến trình?

Đoạn mã có chứa những thao tác lên biến dùng chung.

16 .“Entry / Exit Section” là đoạn mã gì?

Đoạn mã hệ điều hành thêm vào trước và sau đoạn mã nguy cơ (Critical section).

17 .Cho hai tiến trình P1 và P2 quyền tác động lên biến semaphore chia sẻ S và Q (đều cókhởi 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)

Hệ thống sẽ rơi vào trạng thái Deadlock.

18 .Cho đoạn mã của 2 tiến trình như sau:

```
P1:
(các lệnh khác)
wait(mutex);
  critical section
signal(mutex);
(các lệnh khác)
```

```
P2:
(các lệnh khác)
wait(mutex);
  critical section
signal(mutex);
(các lệnh khác)
```

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

Với khởi tạo mutex = 1; chỉ có 1 tiến trình được vào critical section.

19 .Cho đoạn mã của 2 tiến trình P1 và P2 như sau:

```
P1:
(các lệnh khác)
signal(mutex);
func_1( );
(các lệnh khác)
```

```
P2:
(các lệnh khác)
wait(mutex);
func_2( );
(các lệnh khác)
```

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

Để đảm bảo hàm func_1() chạy trước func_2(), khởi tạo mutex = 0.

20 .Semaphore được hiện thực như thế nào?

Biến số nguyên hoặc nhị phân, kèm theo 2 thao tác wait() và signal().

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

Gửi và nhận gói tin qua bộ nhớ chia sẻ có kích thước nhất định.

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

Dữ liệu chia sẻ mà chỉ một vài tiến trình mới có nhu cầu cập nhật dữ liệu.

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

Tranh chấp các tài nguyên chia sẻ riêng biệt giữa từng cặp tiến trình.

24 .Bài toán “Bộ đệm giới hạn” (Bounded Buffer) có thể giải quyết bằng bao nhiêu biến sốsemaphore?

3 biến: mutex, full và empty.

25 .Bài toán “Bộ ghi – Bộ đọc” (Writers and Readers) có đặc trưng gì?

Các bộ đọc mới có thể cập nhật dữ liệu chia sẻ.

27 .API POSIX cung cấp nhiều công cụ đồng bộ, nhưng không bao gồm công cụ nào sauđây?

Dispatcher objects.

26 .Bài toán “Triết gia ăn tối” (Dining Philosophers) nếu sử dụng semaphore thì chúng đượckhởi tạo như thế nào?

semaphore chopstick[5], tất cả phần tử gán bằng 1.

28 .Bài toán “Bộ ghi – Bộ đọc” (Writers and Readers) các biến số được khởi tạo như thế nào?

semaphore rw_mutex = 1, mutex = 1; int read_count = 0;

29 .Bài toán “Triết gia ăn tối” (Dining Philosophers) có thể giải quyết bằng phương pháp nàođể tránh bị tắc nghẽn (deadlock)?

Bộ quan sát (Monitor) với các lệnh test().

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

Nếu một bộ ghi mới đến, nó sẽ được thực thi sớm nhất có thể.

CHƯƠNG 9 + 10

1.Trong cơ chế phân trang bộ nhớ của hệ điều hành, khái niệm trang (Page) là gì? *

Đơn vị phân hoạch trong không gian tiến trình

2. Hiện tượng “Phân mảnh nội” xảy ra với những vùng trống bộ nhớ nào

Thừa ra do tiến trình xin cấp phát nhiều hơn nhu cầu thật sự.

3. Thời điểm nào có thể “ánh xạ” địa chỉ chương trình vào địa chỉ bộ nhớ vật lý? Chọn mọiphương án đúng. *

Complie
Load
Excution

4.Cấp phát bộ nhớ theo phương pháp “Phân trang” có đặc trưng nào sau đây?. *

Kích thước một trang (page) và một khung trang (frame) bằng nhau.

5. Hiện tượng các phần nhỏ không sử dụng trong bộ nhớ được tạo thành từ nhiều lần cấp phátvà giải phóng vùng bộ nhớ gọi là hiện tượng gì?. *

Phân mảnh (Defragment)

6. Phân mảnh ngoại có thể giải quyết bằng biện pháp nào? *

Chia bộ nhớ chính thành các khung trang (frame).

7. Không gian địa chỉ của một quá trình có kích thước 4GBytes. Số lượng bit cần dùng đểđánh địa chỉ là bao nhiêu? *

32.0

8. Phân mảnh ngoại sẽ xảy ra khi giải thuật tìm lỗ trống nào sau đây được áp dụng? *

A. Phân mảnh ngoại luôn xảy ra, bất kể áp dụng giải thuật tìm lỗ trống nào.

9. Cho bộ nhớ chứa các vùng nhớ không liên tục với độ lớn theo thứ tự sau: 10KB, 4KB,20KB, 18KB, 7KB, 9KB, 13KB, và 15KB. Một tiến trình xin cấp phát 12 KB, và được nạpvào vùng nhớ 13KB. Giải thuật chọn lỗ trống trong bộ nhớ nào đã được áp. *

Best – fit

10.Trong hệ thống quản lý bộ nhớ bằng phương pháp phân trang, hai tiến trình có thể liên lạcvới nhau, dưới mô hình bộ nhớ chia sẻ được hiện thực bằng phương pháp nào?. *

Sử dụng chung 1 khung trang trong bộ nhớ vật lý.

11. Trong kỹ thuật phân vùng nhớ kích thước cố định, độ đa lập trình của hệ thống là baonhiêu? *

bằng số phân vùng đã chia.

12. Thông tin chứa trong Bảng phân trang là dùng để: *

D. Lưu thông tin vị trí nạp các trang của tiến trình trong bộ nhớ chính.

13. Hệ điều hành và các tiến trình cần được bảo vệ, tránh bị chỉnh sửa từ các tiến trình đangchạy, bằng cách nào? *

A. địa chỉ sinh ra từ CPU được kiểm tra có thuộc vùng nhớ hợp lệ hay không

14. Chọn phát biểu sai về phương pháp quản lý bộ nhớ *

C. Phân hoạch tuỳ biến kích thước phân vùng (variable partitions) không bị phân mảnh ngoại.

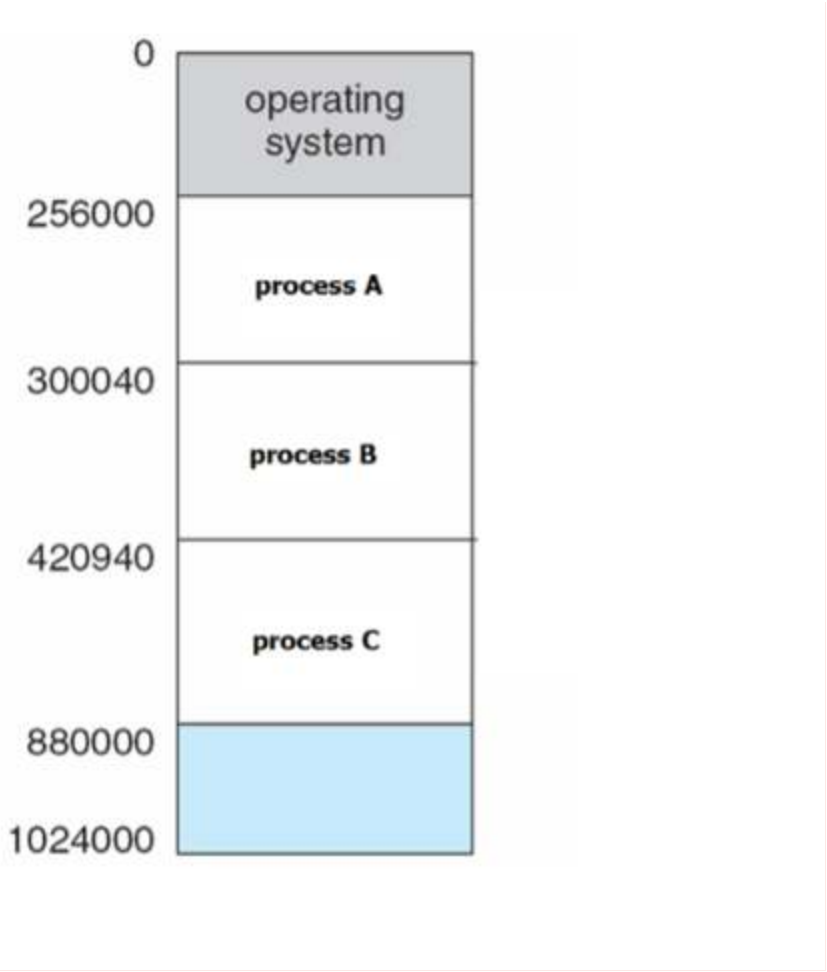
15. Lợi thế của phương pháp biên dịch tại thời điểm nạp (Load) là gì? *

A. Có thể nạp chương trình vào nhiều vị trí trong bộ nhớ mà không cần biên dịch lại.

16. Phân mảnh nội xảy ra khi nào? *

D. Hệ điều hành cấp phát bộ nhớ chính theo khối

17 . Cho mô hình cấp phát bộ nhớ liên tục như sau, trả lời các câu hỏi bên dưới bằng cáchđiền giá trị vào ô trống. *



Khi tiến trình B được xử lý trong CPU thì giá trị thanh ghi Base là 300040

Khi tiến trình C được xử lý trong CPU thì giá trị thanh ghi Limit là 4590

Khi tiến trình A được xử lý trong CPU, địa chỉ truy cập 300040 có hợp lệ không? (Điền Y hoặc N) => N

Có 1 lổ trống trong bộ nhớ

18. Kích thước 1 trang (page size) trong hệ điều hành Windows là bao nhiêu bytes? *

B. 4096.0 / 4.0

19. Hệ điều hành Windows sử dụng phương pháp cấp phát bộ nhớ nào?

D. Phân trang

20. Hệ điều hành ghi và cập nhật mỗi dòng trong bảng phân trang cho đối tượng nào sauđây? *

D. mỗi tiến trình

21. Với thanh ghi tái định vị (base) và thanh ghi giới hạn (limit), mỗi địa chỉ luận lý (logicaladdress) phải có giá trị như thế nào với giá trị thanh ghi limit?

D. @ < limit

22. Địa chỉ của bảng phân trang được lưu trữ ở đâu?

. con trỏ base của bảng phân trang

23. Cho bảng phân đoạn (Segment Table) như sau:

Segment	Base	Limit
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

A. Khi Segment 1 đang được xử lý trong CPU, địa chỉ luận lý 13 được dịch ra địa chỉ vật lý là 2313
B. Khi Segment 2 đang được xử lý trong CPU, đia chỉ vật lý lớn nhất hợp lệ là 189
D. Kích thước của Segment 3 là 580

24. Bộ nhớ ảo (Virtual Memory) là gì?

C. vùng nhớ chứa những phần của tiến trình chưa được nạp vào bộ nhớ chính.

25. Phân trang theo yêu cầu (Demand Paging) hoạt động ra sao?

A. Hệ thống có thể chỉ nạp những trang cần thiết vào khung trang

26. Bất thường Belady nói đến số lỗi trang tăng khi cấp thêm khung trang xảy ra với các giảithuật nào? *

Chỉ có FIFO.

27. Trong hệ thống phân trang theo yêu cầu, nếu thời gian truy cập bộ nhớ chính là 200 nano-giây, thời gian xử lý lỗi trang là 8 mili-giây và một lỗi trang sẽ xảy ra trong 1000 lần truy vấncủa CPU; thì thời gian truy cập hiệu quả là bao nhiêu? *

B. EAT = 8.2 micro-giây

28. Copy-on-write (Sao chép khi ghi) là nguyên tắc gì? *

D. Tiến trình cha và con chia sẻ các trang trong bộ nhớ cho đến khi có thao tác cập nhật.

29. Cho chuỗi tham khảo trang 7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1. Hệ thống dùng 3khung trang, với giải thuật FIFO (First in first out) thì số lỗi trang là bao nhiêu sau khi phụcvụ hết chuỗi tham khảo trên? *

A. 15.0

30. Cho chuỗi tham khảo trang 7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1. Hệ thống dùng 3khung trang, với giải thuật LRU (Least Recently Used) thì số lỗi trang là bao nhiêu sau khiphục vụ hết chuỗi tham khảo trên? *

B. 12.0

31. Cho chuỗi tham khảo trang 7,0,1,2,0,3,0,4,2,3,0,3,0,3,2,1,2,0,1,7,0,1. Hệ thống dùng 3khung trang, với giải thuật Optimal thì số lỗi trang là bao nhiêu sau khi phục vụ hết chuỗitham khảo trên?

B. 12.0

32. Nguyên tắc thay thế trang địa phương (Local Replacement) là gì?

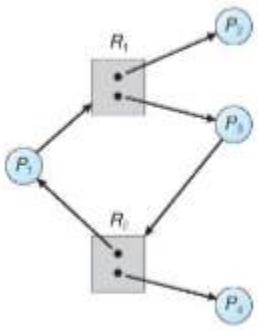
A. Khi thay thế trang, trang nạn nhân được tìm trong phạm vi các khung đã cấp cho tiến trình đó

33. Một hệ thống có 256 khung trang sẵn sàng và có 5 tiến trình vừa đến. Cho rằng kíchthước tiến trình P1 là 10 trang, tổng kích thước các tiến trình còn lại là 50 trang. Hệ thống sẽcấp cho P1 bao nhiêu khung trang nếu quy tắc "cấp phát tỉ lệ thuận" (Proportional Allocation)được áp dụng? *

C. 10.0

CHƯƠNG 8

1 .



Đồ thị phía trên được gọi là đồ thị gì?

Resource Allocation Graph

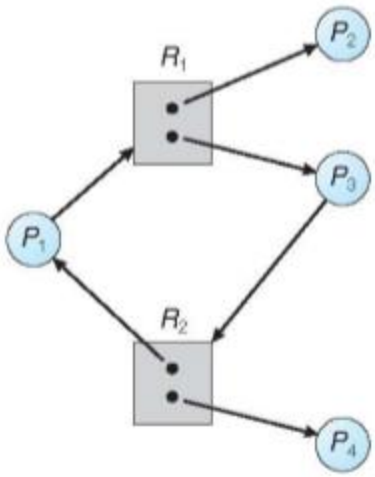
2 .Deadlock là trạng thái như thế nào?

Các tiến trình không thể tiếp tục được thực thi

3 .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)?

n+m

4 .



Đồ thị phía trên thể hiện thông tin gì?

A .Hệ thống có 4 tiến trình và 2 loại tài nguyên.

5 .Mục tiêu của giải thuật "Nhà băng" (Banker) là gì?

Chỉ ra một thứ tự thực thi của các tiến trình sao cho hệ thống luôn an toàn.

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

Starvation

7 .Quan hệ giữa "an toàn" và "deadlock" được diễn đạt như thế nào?

Hệ thống chỉ có thể bị deadlock khi nó có trạng thái không an toàn.

8 .Cho hệ thống có 5 tiến trình và 4 loại tài nguyên: A, B, C và D. Áp dụng giải thuật Banker(Nhà băng).

Process	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	1	2	1	5	2	0
P_1	1	0	0	0	1	7	5	0				
P_2	1	3	5	4	2	3	5	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

Hệ thống này có bao nhiêu thực thể tài nguyên mỗi loại?

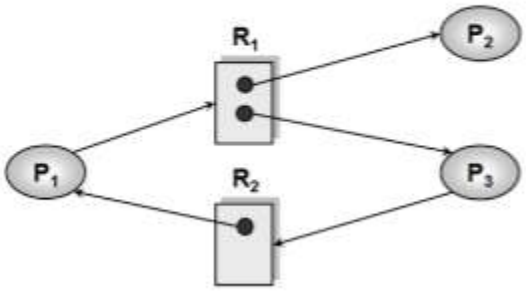
3 thực thể A, 14 thực thể B, 12 thực thể C và 12 thực thể D.

9 .Chọn phát biểu đúng cho điều kiện để tồn tại deadlock no preemtion :

Hệ thống không đòi lại được tài nguyên sau khi đã cấp phát.

10.Theo đồ thị RAG phía trên, có thể kết luận gì cho hệ thống?

10 .



.P1 chỉ có thể chạy tiếp khi P2 kết thúc

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

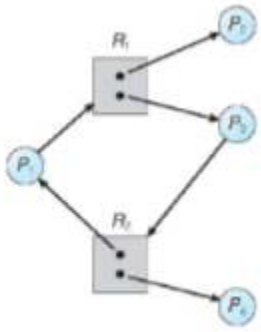
request – use – release.

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

Hệ thống thiếu thốn tài nguyên (Starvation)

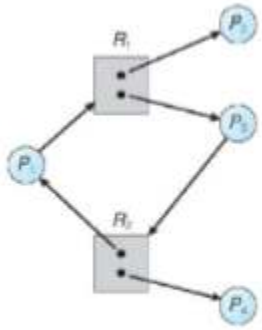
13 .Phát biểu nào sau đây SAI về đồ thị cấp phát tài nguyên

C .Đồ thị có chu trình thì hệ thống bị deadlock



Theo đồ thị RAG phía trên, có thể kết luận gì cho hệ thống?

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



Theo đồ thị RAG phía trên, phát biểu nào ĐÚNG?

P1 đang chiếm giữ một thực thể của tài nguyên R2

16 .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 :

C .RAG trên là RAG vòng.

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

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

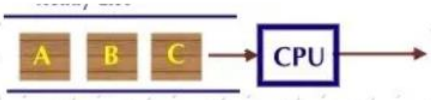
18 .Một hệ thống có n tiến trình và m loại tài nguyên và đang ở trạng thái Deadlock. Lúc đó,nếu chạy giải thuật “Phát hiện deadlock” (Detection Algorithm) thì độ phức tạp là

O(m * n*n)

CHƯƠNG 5

1. Giải thuật định thời First In, First Out (FIFO)

CPU được cấp phát cho tiến trình đầu tiên trong hàng đợi sẵn sàng có yêu cầu, là tiến trình được đưa vào hệ thống sớm nhất. Đây là giải thuật định thời theo **nguyên tắc độc quyền**.



Hình 2.12 Điều phối FIFO

Ví dụ :

Tiến trình	Thời điểm vào RL	Thời gian xử lý
P1	0	24
P2	1	3
P3	2	3

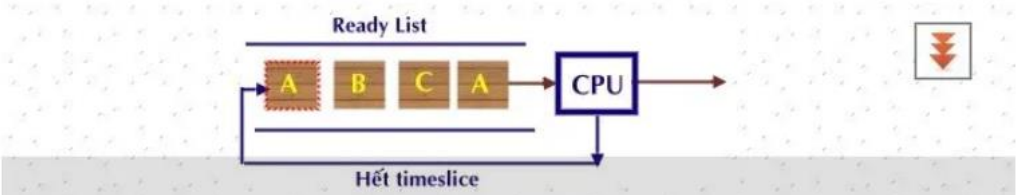
Thứ tự cấp phát CPU cho các tiến trình là :

P1	P2	P3
0	'24	27 30

- Thời gian xử lý: P1=24, P2=26, P3=28
- Thời gian xử lý trung bình: (24 + 26 +28)/3=26
- Thời gian đợi: P1=0, P2=23, P3 = 24
- Thời gian đợi trung bình: (0+23+24)/3 = 15.67

2. Giải thuật định thời Round Robin(RR)

Bộ định thời lần lượt cấp phát cho từng tiến trình trong danh sách một khoảng thời gian sử dụng CPU gọi là quantum. Đây là một giải thuật định thời **không độc quyền** : khi một tiến trình sử dụng CPU đến hết thời gian quantum dành cho nó, hệ điều hành thu hồi CPU và cấp cho tiến trình kế tiếp trong danh sách. Nếu tiến trình bị khóa hay kết thúc trước khi sử dụng hết thời gian quantum, hệ điều hành cũng lập tức cấp phát CPU cho tiến trình khác. Khi tiến trình tiêu thụ hết thời gian CPU dành cho nó mà chưa hoàn tất, tiến trình được đưa trở lại vào cuối danh sách sẵn sàng để đợi được cấp CPU trong lượt kế tiếp.



Hình 2.13 Điều phối Round Robin

Tiến trình	Thời điểm vào RL	Thời gian xử lý
P1	0	24
P2	1	3
P3	2	3

Nếu sử dụng quantum là 4 milisecondes, thứ tự cấp phát CPU sẽ là :

P1	P2	P3	P1	P1	P1	P1	P1
0	4	7	10	14	18	22	26 30

- Thời gian xử lý: P1=30, P2=6, P3=8
- Thời gian xử lý trung bình: (30+6+8)/3 = 14.67
- Thời gian đợi: P1=6, P2=3, P3=5
- Thời gian đợi trung bình: (6+3+5)/3=4.67

Thời gian hoàn thành: Thời gian mà quá trình hoàn thành việc thực thi.

Thời gian quay vòng: Thời gian Chênh lệch giữa thời gian hoàn thành và thời gian đến. Thời gian quay vòng = Thời gian hoàn thành - Thời gian đến

Thời gian chờ (WT): Thời gian Chênh lệch giữa thời gian quay vòng và thời gian liên tục.

Thời gian chờ = Thời gian quay vòng - Thời gian liên tục

Thời gian lần tiếp theo = 50% thời gian chạy trước + 50% thời gian dự đáon lần chạy trước

Convoy effect (Hiệu ứng hộ tống) : Mô tả nhiều tiến trình có burst ngắn chờ một tiến trình có burst dài, dẫn đến sự gia tăng thời gian chờ trung bình lên rất cao.

Cân bằng tải trên các hệ thống đa nhân sẽ cân bằng tải giữa các nhân CPU, mặc dù việc di chuyển các tiểu trình giữa các nhân để cân bằng tải có thể làm cho nội dung bộ nhớ cache bị vô hiệu và vì vậy có thể làm tăng thời gian truy cập bộ nhớ.

Lập lịch thời gian thực mềm (Soft real-time) ưu tiên cho các tác vụ thời gian thực hơn các tiến trình không cần thời gian thực. Lập lịch thời gian thực cứng (Hard real-time) cung cấp khoảng thời gian đảm bảo cho tác vụ thời gian thực.

Lập lịch “Thời gian thực tỷ lệ đơn điệu” (Rate-monotonic real-time) sử dụng một chính sách ưu tiên tĩnh và bộ lập lịch giữ quyền ưu tiên thu hồi CPU.

Lập lịch “Tác vụ tới hạn” (EDF / Earliest-deadline-first) chỉ định các ưu tiên theo thời hạn chót. tiến trình nào càng gần thời hạn chót, mức độ ưu tiên càng cao; tiến trình nào càng xa thời hạn chót, mức độ ưu tiên càng thấp.

Lập lịch theo tỷ lệ phân bổ (Proportional share) chia cho tất cả các ứng dụng T phần thời gian. Nếu một ứng dụng được phân bổ N phần thời gian, nó được đảm bảo có được tỉ lệ N／T của tổng thời gian xử lý.

Linux sử dụng bộ lập lịch “hoàn toàn công bằng” (CFS / completely fair scheduler), chỉ định tỷ lệ thời gian sử dụng cho từng tác vụ. Tỷ lệ này dựa trên giá trị thời gian chạy giả lập (vruntime) đã gán cho mỗi tác vụ.

Bộ lặp lịch Windows giữ quyền ưu tiên (preemptive), và có 32 cấp độ ưu tiên để xác lập thứ tự định thời tiểu trình.

Mô hình hóa và mô phỏng có thể được sử dụng để đánh giá một thuật toán lập lịch CPU.

Khóa loại trừ lẫn nhau (Mutex lock) cung cấp loại trừ lẫn nhau bằng cách yêu cầu tiến trình có được khóa trước khi vào đoạn mã nguy cơ và giải phóng khóa khi thoát khỏi đoạn mã nguy cơ vừa thực thi xong.

Đồ thị RAG và đồ thị tập hợp của đỉnh V và cạnh E.

Đỉnh của tiến trình và đỉnh của tài nguyên. Cạnh thì có 2 loại cạnh, cạnh của yêu cầu và cạnh cấp phát.

Có bốn điều kiện cần thiết để xảy ra tắc nghẽn: (1) loại trừ lẫn nhau (mutual exclusion), (2) giữ và chờ đợi (hold and wait), (3) hệ thống không thể chiếm quyền ưu tiên (non-

preemptive) và (4) chờ nhau theo vòng tròn (circular wait). Tắc nghẽn chỉ có thể xảy ra khi tất cả bốn điều kiện có mặt.

Tắc nghẽn có thể được ngăn chặn bằng cách loại trừ ít nhất một trong bốn điều kiện cần thiết để xảy ra tắc nghẽn. Trong bốn điều kiện cần thiết, loại bỏ sự chờ nhau theo vòng tròn là cách tiếp cận thực tế duy nhất.

Nếu tắc nghẽn xảy ra, một hệ thống có thể cố gắng phục hồi từ tắc nghẽn bằng cách hủy bỏ một trong các tiến trình đang nằm trong vòng tròn chờ đợi hoặc chiếm quyền lấy lại các tài nguyên đã được gán cho tiến trình tắc nghẽn.

Bộ quản lý bộ nhớ [Memory-Management Unit (MMU)] là thiết bị ánh xạ địa chỉ ảo sang địa chỉ vật lý.

Phương pháp cơ bản để thực hiện phân trang liên quan đến việc phân chia bộ nhớ vật lý thành các khối có kích thước cố định được gọi là khung (frames) và chia bộ nhớ luận lý thành các khối có cùng kích thước được gọi là trang (pages).

Bộ nhớ luận lý cũng được chia thành các khối cố định có cùng kích thước gọi là *trang nhớ* (page).

Frame và trang nhớ có kích thước bằng nhau.

Hệ điều hành phải thiết lập một *bảng phân trang* (page table) để ánh xạ địa chỉ luận lý thành địa chỉ thực

- Mỗi process được cấp phát một bảng phân trang
- Thiết lập bảng phân trang cho process là một phần của chuyển ngữ cảnh

Kỹ thuật phân trang khiến bộ nhớ bị phân mảnh nội, nhưng khắc phục được phân mảnh ngoại.

Địa chỉ do CPU tạo ra thường được gọi là địa chỉ luận lý (logical address), trong khi một địa chỉ mà đơn vị bộ nhớ cần biết — tức là giá trị địa chỉ được nạp vào thanh ghi địa chỉ bộ nhớ (memory-address register) của bộ nhớ - thường được gọi là địa chỉ vật lý (physical address).

Một cách để phân bổ một không gian địa chỉ cho mỗi tiến trình là thông qua việc sử dụng các thanh ghi cơ sở (base) và giới hạn (limit). Thanh ghi cơ sở giữ địa chỉ vật lý đầu tiên (nhỏ nhất) của vùng nhớ hợp pháp và thanh ghi giới hạn chỉ định kích thước của phạm vi bộ nhớ mà tiến trình đó được phép truy cập.

Thời kỳ biên dịch (compile time). Nếu bạn biết tại thời điểm biên dịch nơi tiến trình sẽ nằm trong bộ nhớ, thì mã tuyệt đối có thể được tạo ra. Ví dụ: nếu một tiến trình người

dùng sẽ bắt đầu từ vị trí R, thì mã trình biên dịch được tạo sẽ bắt đầu tại vị trí đó và được tăng dần từ đó. Nếu một thời gian sau, vị trí bắt đầu thay đổi, thì cần phải biên dịch lại toàn bộ mã này.

Thời kỳ nạp (load time). Nếu tại thời điểm biên dịch, nếu không biết tiến trình sẽ được đặt ở đâu trong bộ nhớ, thì trình biên dịch phải tạo ra mã có thể di dời. Trong trường hợp này, việc liên kết địa chỉ sau cùng bị trì hoãn cho đến thời điểm nạp. Nếu địa chỉ bắt đầu thay đổi, chúng ta chỉ cần nạp lại mã người dùng (vào một ví trí khác trong bộ nhớ) để điều chỉnh sự thay đổi này.

Thời kỳ thực thi (Execution time). Nếu tiến trình có thể được di chuyển trong quá trình thực thi của nó từ phân đoạn bộ nhớ này sang phân đoạn bộ nhớ khác, thì việc liên kết địa chỉ phải được trì hoãn cho đến thời điểm chạy. Một bộ phận phần cứng đặc biệt phải có sẵn để chương trình này hoạt động (sẽ được thảo luận trong Phần 9.1.3). Hầu hết các hệ điều hành sử dụng phương pháp này.

- Cấu trúc của một tiến trình gồm những phần code nào?

Đoạn mã đi vào (entry section)

Vùng nguy cơ

Đoạn mã kết thúc (exit section)

Phần còn lại

- Lệnh đơn nguyên là gì? Biến số đơn nguyên là gì?

Lệnh đơn nguyên là không bị tách rời, không bị ngắt.

Biến đơn nguyên là cung cấp những cập nhật đơn nguyên về dữ liệu cơ bản.

- Rào cản (Barrier) là gì? Cách sử dụng?

Rào cản bộ nhớ là một lệnh buộc bất kỳ thay đổi nào trong bộ nhớ phải được truyền (hiển thị) cho tất cả các bộ xử lý khác.

Khi một lệnh hàng rào bộ nhớ được thực hiện, hệ thống đảm bảo rằng tất cả các tải và lưu trữ được hoàn thành trước khi thực hiện bất kỳ hoạt động tải hoặc xé nào tiếp theo. Do đó, ngay cả khi các lệnh được sắp xếp lại, rào cản bộ nhớ đảm bảo rằng các hoạt động lưu trữ được hoàn thành trong bộ nhớ và hiển thị cho các bộ xử lý khác trước khi các hoạt động tải hoặc lưu trữ trong tương lai được thực hiện.

- Test_and_set() hoạt động ra sao?

Giải pháp Thử và Thiết lập Test-And-Set: Các khoá luận lý chia sẻ, khởi tạo là FALSE.

- Semaphore là gì? Cách sử dụng?

Semaphore: Công cụ đồng bộ mà không bị hạn chế bận chờ đợi.

+ Các thao tác đính kèm: wait() và signal() ← chỉ có các thao tác này mới có quyền truy cập semaphore S.

+ Semaphore có thể là kiểu số nguyên counting (dãy không giới hạn) hay kiểu nhị phân binary (0 hoặc 1).

- Monitor là gì? Cách nó hoạt động?

- Bộ quan sát (Monitor) là một trừu tượng mức cao, cung cấp cơ chế đồng bộ hiệu quả và tiện lợi.

+ Chỉ có duy nhất một tiến trình được hoạt động bên trong một bộ quan sát tại một thời điểm.

+ Chúng ta có thể tuỳ biến các biến số điều kiện (condition variables) để tạm dừng hay khởi chạy lại một tiến trình (ví dụ condition x, y);

- x.wait() – tiến trình đã thực hiện gọi sẽ bị tạm dừng cho đến khi có thực thi lệnh x.signal()
- x.signal() – khởi chạy tiếp tục một trong những tiến trình đang bị tạm dừng.

+ Có thể hiện thực thông qua semaphore.

- Starvation nói đến tình trạng nào trong hệ thống?

Cạn kiệt tài nguyên

- Tình trạng cạnh tranh (race condition) là gì?

Tình trạng cạnh tranh (race condition) xảy ra khi các tiến trình có quyền truy cập đồng thời vào dữ liệu chia sẻ và kết quả có thể khác nhau, phụ thuộc vào thứ tự thực thi cụ thể trong quá trình xảy ra truy cập đồng thời. Điều kiện cạnh tranh có thể dẫn đến các giá trị bị sai sót (không đồng nhất) của dữ liệu chia sẻ.

- 03 tiêu chí của một giải thuật đồng bộ là gì?

Bất kỳ một giải pháp cho bài toán đoạn mã nguy cơ phải đáp ứng ba yêu cầu sau: (1) loại trừ lẫn nhau, (2) đảm bảo sự tiến triển và (3) chờ đợi có hạn định. Loại trừ lẫn nhau đảm bảo rằng chỉ có một tiến trình tại một thời điểm được thực thi đoạn mã nguy cơ của nó. Đảm bảo sự tiến triển cần đảm bảo rằng các tiến trình sẽ hợp tác xác định tiến trình nào tiếp theo sẽ thực thi đoạn mã nguy cơ của nó. Chờ đợi có hạn định sẽ đảm bảo thời gian một tiến trình chờ trước khi nó có thể thực thi đoạn mã nguy cơ của nó là có hạn định.

CHƯƠNG 10

- Lỗi trang, các bước xử lý của hệ thống.

Lỗi trang [Page fault] là kết quả của lần đầu tiên một trang cụ thể được truy cập, gây ra một bẫy [trap] trong bộ nhớ.

CHƯƠNG 11: LƯU TRỮ THỨ CẤP

- Đặc trưng của đĩa cứng HDD và các thông số hoạt động của nó.

- Đặc trưng của SSD.

- Các thiết bị NVM.
- Định thời đĩa: FCFS, SCAN, C-SCAN.
- Các cấp độ RAID.

- Kỹ thuật lưu trữ dư thừa từ mảng các đĩa độc lập [RAID]: nhiều ổ đĩa sẽ làm tăng độ tin cậy bằng cách lưu trữ dư thừa – từ đó làm tăng thời gian hoạt động trước khi xảy ra lỗi MTTF [mean time to failure]
    - Chia đĩa [striping] / RAID 0: sử dụng nhóm các đĩa thành một đơn vị lưu trữ duy nhất.
    - Nhân đôi / đổ bóng [Mirroring/shadowing] / RAID 1: ghi dữ liệu giống nhau lên 2 đĩa cứng.
    - Nhân đôi những đĩa đã chia [Striped mirrors] /RAID 1+0 hoặc Chia ra phần đã nhân đôi [mirrored striped] (RAID 0+1) cung cấp cả độ tin cậy lẫn tốc độ truy xuất nhanh.
    - Khối kiểm tra xen kẽ [Block interleaved parity] /RAID 4, 5, 6: sử dụng dư thừa ít hơn.
- Solaris ZFS bổ sung tổng kiểm tra [checksums] tất cả dữ liệu và siêu dữ liệu - phát hiện xem đối tượng có đúng cái đang cần không và liệu nó có thay đổi không