



## Nguyên lý hệ điều hành (2324I\_INT2214\_20) Part1- Multip

Bắt đầu vào lúc	Saturday, 7 October 2023, 8:00 PM
Trạng thái	Đã xong
Kết thúc lúc	Saturday, 7 October 2023, 8:36 PM
Thời gian thực hiện	35 phút 48 giây
Điểm	38,00/60,00
Điểm	6,33 trên 10,00 (63,33%)

### Câu Hỏi 1 Hoàn thành

Mức độ đa chương trình là:

- ☐ A. Số lượng các tiến trình trong hàng đợi vào ra
- ☐ B. Số lượng các tiến trình trong bộ nhớ
- ☐ C. Số lượng các tiến trình thực thi trong một đơn vị thời gian
- ☒ D. Số lượng các tiến trình trong hàng đợi sẵn sàng



**Câu Hỏi 2**

Hoàn thành

Hai tiến trình, P1 và P2, cần truy cập vào khu vực quan trọng. Xem xét đoạn mã đồng bộ hóa sau được sử dụng

P1:

```
while(true)
{
w1 = true;
while(w2 == true);
Critical section
w1 = false;
}
Remainder Section
```

P2 :

```
while(true)
{
w2 = true;
while(w1 == true);
Critical section
w2 = false;
}
Remainder Section
```

Ở đây, w1 và w2 là các biến chia sẻ, được khởi tạo bằng false. Điều nào dưới đây là đúng về đoạn mã trên ?

- ☒ A. Bế tắc có thể xảy ra, nhưng đảm bảo tính loại trừ lẫn nhau
- ☐ B. Các tiến trình lần lượt vào khu vực quan trọng
- ☐ C. Nó không đảm bảo tính loại trừ lẫn nhau
- ☐ D. Nó không đảm bảo tính chờ đợi có giới hạn

**Câu Hỏi 3**

Hoàn thành

Hệ điều hành Linux có

- ☐ A. Vi nhân (microkernel)
- ☒ B. Nhân (kernel) monolithic có modules
- ☐ C. Nhân (kernel) monolithic
- ☐ D. Các phương án trên đều sai

**Câu Hỏi 4**

Hoàn thành

Kỹ thuật nào sau đây giúp giảm thiểu tình trạng nhàn rỗi của CPU hoặc các thiết bị vào/ra khi hệ điều hành đơn vi xử lý ?

- ☐ A. Hệ điều hành phân tán
- ☐ B. Sử dụng hành đợi
- ☒ C. Đa chương trình
- ☐ D. sử dụng pipe

**Câu Hỏi 5**

Hoàn thành

Truyền thông giữa các tiến trình\_\_\_\_\_

- ☐ A. không bao giờ cần
- ☐ B. được thực hiện thông qua ổ đĩa
- ☒ C. cho phép tiến trình đồng bộ hoá hoạt động
- ☐ D. là cần thiết đối với tất cả các tiến trình

**Câu Hỏi 6**

Hoàn thành

Điều kiện loại trừ lẫn nhau (trong bế tắc) xảy ra khi:

- ☒ A. Có ít nhất một tài nguyên chia sẻ
- ☐ B. Tất cả các phương án trên
- ☐ C. Có ít nhất một tài nguyên bị chiếm giữ trong chế độ không chia sẻ
- ☐ D. Bộ vi xử lý phải là đơn vi xử lý

**Câu Hỏi 7**

Hoàn thành

Thuật toán lập lịch Round Robin là \_\_\_\_\_

- ☐ A. thuật toán lập lịch phân tiến trình thành các nhóm khác nhau
- ☐ B. thuật toán lập lịch lúc cho phép dừng, lúc không cho phép dừng
- ☒ C. thuật toán lập lịch cho phép dừng
- ☐ D. thuật toán lập lịch không cho phép dừng

**Câu Hỏi 8**

Không trả lời

A system uses Round Robin scheduling algorithm with time quantum = 3. The process 's information (arrival time, burst time) is given below:  
(2,4,3), P4 (6,4,4). The waiting time of the process P1, P2, P3, P4 are respectively

(Each number is rounded by 2 decimal digits, separated by a comma following by a single blank character e.g. 1.23, 4.56, 7.89)

(Supposing that P<sub>c</sub> is currently on CPU, P<sub>c</sub> is going to leave CPU.

- If P<sub>i</sub> and P<sub>j</sub> arrives the ready queue at the same time,  $i \neq c, j \neq c, i < j$ , then P<sub>i</sub> will enter the ready queue before P<sub>j</sub>
- If P<sub>c</sub> and P<sub>i</sub> arrives the ready queue at the same time,  $i \neq c$ , then P<sub>i</sub> will enter the ready queue before P<sub>c</sub> )

**Câu Hỏi 9**

Hoàn thành

Ở trạng thái chờ, tiến trình \_\_\_\_\_

- ☐ A. đang chờ bộ vi xử lý trong hàng đợi sẵn sàng
- ☐ B. đã kết thúc
- ☐ C. đang chạy
- ☒ D. đang chờ cho kết thúc của việc vào/ra dữ liệu

**Câu Hỏi 10**

Hoàn thành

Đâu không phải là một trạng thái của tiến trình:

- ☐ A. chờ đợi
- ☒ B. ưu tiên
- ☐ C. sẵn sàng
- ☐ D. đang chạy

**Câu Hỏi 11**

Hoàn thành

Thuật toán lập lịch CPU nào thường cho thời gian phản hồi trung bình nhỏ nhất:

- ☒ A. Round Robin với time quantum nhỏ
- ☐ B. SJF cho phép đứng
- ☐ C. Thứ tự ưu tiên cho phép đứng
- ☐ D. FCFS

**Câu Hỏi 12**

Hoàn thành

Đoạn mã, trong đó tiến trình có thể thay đổi các biến toàn cục, cập nhật vào các tệp tin được gọi là:

- ☐ A. khu vực không quan trọng
- ☒ B. khu vực quan trọng
- ☐ C. một chương trình
- ☐ D. khu vực đồng bộ hóa

**Câu Hỏi 13**

Hoàn thành

Trong môi trường đa nhiệm, khi một tiến trình luôn bị từ chối cho phép dùng tài nguyên nó được gọi là:

- ☐ A. lão hóa
- ☒ B. nạn đói
- ☐ C. bế tắc
- ☐ D. đảo ngược

**Câu Hỏi 14**

Hoàn thành

Trong chế độ không chờ gửi:

- ☒ A. tiến trình gửi sẽ gửi thông điệp và rồi quay lại thực thi
- ☐ B. tiến trình gửi liên tục gửi cho đến khi nó nhận được một thông điệp
- ☐ C. tiến trình gửi liên tục gửi cho đến khi thông điệp được nhận
- ☐ D. Cả ba phương án trên đều sai

**Câu Hỏi 15**

Hoàn thành

Hai tiến trình sau chia sẻ một biến chung X, và một biến semaphore nhị phân T:

Tiến trình A

```
{Int Y;  
Wait (T)  
A1: Y = X * 2;  
A2: X = Y;  
}
```

Tiến trình B

```
{Int Z;  
B1: Z = X + 1;  
B2: X = Z;  
Signal(T)  
}
```

X được khởi tạo bằng 5, T được khởi tạo bằng 0 trước khi các tiến trình bắt đầu thực thi. Thông thường, các câu trả lời đúng có thể khác nhau. Giả sử giá trị cuối cùng của X sau khi thực thi xong hai tiến trình A và B sẽ là:

- ☐ A. 10
- ☒ B. 12
- ☐ C. 6
- ☐ D. 11

**Câu Hỏi 16**

Hoàn thành

A system uses non-preemptive Priority scheduling algorithm (the process with the higher priority will be executed first). The process information (arrival time, duration, priority) is P1 (0,2,1), P2 (4,10,2), P3 (5,4,3), P4 (7,4,4). The waiting time of process P2 is

0, 4, 5, 7

(Each number is rounded by 2 decimal digits, separated by a comma following by a single blank character, e.g. 1.23, 4.56, 7.89)

**Câu Hỏi 17**

Hoàn thành

Trong thuật toán lập lịch đa cấp phân hồi,

- ☐ A. tiến trình không thể chuyển đổi giữa các cấp hàng đợi
- ☐ B. tiến trình liên tục gửi thông điệp cho hệ thống
- ☐ C. tiến trình không bị phân thành các nhóm khác nhau
- ☒ D. tiến trình có thể di chuyển đến một hàng đợi sẵn sàng được phân cấp khác

**Câu Hỏi 18**

Hoàn thành

Tiến trình không thể thực thi việc chuyển đổi trạng thái nào sau đây:

- ☐ A. từ sẵn sàng sang chạy
- ☒ B. từ bị chặn sang chạy
- ☐ C. từ chạy sang bị chặn
- ☐ D. từ bị chặn sang sẵn sàng

**Câu Hỏi 19**

Hoàn thành

Một cách để đảm bảo điều kiện chờ đợi vòng tròn không xảy ra:

- ☐ A. Không bao giờ để một tiến trình có được nguồn tài nguyên đang nắm giữ bởi tiến trình khác
- ☒ B. Tạo một trình tự sử dụng các loại tài nguyên
- ☐ C. Tất cả các phương án trên
- ☐ D. Cho phép tiến trình chỉ chờ đợi một nguồn tài nguyên tại một thời điểm

**Câu Hỏi 20**

Hoàn thành

A system uses non-preemptive SJF scheduling algorithm. The process 's information (arrival time, duration, priority) is given below. The waiting time of the process P1, P2, P3, P4 are respectively

(Each number is rounded by 2 decimal digits, separated by a comma following by a single blank character, e.g. 0.12, 1.34, 5.67)

**Câu Hỏi 21**

Hoàn thành

Khi nào hệ điều hành nên kiểm tra bế tắc ?

- ☐ A. Tất cả các phương án trên đều sai
- ☐ B. mỗi khi có yêu cầu sử dụng tài nguyên của tiến trình
- ☐ C. theo khoảng thời gian cố định
- ☒ D. mỗi khi có yêu cầu sử dụng tài nguyên của tiến trình và theo khoảng thời gian cố định

**Câu Hỏi 22**

Hoàn thành

Trong truyền thông gián tiếp giữa tiến trình P và Q:

- ☐ A. Cả ba phương án trên đều sai
- ☐ B. có một tiến trình xử lý và truyền các thông điệp giữa P và Q
- ☒ C. có một hộp thư để giúp cho việc truyền thông giữa P và Q
- ☐ D. có một máy giữa hai tiến trình để giúp cho việc truyền thông

**Câu Hỏi 23**

Hoàn thành

Trong chế độ chờ nhận:

- ☐ A. tiến trình nhận không hoạt động
- ☐ B. tiến trình nhận liên tục gửi và nhận thông điệp từ hệ thống
- ☒ C. tiến trình nhận phải chờ cho đến khi nhận được thông điệp
- ☐ D. tiến trình nhận sẽ nhận thông điệp và rồi quay lại thực thi

**Câu Hỏi 24**

Hoàn thành

Nội dung nào sau đây không được lưu khi chuyển trạng giữa các tiến trình:

- ☐ A. TLB
- ☐ B. Tất cả các phương án trên
- ☐ C. Các thanh ghi thông thường
- ☒ D. Bộ đếm chương trình

**Câu Hỏi 25**

Hoàn thành

Vấn đề khó khăn của bộ lập lịch ngắn hạn khi sử dụng SJF là:

- ☐ A. Thuật toán sử dụng quá nhiều tài nguyên của hệ thống
- ☐ B. Thuật toán rất khó hiểu
- ☒ C. Không biết được độ dài của phiên sử dụng CPU tiếp theo của tiến trình
- ☐ D. Thuật toán không tối ưu về thời gian chờ

**Câu Hỏi 26**

Hoàn thành

Round Robin là phiên bản cho phép dừng của thuật toán nào sau đây ?

- ☐ A. SJF
- ☐ B. Priority (Thứ tự ưu tiên)
- ☒ C. FCFS
- ☐ D. LJF (thời gian ngắn nhất trước)

**Câu Hỏi 27**

Hoàn thành

Tinh huống mà một số tiến trình truy cập và thao tác đồng thời trên cùng một vùng dữ liệu và kết quả của việc thực hiện phụ thuộc vào thứ tự các tiến trình truy cập vào vùng dữ liệu được gọi là:

- ☐ A. nạn đói
- ☐ B. tính nhất quán dữ liệu
- ☐ C. quá trình lão hóa
- ☒ D. điều kiện tương tranh

**Câu Hỏi 28**

Hoàn thành

Ngăn xếp (stack) của một tiến trình không chứa:

- ☐ A. PID của tiến trình con
- ☐ B. Địa chỉ trả về
- ☐ C. Biến địa phương
- ☒ D. Tham số của các hàm

**Câu Hỏi 29**

Hoàn thành

Thuật ngữ phân chia thời gian tương đương với \_\_\_\_\_

- ☐ A. Đa người dùng
- ☐ B. Đa chương trình
- ☒ C. Đa nhiệm
- ☐ D. Đa vi xử lý

**Câu Hỏi 30**

Hoàn thành

Giả sử rằng một tiến trình ở trạng thái "chờ" để đợi một số dịch vụ vào/ra dữ liệu. Khi dịch vụ này hoàn thành, nó chuyển sang trạng thái:

- ☐ A. trạng thái kết thúc
- ☐ B. trạng thái chạy
- ☒ C. trạng thái sẵn sàng
- ☐ D. trạng thái treo

**Câu Hỏi 31**

Hoàn thành

A system uses Round Robin scheduling algorithm with time quantum = 2. The process 's information (arrival time, duration, priority) is P1 (4,6,3), P2 (5,4,4), P3 (5,3,3), P4 (6,4,4). The turn around time of the process P1, P2, P3, P4 are respectively

(Each number is rounded by 2 decimal digits, separated by a comma following by a single blank character e.g. 14, 15.33, 20.10, 18)

(Supposing that P<sub>c</sub> is currently on CPU, P<sub>c</sub> is going to leave CPU.

- If P<sub>i</sub> and P<sub>j</sub> arrives the ready queue at the same time,  $i \neq c, j \neq c, i < j$ , then P<sub>i</sub> will enter the ready queue before P<sub>j</sub>
- If P<sub>c</sub> and P<sub>i</sub> arrives the ready queue at the same time,  $i \neq c$ , then P<sub>i</sub> will enter the ready queue before P<sub>c</sub> )

**Câu Hỏi 32**

Hoàn thành

A system uses FCFS scheduling algorithm. The process 's information (arrival time, duration, priority) is P1 (0,2,1), P2 (4,6,2), P3 (5,3,3), P4 (6,4,4). The waiting time of the process P1, P2, P3, P4 are respectively

(Each number is rounded by 2 decimal digits, separated by a comma following by a single blank character, e.g. 14, 15.33, 20.10, 18)



**Câu Hỏi 33**

Hoàn thành

Thời gian chuyển trạng thái là \_\_\_\_\_.

- ☐ A. khoảng thời gian chương trình chuyển từ trạng thái này sang trạng thái khác
- ☐ B. khoảng thời gian tiến trình chuyển từ trạng thái này sang trạng thái khác
- ☐ C. khoảng thời gian chuyển trạng giữa bộ nhớ ngoài và bộ nhớ chính
- ☒ D. khoảng thời gian CPU chuyển giữa các tiến trình

**Câu Hỏi 34**

Hoàn thành

Có 5 tiến trình P0, P1, P2, P3, P4 và 4 kiểu tài nguyên A, B, C, D. Tại thời điểm t0, trạng thái của hệ thống như sau:

	Đã cấp phát	Cần tối đa	Còn
	A B C D	A B C D	A B C D
P0	2 0 0 1	4 2 1 2	3 3 2 1
P1	3 1 2 1	5 2 5 2	
P2	2 1 0 3	2 3 1 6	
P3	1 3 1 2	1 4 2 4	
P4	1 4 3 2	3 6 6 5	

Yêu cầu (0,2,1,0) của P4 sẽ dẫn hệ thống đến trạng thái nào:

- ☐ A. Được bảo vệ
- ☐ B. Bế tắc
- ☒ C. Không an toàn
- ☐ D. An toàn

**Câu Hỏi 35**

Hoàn thành

Hệ thống có 6 đĩa từ, và N tiến trình chia sẻ tài nguyên này. Mỗi tiến trình cần 3 đĩa từ. Giá trị lớn nhất của N để

- ☒ A. 2
- ☐ B. 1
- ☐ C. 3
- ☐ D. 4

**Câu Hỏi 36**

Hoàn thành

Khi tiến trình có yêu cầu vào/ra dữ liệu thì \_\_\_\_\_.

- ☐ A. Nó được chuyển vào hàng đợi sẵn sàng
- ☐ B. Nó được chuyển vào hàng đợi nhiệm vụ
- ☐ C. Nó được chuyển vào hàng đợi vào/ra
- ☒ D. Nó được chuyển vào hàng đợi chờ (waiting queue)

### Câu Hỏi 37

Hoàn thành

Có 5 tiến trình P0, P1, P2, P3, P4 và 4 kiểu tài nguyên A, B, C, D. Tại thời điểm t0, trạng thái của hệ thống như sau:

	Đã cấp phát	Cần tối đa	Còn
	A B C D	A B C D	A B C D
P0	2 0 0 1	4 2 1 2	3 3 2 1
P1	3 1 2 1	5 2 5 2	
P2	2 1 0 3	2 3 1 6	
P3	1 3 1 2	1 4 2 4	
P4	1 4 3 2	3 6 6 5	

Đây là một trình tự an toàn của hệ thống:

- ☐ A. <P0, P1, P2, P3, P4>
- ☐ B. <P0, P2, P4, P1, P3>
- ☐ C. <P0, P4, P1, P2, P3>
- ☒ D. <P0, P3, P1, P2, P4>

### Câu Hỏi 38

Hoàn thành

Một cách để đảm bảo điều kiện giữ và chờ (trong bế tắc) không xảy ra:

- ☒ A. Tất cả các phương án trên
- ☐ B. Tiến trình muốn yêu cầu thêm tài nguyên phải giải phóng tất cả các tài nguyên đang chiếm giữ
- ☐ C. Tiến trình chỉ được yêu cầu tài nguyên khi nó không chiếm giữ tài nguyên nào
- ☐ D. Tất cả các tiến trình yêu cầu tài nguyên và được cấp phát tất cả tài nguyên cần thiết trước bắt đầu thực thi

### Câu Hỏi 39

Hoàn thành

A system uses Round Robin scheduling algorithm with time quantum = 2. The process's information (arrival time, burst time) is as follows: P1 (2,4,3), P2 (6,4,4). The waiting time of the process P1, P2, P3, P4 are respectively 8, 11, 3, 13

(Each number is rounded by 2 decimal digits, separated by a comma following by a single blank character)

(Supposing that P1 is currently on CPU, P1 is going to leave CPU.)

- If P1 and P2 arrives the ready queue at the same time,  $i \neq c, j \neq c, i < j$ , then P1 will enter the ready queue before P2
- If P1 and P2 arrives the ready queue at the same time,  $i \neq c, j \neq c$ , then P1 will enter the ready queue before P2

### Câu Hỏi 40

Hoàn thành

Khi một tiến trình P thực hiện phần mã ở khu vực quan trọng (critical section), không có tiến trình Q nào khác có thể thực hiện trong khu vực quan trọng (của Q). Tình huống này được gọi là

- ☒ A. Điều kiện tương tranh
- ☐ B. Điều kiện cần thiết
- ☐ C. Điều kiện đồng bộ
- ☐ D. Điều kiện ngăn chặn

