CHƯƠNG 3: LẬP LỊCH CPU

Những kiến thức trọng tâm của chương gồm có:

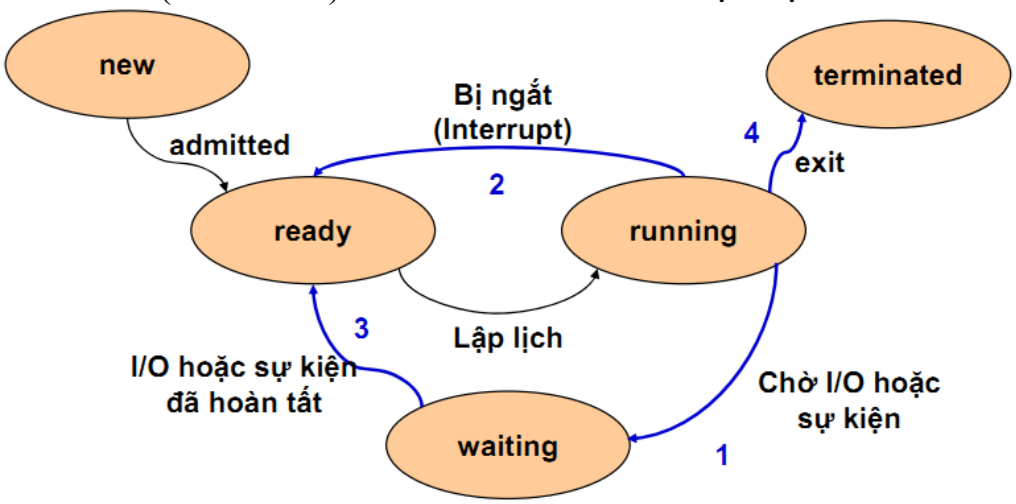
1. Các khái niệm cơ bản
2. Các thuật toán lập lịch
3. Ngắt

1. Khái niệm cơ bản

Trạng thái tiến trình

Khi một tiến trình thực thi, nó thay đổi trạng thái. Trạng thái của tiến trình được định nghĩa bởi các hoạt động hiện hành của tiến trình đó. Mỗi tiến trình có thể ở một trong những trạng thái sau:

- Mới (new): tiến trình đang được tạo ra
- Đang chạy (running): các chỉ thị đang được thực thi
- Chờ (waiting): tiến trình đang chờ sự kiện xảy ra (như hoàn thành việc nhập/xuất hay nhận tín hiệu)
- Sẵn sàng (ready): tiến trình đang chờ được gán tới một bộ xử lý.
- Kết thúc (terminated): tiến trình hoàn thành việc thực thi



Các tên trạng thái này là bất kỳ, và chúng khác nhau ở các hệ điều hành khác nhau. Tuy nhiên, các trạng thái mà chúng hiện diện được tìm thấy trên tất cả hệ thống. Các hệ điều hành xác định mô tả trạng thái tiến trình . Chỉ một tiến trình có thể đang chạy tức thì trên bất kỳ bộ xử lý nào mặc dù nhiều tiến trình có thể ở trạng thái sẵn sàng và chờ.

Lập lịch tiến trình

Mục tiêu của việc đa chương là có vài tiến trình chạy tại tất cả thời điểm để tận dụng tối đa việc sử dụng CPU. Mục tiêu của chia thời là chuyển CPU giữa các tiến trình thường xuyên để người dùng có thể giao tiếp với mỗi chương trình trong khi đang chạy. Một hệ thống đơn xử lý chỉ có thể chạy một tiến trình. Nếu nhiều hơn một tiến trình tồn tại, các tiến trình còn lại phải chờ cho tới khi CPU rảnh và có thể lập thời biểu lại.

1.1. Hàng đợi lập lịch

Khi các tiến trình được đưa vào hệ thống, chúng được đặt vào hàng đợi công việc. Hàng đợi chứa tất cả tiến trình trong hệ thống. Các tiến trình đang nằm trong bộ nhớ chính sẵn sàng và chờ để thực thi được giữ trên một danh sách được gọi là hàng đợi sẵn sàng. Hàng đợi này thường được lưu như một danh sách liên kết. Đầu của hàng đợi sẵn sàng chứa hai con trỏ: một chỉ đến PCB đầu tiên và một chỉ tới PCB cuối cùng trong danh sách. Chúng ta bổ sung thêm trong mỗi PCB một trường con trỏ chỉ tới PCB kế tiếp trong hàng đợi sẵn sàng.

Hệ điều hành cũng có các hàng đợi khác. Khi một tiến trình được cấp phát CPU, nó thực thi một khoảng thời gian và cuối cùng kết thúc, được ngắt, hay chờ một sự kiện xác định xảy ra, chẳng hạn như hoàn thành một yêu cầu nhập/xuất. Trong trường hợp yêu cầu nhập/xuất, một yêu cầu có thể là ổ đĩa băng từ tận hiến hay một thiết bị được chia sẻ như đĩa. Vì hệ thống có nhiều tiến trình, đĩa có thể bận với yêu cầu nhập/xuất của các tiến trình khác. Do đó, tiến trình phải chờ đĩa. Danh sách tiến trình chờ một thiết bị nhập/xuất cụ thể được gọi là hàng đợi thiết bị. Mỗi thiết bị có hàng đợi của chính nó như hình 0-4.

Một biểu diễn chung của lập thời biểu tiến trình là một lưu đồ hàng đợi. Mỗi hình chữ nhật hiện diện một hàng đợi. Hai loại hàng đợi được hiện diện: hàng đợi sẵn sàng và tập các hàng đợi thiết bị, vòng tròn hiện diện tài nguyên phục vụ hàng đợi, các mũi tên hiển thị dòng các tiến trình trong hệ thống.

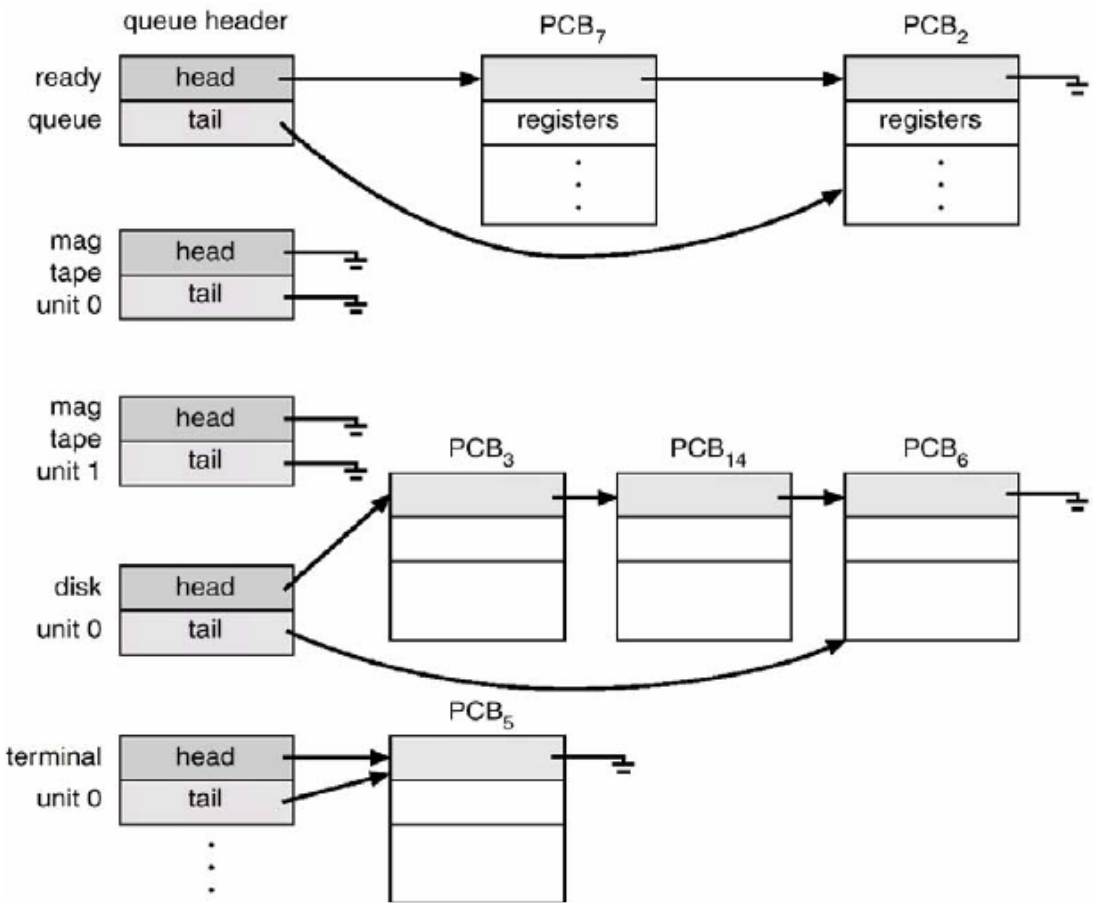
Một tiến trình mới khởi đầu được đặt vào hàng đợi. Nó chờ trong hàng đợi sẵn sàng cho tới khi nó được chọn thực thi. Một khi tiến trình được gán tới CPU và đang thực thi, một trong nhiều sự kiện có thể xảy ra:

• Tiến trình có thể phát ra một yêu cầu nhập/xuất và sau đó được đặt vào trong hàng đợi nhập/xuất.

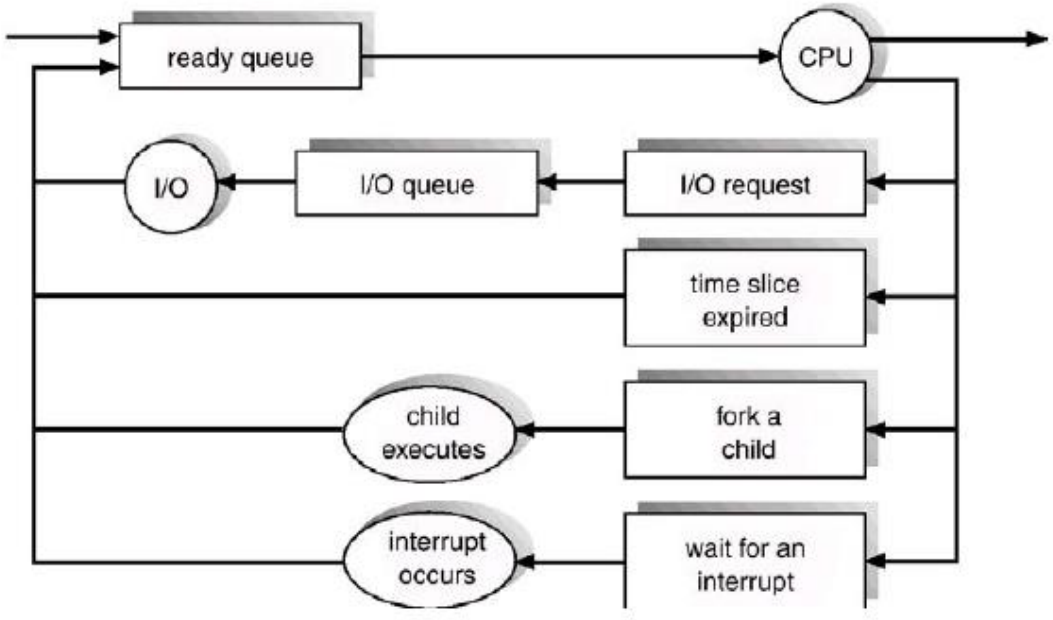
• Tiến trình có thể tạo một tiến trình con và chờ cho tiến trình con kết thúc

• Khi một ngắt xảy ra, tiến trình có thể bị buộc trả lại CPU và được đặt trở lại trong hàng đợi sẵn sàng.

Trong hai trường hợp đầu, cuối cùng tiến trình chuyển từ trạng thái chờ tới trạng thái sẵn sàng và sau đó đặt trở lại vào hàng đợi sẵn sàng. Một tiến trình tiếp tục chu kỳ này cho tới khi nó kết thúc. Tại thời điểm kết thúc nó bị xoá từ tất cả hàng đợi. Sau đó, PCB và tài nguyên của nó được thu hồi.



Hình - Hàng đợi sẵn sàng và các hàng đợi nhập/xuất khác nhau



Hình - Biểu diễn lưu đồ hàng đợi của lập thời biểu tiến trình

1.2. Bộ định thời biểu

Một tiến trình di dời giữa hai hàng đợi định thời khác nhau suốt thời gian sống của nó. Hệ điều hành phải chọn, cho mục đích định thời, các tiến trình từ

các hàng đợi này. Tiến trình chọn lựa này được thực hiện bởi bộ định thời hợp lý.

Trong hệ thống bó, thường nhiều hơn một tiến trình được gởi đến hơn là có thể được thực thi tức thì. Các tiến trình này được lưu tới thiết bị lưu trữ (như đĩa), nơi chúng được giữ cho việc thực thi sau đó. **Bộ định thời dài** (long-term scheduler) hay bộ định thời công việc (job scheduler), chọn các tiến trình từ vùng đệm và nạp chúng vào bộ nhớ để thực thi. **Bộ định thời ngắn** (short-term scheduler) hay bộ định thời CPU chọn một tiến trình từ các tiến trình sẳn sàng thực thi và cấp phát CPU cho tiến trình đó.

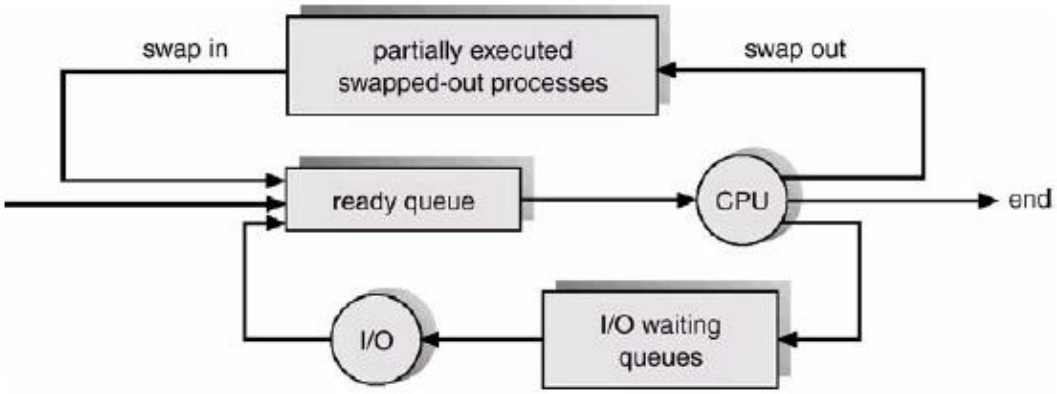
Sự khác biệt chủ yếu giữa hai bộ định thời là tính thường xuyên của việc thực thi. Bộ định thời CPU phải chọn một tiến trình mới cho CPU thường xuyên. Một tiến trình có thể thực thi chỉ một vài mili giây trước khi chờ yêu cầu nhập/xuất. Bộ định thời CPU thường thực thi ít nhất một lần mỗi 100 mili giây. Vì thời gian ngắn giữa việc thực thi nên bộ định thời phải nhanh. Nếu nó mất 10 mili giây để quyết định thực thi một tiến trình 100 mili giây thì 10/(100+10) = 9 phần trăm của CPU đang được dùng (hay bị lãng phí) đơn giản cho định thời công việc.

Ngược lại, bộ định thời công việc thực thi ít thường xuyên hơn. Có vài phút giữa việc tạo các tiến trình mới trong hệ thống. Bộ định thời công việc điều khiển mức độ đa chương – số tiến trình trong bộ nhớ. Nếu mức độ đa chương ổn định thì tốc độ trung bình của việc tạo tiến trình phải bằng tốc độ khởi hành trung bình của tiến trình rời hệ thống. Vì khoảng thời gian dài hơn giữa việc thực thi nên bộ định thời công việc có thể cấp nhiều thời gian hơn để chọn một tiến trình thực thi.

Bộ định thời công việc phải thực hiện một chọn lựa cẩn thận. Thông thường, hầu hết các tiến trình có thể được mô tả như là **tiến trình hướng nhập/xuất** (I/O-bound proces) hay **tiến trình hướng CPU** (CPU-bound process). Một tiến trình hướng nhập/xuất mất nhiều thời gian hơn để thực hiện nhập/xuất hơn thời gian tính toán. Ngược lại, một tiến trình hướng CPU phát sinh các yêu cầu nhập/xuất không thường xuyên, dùng thời gian để thực hiện việc tính toán hơn một tiến trình hướng nhập/xuất dùng. Bộ định thời công việc nên chọn sự kết hợp hài hoà giữa tiến trình hướng nhập/xuất và tiến trình hướng CPU. Nếu tất cả tiến trình là hướng nhập/xuất thì hàng đợi sẳn sàng sẽ luôn rỗng và bộ định thời CPU sẽ có ít công việc để thực hiện. Nếu tất cả tiến trình là hướng CPU thì hàng đợi nhập/xuất sẽ luôn rỗng, các thiết bị sẽ không được sử dụng và hệ thống sẽ mất cân bằng. Hệ thống với năng lực tốt nhất sẽ có sự kết hợp các tiến trình hướng CPU và hướng nhập/xuất.

Trên một vài hệ thống, bộ định thời công việc có thể không có hay rất ít. Thí dụ, các hệ thống chia thời như UNIX thường không có bộ định thời công việc nhưng đơn giản đặt mỗi tiến trình mới vào bộ nhớ cho bộ định thời CPU. Khả năng ổn định của hệ thống này phụ thuộc vào giới hạn vật lý (như số lượng thiết bị cuối sẳn dùng) hay tính tự nhiên tự chuyển đổi của người dùng. Nếu năng lực thực hiện giảm tới mức độ không thể chấp nhận được thì một số người dùng sẽ thoát khỏi hệ thống.

Một số hệ thống như hệ chia thời có thể đưa vào một cấp độ định thời bổ sung hay định thời trung gian. **Bộ định thời trung gian** (medium-term process) này (được hiển thị trong lưu đồ hình 0-6) xóa các tiến trình ra khỏi bộ nhớ (từ sự canh tranh CPU) và do đó giảm mức độ đa chương. Tại thời điểm sau đó, tiến trình có thể được đưa trở lại bộ nhớ và việc thực thi của nó có thể được tiếp tục nơi nó bị đưa ra. Cơ chế này được gọi là **hoán vị** (swapping). Tiến trình được hoán vị ra và sau đó được hoán vị vào bởi bộ định thời trung gian. Hoán vị là cần thiết để cải tiến sự trộn lẫn tiến trình (giữa các tiến trình hướng nhập/xuất và hướng CPU), hay vì một thay đổi trong yêu cầu bộ nhớ vượt quá kích thước bộ nhớ sẵn dùng. Hoán vị sẽ được thảo luận trong chương sau.



Hình-Lưu đồ bổ sung định thời trung bình tới hàng đợi

1.3. Chuyển ngữ cảnh

Chuyển CPU tới một tiến trình khác yêu cầu lưu trạng thái của tiến trình cũ và nạp trạng thái được lưu cho tiến trình mới. Tác vụ này được xem như **chuyển ngữ cảnh** (context switch). **Ngữ cảnh** của tiến trình được hiện diện trong PCB của tiến trình; Nó chứa giá trị các thanh ghi, trạng thái tiến trình và thông tin quản lý bộ nhớ. Khi chuyển ngữ cảnh ngữ cảnh xảy ra, nhân lưu ngữ cảnh của tiến trình cũ trong PCB của nó và nạp ngữ cảnh được lưu của tiến trình mới được định thời để chạy. Thời gian chuyển ngữ cảnh là chi phí thuần vì hệ thống không thực hiện công việc có ích trong khi chuyển. Tốc độ của nó khác từ máy này tới máy khác phụ thuộc vào tốc độ bộ nhớ, số lượng thanh ghi phải được chép và sự tồn tại của các chỉ thị đặc biệt (như chỉ thị để nạp và lưu tất cả thanh ghi). Điển hình dãy tốc độ từ 1 tới 1000 mili giây.

Những lần chuyển đổi ngữ cảnh phụ thuộc nhiều vào hỗ trợ phần cứng. Thí dụ, vài bộ xử lý (như Sun UltraSPARC) cung cấp nhiều tập thanh ghi. Một chuyển ngữ cảnh đơn giản chứa chuyển đổi con trỏ tới tập thanh ghi hiện hành. Dĩ nhiên, nếu tiến trình hoạt động vượt quá tập thanh ghi thì hệ thống sắp xếp lại dữ liệu thanh ghi tới và từ bộ nhớ. Cũng vì thế mà hệ điều hành phức tạp hơn và nhiều công việc được làm hơn trong khi chuyển ngữ cảnh. Kỹ thuật quản lý bộ nhớ nâng cao có thể yêu cầu dữ liệu bổ sung để được chuyển với mỗi ngữ cảnh. Thí dụ, không gian địa chỉ của tiến trình hiện hành phải được lưu khi không gian của tác vụ kế tiếp được chuẩn bị dùng. Không gian địa chỉ được lưu như thế nào và lượng công việc được yêu cầu để lưu nó phụ thuộc vào phương pháp quản lý

bộ nhớ của hệ điều hành. Chuyển ngữ cảnh có thể dẫn đến thắt cổ chai năng lực thực hiện vì thế các lập trình viên đang sử dụng các cấu trúc mới để tránh nó bất cứ khi nào có thể.

1.4. Thao tác trên tiến trình

Các tiến trình trong hệ thống có thể thực thi đồng hành và chúng phải được tạo và xóa tự động. Do đó, hệ điều hành phải cung cấp một cơ chế (hay phương tiện) cho việc tạo tiến trình và kết thúc tiến trình .

Tiến trình là một chương trình đang thực thi. Khi một tiến trình thực thi, nó thay đổi trạng thái. Trạng thái của một tiến trình được định nghĩa bởi một hoạt động hiện tại của tiến trình đó. Mỗi tiến trình có thể ở một trong những trạng thái sau: mới (new), sẵn sàng (ready), đang chạy (running), chờ (waiting), hay kết thúc (terminated). Mỗi tiến trình được biểu diễn trong hệ điều hành bởi khối điều khiển tiến trình của chính nó (PCB).

Một tiến trình khi không thựcc thi, được đặt vào hàng đợi. Hai cấp chủ yếu của hàng đợi trong hệ điều hành là hàng đợi yêu cầu nhập/xuất và hàng đợi sẵn sàng. Hàng đợi sẵn sàng chứa tất cả tiến trình sẵn sàng để thực thi và đang chờ CPU. Mỗi tiến trình được biểu diễn bởi một PCB và các PCB có thể được liên kết với nhau để hình thành một hàng đợi sẵn sàng. **Định thời biểu dài** (long-term scheduling) (hay định thời biểu công việc) là chọn các tiến trình được phép cạnh tranh CPU. Thông thường, định thời biểu dài bị ảnh hưởng nặng nề bởi việc xem xét cấp phát tài nguyên, đặc biệt quản lý bộ nhớ. **Định thời ngắn** (short-term scheduling) là sự chọn lựa một tiến trình từ các hàng đợi sẵn sàng.

Các tiến trình trong hệ thống có thể thực thi đồng hành. Có nhiều lý do các thực thi đồng hành: chia sẻ thông tin, tăng tốc độ tính toán, hiệu chỉnh và tiện dụng. Thực thi đồng hành yêu cầu cơ chế cho việc tạo và xóa tiến trình .

Tiến trình thực thi trong hệ điều hành có thể là các tiến trình độc lập hay các tiến trình hợp tác. Các tiến trình hợp tác phải có phương tiện giao tiếp với nhau. Chủ yếu, có hai cơ chế giao tiếp bổ sung cho nhau cùng tồn tại: chia sẻ bộ nhớ và hệ thống truyền thông điệp. Phương pháp chia sẻ bộ nhớ yêu cầu các tiến trình giao tiếp chia sẻ một số biến. Các tiến trình được mong đợi trao đổi thông tin thông qua việc sử dụng các biến dùng chung này. Trong hệ thống bộ nhớ được chia sẻ, nhiệm vụ cho việc cung cấp giao tiếp tách rời với người lập trình ứng dụng; chỉ hệ điều hành cung cấp hệ thống bộ nhớ được chia sẻ. Phương pháp truyền thông điệp cho phép các tiến trình trong đổi thông điệp. Nhiệm vụ cung cấp giao tiếp có thể tách rời với hệ điều hành. Hai cơ chế này không loại trừ hỗ tương và có thể được dùng cùng một lúc trong phạm vi một hệ điều hành.

2. Các thuật toán lập lịch CPU cho tiến trình

Lập lịch CPU hay còn gọi định thời biểu CPU là cơ sở của các hệ điều hành đa chương. Bằng cách chuyển đổi CPU giữa các quá trình, hệ điều hành có thể làm máy tính hoạt động nhiều hơn. Trong hệ thống đa chương có nhiều quá trình chạy cùng thời điểm để tối ưu hóa việc sử dụng CPU, còn trong hệ thống đơn xử lý, chỉ một quá trình có thể chạy tại một thời điểm; bất cứ quá trình nào khác đều phải chờ cho đến khi CPU rảnh và có thể được định thời lại.

Ý tưởng của đa chương là tương đối đơn giản. Một quá trình được thực thi cho đến khi nó phải chờ yêu cầu nhập/xuất hoàn thành. Trong một hệ thống máy tính đơn giản thì CPU sẽ rảnh rỗi; tất cả thời gian chờ này là lãng phí. Với đa chương, chúng ta cố gắng dùng thời gian này để CPU có thể phục vụ cho các quá trình khác. Nhiều quá trình được giữ trong bộ nhớ tại cùng thời điểm. Khi một quá trình phải chờ, hệ điều hành lấy CPU từ quá trình này và cấp CPU tới quá trình khác.

Định thời biểu là chức năng cơ bản của hệ điều hành. Hầu hết tài nguyên máy tính được định thời biểu trước khi dùng. Dĩ nhiên, CPU là một trong những tài nguyên máy tính ưu tiên. Do đó, định thời biểu là trọng tâm trong việc thiết kế hệ điều hành.

Sự thành công của việc định thời biểu CPU phụ thuộc vào thuộc tính được xem xét sau đây của quá trình. Việc thực thi quá trình chứa một chu kỳ (cycle) thực thi CPU và chờ đợi nhập/xuất. Các quá trình chuyển đổi giữa hai trạng thái này. Sự thực thi quá trình bắt đầu với một chu kỳ CPU (CPU burst), theo sau bởi một chu kỳ nhập/xuất (I/O burst), sau đó một chu kỳ CPU khác, sau đó lại tới một chu kỳ nhập/xuất khác khác,... Sau cùng, chu kỳ CPU cuối cùng sẽ kết thúc với một yêu cầu hệ thống để kết thúc việc thực thi, hơn là với một chu kỳ nhập/xuất khác. Một chương trình hướng nhập/xuất (I/O-bound) thường có nhiều chu kỳ CPU ngắn. Một chương trình hướng xử lý (CPU-bound) có thể có một nhiều chu kỳ CPU dài. Sự phân bổ này có thể giúp chúng ta chọn giải thuật định thời CPU hợp lý.

2.1. Thuật toán FCFS (first-come, first-served)

Giải thuật định thời biểu CPU đơn giản nhất là **đến trước, được phục vụ trước** (first-come, first-served-FCFS). Với cơ chế này, quá trình yêu cầu CPU trước được cấp phát CPU trước. Việc cài đặt chính sách FCFS được quản lý dễ dàng với hàng đợi FIFO. Khi một quá trình đi vào hàng đợi sẵn sàng, PCB của nó được liên kết tới đuôi của hàng đợi. Khi CPU rảnh, nó được cấp phát tới một quá trình tại đầu hàng đợi. Sau đó, quá trình đang chạy được lấy ra khỏi hàng đợi. Mã của giải thuật FCFS đơn giản để viết và hiểu.

Tuy nhiên, thời gian chờ đợi trung bình dưới chính sách FCFS thường là dài. Xét tập hợp các quá trình sau đến tại thời điểm 0, với chiều dài thời gian chu kỳ CPU được cho theo mini giây.

Quá trình	Thời gian xử lý
P1	24
P2	3
P3	3

Nếu các quá trình đến theo thứ tự P_1, P_2, P_3 và được phục vụ theo thứ tự FCFS, chúng ta nhận được kết quả được hiển thị trong **lưu đồ Gantt** như sau:

P1	P2	P3
0	27	30

24		

Thời gian chờ là 0 mili giây cho quá trình P_1, 24 mili giây cho quá trình P_2 và 27 mili giây cho quá trình P_3. Do đó, thời gian chờ đợi trung bình là (0+24+27)/3=17 mili giây. Tuy nhiên, nếu các quá trình đến theo thứ tự P_2, P_3, P_1 thì các kết quả được hiển thị trong **lưu đồ Gannt** như sau:

P2	P3	P1
0 3	6	30

Thời gian chờ đợi trung bình bây giờ là (6+0+3)/3=3 mili giây. Việc cắt giảm này là quan trọng. Do đó, thời gian chờ đợi trung bình dưới chính sách FCFS thường không là tối thiểu và có sự thay đổi rất quan trọng nếu các thời gian CPU dành cho các quá trình khác nhau rất lớn.

Ngoài ra, xét năng lực của định thời FCFS trong trường hợp động. Giả sử chúng ta có một quá trình hướng xử lý (CPU-bound) và nhiều quá trình hướng nhập/xuất (I/O bound). Khi các quá trình đưa đến quanh hệ thống, ngữ cảnh sau có thể xảy ra. Quá trình hướng xử lý sẽ nhận CPU và giữ nó. Trong suốt thời gian này, tất cả quá trình khác sẽ kết thúc việc nhập/xuất của nó và chuyển vào hàng đợi sẵn sàng, các thiết bị nhập/xuất ở trạng thái rảnh. Cuối cùng, quá trình hướng xử lý kết thúc chu kỳ CPU của nó và chuyển tới thiết bị nhập/xuất. Tất cả các quá trình hướng xử lý có chu kỳ CPU rất ngắn sẽ nhanh chóng thực thi và di chuyển trở về hàng đợi nhập/xuất. Tại thời điểm này CPU ở trạng thái rảnh. Sau đó, quá trình hướng xử lý sẽ di chuyển trở lại hàng đợi sẵn sàng và được cấp CPU. Một lần nữa, tất cả quá trình hướng nhập/xuất kết thúc việc chờ trong hàng đợi sẵn sàng cho đến khi quá trình hướng xử lý được thực hiện. Có một **tác dụng phụ** (convoy effect) khi tất cả các quá trình khác chờ một quá trình lớn trả lại CPU. Tác dụng phụ này dẫn đến việc sử dụng thiết bị và CPU thấp hơn nếu các quá trình ngắn hơn được cấp trước.

Giải thuật FCSF là giải thuật định thời không trưng dụng CPU. Một khi CPU được cấp phát tới một quá trình, quá trình đó giữ CPU cho tới khi nó giải phóng CPU bằng cách kết thúc hay yêu cầu nhập/xuất. Giải thuật FCFS đặc biệt không phù hợp đối với hệ thống chia sẻ thời gian, ở đó mỗi người dùng nhận được sự chia sẻ CPU với những khoảng thời gian đều nhau.

2.2. Thuật toán SJF (shortest-job-first-SJF)

Lập lịch ưu tiên thời gian sử dụng CPU ngắn nhất trước. Giải thuật này gán tới mỗi quá trình chiều dài của chu kỳ CPU tiếp theo cho quá trình sau đó. Khi CPU sẵn dùng, nó được gán tới quá trình có chu kỳ CPU kế tiếp ngắn nhất. Nếu hai quá trình có cùng chiều dài chu kỳ CPU kế tiếp, định thời FCFS được dùng. Chú ý rằng thuật ngữ phù hợp hơn là chu kỳ CPU kế tiếp ngắn nhất (shortest next CPU burst) vì định thời được thực hiện bằng cách xem xét chiều dài của chu kỳ CPU kế tiếp của quá trình hơn là toàn bộ chiều dài của nó. Chúng

ta dùng thuật ngữ SJF vì hầu hết mọi người và mọi sách tham khảo tới nguyên lý của loại định thời biểu này như SJF.

Nguyên tắc:

- Gắn với mỗi tiến trình *độ dài loạt thời gian sử dụng CPU tiếp theo* của nó. Sử dụng các độ dài này để thực hiện việc lập lịch cho tiến trình có thời gian kết thúc ngắn nhất.
- Tiến trình có thời gian sử dụng CPU ngắn nhất sẽ sở hữu CPU.

Sử dụng theo 2 sơ đồ:

Không ưu tiên: khi CPU được dành cho 1 tiến trình, nó sẽ không bị tước CPU đến khi thời gian sử dụng của nó kết thúc.

Ưu tiên: nếu 1 tiến trình mới đưa vào danh sách sẵn sàng có thời gian sử dụng CPU nhỏ hơn thời gian còn lại của tiến trình đang thi hành, CPU được dành lại cho tiến trình mới (thời gian còn lại ngắn nhất Shortest-Remaining-Time-First (SRTF)).

SJF là 1 giải thuật tối ưu, cho thời gian chờ trung bình nhỏ nhất.

Giải thuật SJF có thể là tối ưu, trong đó nó cho thời gian chờ đợi trung bình nhỏ nhất cho các quá trình được cho. Bằng cách chuyển một quá trình ngắn trước một quá trình dài thì thời gian chờ đợi của quá trình ngắn giảm hơn so với việc tăng thời gian chờ đợi của quá trình dài. Do đó, thời gian chờ đợi trung bình giảm.

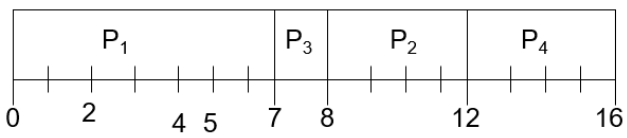
Khó khăn thật sự với giải thuật SJF là làm thế nào để biết chiều dài của yêu cầu CPU tiếp theo. Đối với định thời dài trong hệ thống bó, chúng ta có thể dùng chiều dài như giới hạn thời gian xử lý mà người dùng xác định khi gởi công việc. Do đó, người dùng được cơ động để ước lượng chính xác giới hạn thời gian xử lý vì giá trị thấp hơn có nghĩa là đáp ứng nhanh hơn. Định thời SJF được dùng thường xuyên trong định thời dài.

Ví dụ:

Ví dụ về giải thuật SJF không mức ưu tiên

Process	Arrival Time ($t_{nạp}$)	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- SJF (non-preemptive)



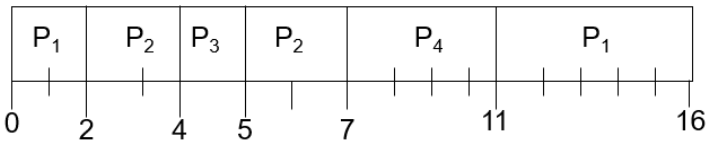
$T_{chờ} = t_{kết\ thúc} - t_{thực\ hiện} - t_{nạp}$

$t_{chờ\ TB} = (0 + 6 + 3 + 7)/4 = 4$

Ví dụ về giải thuật SJF với mức ưu tiên

Process	Arrival Time	Burst Time
P_1	0.0	7(5)
P_2	2.0	4(2)(0)
P_3	4.0	1(0)
P_4	5.0	4

- SJF (preemptive)



$T_{chờ} = t_{kết\ thúc} - t_{thực\ hiện} - t_{nạp}$

Average waiting time = (9 + 1 + 0 +2)/4 = 3

2.3. Thuật toán lập lịch theo độ ưu tiên

Giải thuật SJF là trường hợp đặc biệt của giải thuật định thời theo độ ưu tiên (priority-scheduling algorithm). Độ ưu tiên được gán với mỗi quá trình và CPU được cấp phát tới quá trình với độ ưu tiên cao nhất. Quá trình có độ ưu tiên bằng nhau được định thời trong thứ tự FCFS.

Giải thuật SJF là giải thuật ưu tiên đơn giản ở đó độ ưu tiên p là nghịch đảo với chu kỳ CPU được đoán tiếp theo. Chu kỳ CPU lớn hơn có độ ưu tiên thấp hơn và ngược lại.

Bây giờ chúng ta thảo luận định thời có độ ưu tiên cao và thấp. Các độ ưu tiên thường nằm trong dãy số cố định, chẳng hạn 0 tới 7 hay 0 tới 4,095. Tuy nhiên, không có sự thoả thuận chung về 0 là độ ưu tiên thấp nhất hay cao nhất. Một vài hệ thống dùng số thấp để biểu diễn độ ưu tiên thấp; ngược lại các hệ thống khác dùng các số thấp cho độ ưu tiên cao. Sự khác nhau này có thể dẫn đến sự lẫn lộn. Trong giáo trình này chúng ta dùng các số thấp để biểu diễn độ ưu tiên cao.

Thí dụ, xét tập hợp quá trình sau đến tại thời điểm 0 theo thứ tự P1, P2,…, P5 với chiều dài thời gian chu kỳ CPU được tính bằng mili giây:

Quá trình	Thời gian xử lý	Độ ưu tiên
P1	10	3
P2	1	1
P3	2	4
P4	1	5
P5	5	2

Sử dụng định thời theo độ ưu tiên, chúng ta sẽ định thời các quá trình này theo lưu đồ Gannt như sau:

P2	P5	P1	P3	P4
0	1 6	16	18	19

Thời gian chờ đợi trung bình là 8.2 mili giây.

Độ ưu tiên có thể được định nghĩa bên trong hay bên ngoài. Độ ưu tiên được định nghĩa bên trong thường dùng định lượng hoặc nhiều định lượng có thể đo để tính toán độ ưu tiên của một quá trình. Thí dụ, các giới hạn thời gian, các yêu cầu bộ nhớ, số lượng tập tin đang mở và tỉ lệ của chu kỳ nhập/xuất trung bình với tỉ lệ của chu kỳ CPU trung bình. Các độ ưu tiên bên ngoài được thiết lập bởi các tiêu chuẩn bên ngoài đối với hệ điều hành như sự quan trọng của quá trình, loại và lượng chi phí đang được trả cho việc dùng máy tính, văn phòng hỗ trợ công việc, ..

Định thời biểu theo độ ưu tiên có thể trưng dụng hoặc không trưng dụng CPU. Khi một quá trình đến hàng đợi sẵn sàng, độ ưu tiên của nó được so sánh với độ ưu tiên của quá trình hiện đang chạy. Giải thuật định thời theo độ ưu tiên trưng dụng sẽ chiếm CPU nếu độ ưu tiên của quá trình mới đến cao hơn độ ưu tiên của quá trình đang thực thi. Giải thuật định thời theo độ ưu tiên không trưng dụng sẽ đơn giản đặt quá trình mới tại đầu hàng đợi sẵn sàng.

Vấn đề chính với giải thuật định thời theo độ ưu tiên là nghẽn không hạn định (indefinite blocking) hay đói CPU (starvation). Một quá trình sẵn sàng chạy nhưng thiếu CPU có thể xem như bị nghẽn-chờ đợi CPU. Giải thuật định thời theo độ ưu tiên có thể để lại nhiều quá trình có độ ưu tiên thấp chờ CPU không hạn định. Trong một hệ thống máy tính tải cao, dòng đều đặn các quá trình có độ ưu tiên cao hơn có thể ngăn chặn việc nhận CPU của quá trình có độ ưu tiên thấp.. Thông thường, một trong hai trường hợp xảy ra. Cuối cùng, một quá trình sẽ được chạy (lúc 2 a.m chủ nhật là thời điểm cuối cùng hệ thống nạp các quá trình nhẹ), hay cuối cùng hệ thống máy tính sẽ đổ vỡ và mất tất cả các quá trình có độ ưu tiên thấp chưa được kết thúc.

Một giải pháp cho vấn đề nghẽn không hạn định này là sự hoá già (aging). Hóa già là kỹ thuật tăng dần độ ưu tiên của quá trình chờ trong hệ thống một thời gian dài. Thí dụ, nếu các độ ưu tiên nằm trong dãy từ 127 (thấp) đến 0 (cao), chúng ta giảm độ ưu tiên của quá trình đang chờ xuống 1 mỗi 15 phút. Cuối cùng, thậm chí một quá trình với độ ưu tiên khởi đầu 127 sẽ đạt độ ưu tiên cao nhất trong hệ thống và sẽ được thực thi. Thật vậy, một quá trình sẽ mất không quá 32 giờ để đạt được độ ưu tiên từ 127 tới 0.

2.4. Thuật toán luân phiên (round-robin scheduling algorithm-RR)

thiết kế đặc biệt cho hệ thống chia sẻ thời gian. Tương tự như định thời FCFS nhưng sự trưng dụng CPU được thêm vào để chuyển CPU giữa các quá trình. Đơn vị thời gian nhỏ được gọi là định mức thời gian (time quantum) hay phần thời gian (time slice) được định nghĩa. Định mức thời gian thường từ 10 đến 100 mili giây. Hàng đợi sẵn sàng được xem như một hàng đợi vòng. Bộ định thời CPU di chuyển vòng quanh hàng đợi sẵn sàng, cấp phát CPU tới mỗi quá trình có khoảng thời gian tối đa bằng một định mức thời gian.

Để cài đặt định thời RR, chúng ta quản lý hàng đợi sẵn sàng như một hàng đợi FIFO của các quá trình. Các quá trình mới được thêm vào đuôi hàng đợi. Bộ định thời CPU chọn quá trình đầu tiên từ hàng đợi sẵn sàng, đặt bộ đếm thời gian để ngắt sau 1 định mức thời gian và gởi tới quá trình.

Sau đó, một trong hai trường hợp sẽ xảy ra. Quá trình có 1 chu kỳ CPU ít hơn 1 định mức thời gian. Trong trường hợp này, quá trình sẽ tự giải phóng. Sau đó, bộ định thời biểu sẽ xử lý quá trình tiếp theo trong hàng đợi sẵn sàng. Ngược lại, nếu chu kỳ CPU của quá trình đang chạy dài hơn 1 định mức thời gian thì độ đếm thời gian sẽ báo và gây ra một ngắt tới hệ điều hành. Chuyển đổi ngữ cảnh sẽ được thực thi và quá trình được đặt trở lại tại đuôi của hàng đợi sẵn sàng. Sau đó, bộ định thời biểu CPU sẽ chọn quá trình tiếp theo trong hàng đợi sẵn sàng.

Tuy nhiên, thời gian chờ đợi trung bình dưới chính sách RR thường là quá dài. Xét một tập hợp các quá trình đến tại thời điểm 0 với chiều dài thời gian CPU-burst được tính bằng mili giây:

Quá trình	Thời gian xử lý
P1	24
P2	3
P3	3

Nếu sử dụng định mức thời gian là 4 mili giây thì quá trình P_1 nhận 4 mili giây đầu tiên. Vì nó yêu cầu 20 mili giây còn lại nên nó bị trưng dụng CPU sau định mức thời gian đầu tiên và CPU được cấp tới quá trình tiếp theo trong hàng đợi, quá trình P_2. Vì P_2 không cần tới 4 mili giây nên nó kết thúc trước khi định mức thời gian của nó hết hạn. Sau đó, CPU được cho tới quá trình kế tiếp, quá trình P_3. Một khi mỗi quá trình nhận 1 định mức thời gian thì CPU trả về quá trình P_1 cho định mức thời gian tiếp theo. Thời biểu RR là:

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

Thời gian chờ đợi trung bình là 17/3=5.66 mili giây.

Trong giải thuật RR, không quá trình nào được cấp phát CPU cho nhiều hơn 1 định mức thời gian trong một hàng. Nếu chu kỳ CPU của quá trình vượt quá 1 định mức thời gian thì quá trình đó bị trưng dụng CPU và nó được đặt trở lại hàng đợi sẵn sàng. Giải thuật RR là giải thuật trưng dụng CPU.

Nếu có n quá trình trong hàng đợi sẵn sàng và định mức thời gian là q thì mỗi quá trình nhận 1/n thời gian CPU trong các phần, nhiều nhất q đơn vị thời gian. Mỗi quá trình sẽ chờ không dài hơn (n-1)x q đơn vị thời gian cho tới khi định mức thời gian tiếp theo của nó. Thí dụ, nếu có 5 quá trình với định mức thời gian 20 mili giây thì mỗi quá trình sẽ nhận 20 mili giây sau mỗi 100 mili giây.

Năng lực của giải thuật RR phụ thuộc nhiều vào kích thước của định mức thời gian. Nếu định mức thời gian rất lớn (lượng vô hạn) thì chính sách RR tương tự như chính sách FCFS. Nếu định mức thời gian là rất nhỏ (1 mili giây)

thì tiếp cận RR được gọi là **chia sẻ bộ xử lý** (processor sharing) và xuất hiện (trong lý thuyết) tới người dùng như thể mỗi quá trình trong n quá trình có bộ xử lý riêng của chính nó chạy tại 1/n tốc độ của bộ xử lý thật.

Đánh giá giải thuật

Chúng ta chọn một giải thuật định thời CPU cho một hệ thống xác định như thế nào? Có nhiều giải thuật định thời, mỗi giải thuật với các tham số của riêng nó. Do đó, chọn một giải thuật có thể là khó.

Vấn đề đầu tiên là định nghĩa các tiêu chuẩn được dùng trong việc chọn một giải thuật. Các tiêu chuẩn thường được định nghĩa trong thuật ngữ khả năng sử dụng CPU, thời gian đáp ứng hay thông lượng. Để chọn một giải thuật, trước hết chúng ta phải định nghĩa trọng số quan trọng của các thước đo này. Tiêu chuẩn của chúng ta có thể gồm các thước đo như:

• Khả năng sử dụng CPU tối đa dưới sự ràng buộc thời gian đáp ứng tối đa là 1 giây.

• Thông lượng tối đa như thời gian hoàn thành (trung bình) tỉ lệ tuyến tính với tổng số thời gian thực thi.

Một khi các tiêu chuẩn chọn lựa được định nghĩa, chúng ta muốn đánh giá các giải thuật khác nhau dưới sự xem xét.

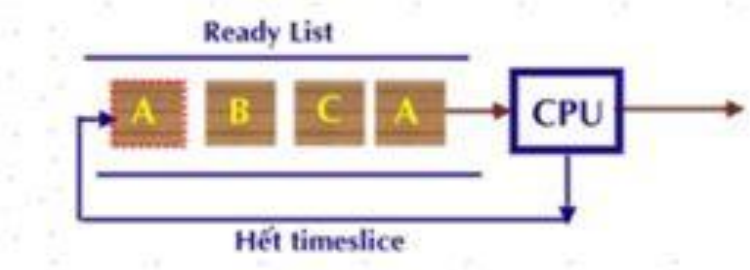
CÂU HỎI VÀ BÀI TẬP

Câu 1: Lập lịch non-preemptive là:
A. Lập lịch ưu tiên
B. Lập lịch ngắn
C. Lập lịch dài
D. Lập lịch không ưu tiên.

Câu 2: Lập lịch preemptive là:
A. Lập lịch không ưu tiên
B. Lập lịch ngắn
C. Lập lịch dài
D. Lập lịch ưu tiên.

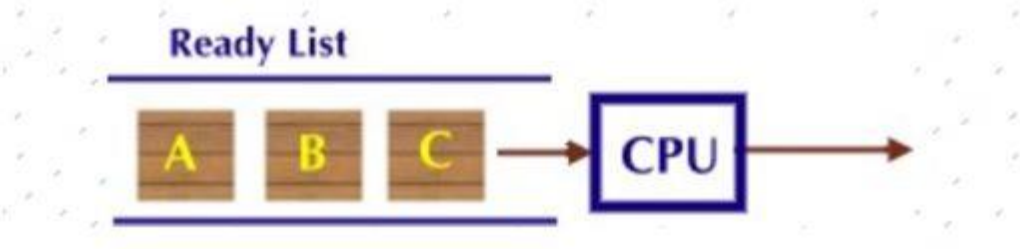
Câu 3: Lập lịch preemptive là:
A. Một tiến trình giữ I/O đến khi nó kết thúc hoặc chuyển sang trạng thái waiting.
B. Tiến trình được phép giữ CPU đến khi kết thúc.
C. Tiến trình được phép thực hiện trong khoảng thời gian, hệ thống không có quyền ngắt CPU.
D. Tiến trình chỉ được phép thực hiện trong khoảng thời gian, hệ thống có quyền ngắt CPU bất cứ lúc nào.

Câu 4: *Hình sau thuộc về loại lập lịch nào:*



A. FCFS
B. SJF
C. RR
D. Ưu tiên

Câu 5: Hình sau mô tả loại lập lịch gì:



A. FCFS
B. Ưu tiên
C. SJF
D. RR

Câu 6: *Đoạn sau mô tả lập lịch gì:*

Bộ điều phối lần lượt cấp phát cho từng tiến trình trong danh sách một khoảng thời gian sử dụng CPU gọi là *quantum*

A. FCFS
B. Ưu tiên
C. SJF
D. RR

Câu 7: *Câu dưới đây thuộc về loại lập lịch nào:*

Lập lịch bao gồm cả hai trường hợp ưu tiên và không ưu tiên?

A. FCFS
B. Ưu tiên
C. SJF
D. RR

Câu 8: Chọn câu trả lời đúng:

Việc lập lịch CPU sẽ được kích hoạt khi 1 tiến trình ở 1 trong các trạng thái

A. Chuyển từ trạng thái đang chạy sang trạng thái chờ, Chuyển từ trạng thái đang chờ sang trạng thái sẵn sàng

B. Chuyển từ trạng thái đang chạy sang trạng thái sẵn sàng, Chuyển từ trạng thái đang chờ sang trạng thái sẵn sàng

C. Chuyển từ trạng thái đang chạy sang trạng thái chờ, Chuyển từ trạng

thái đang chạy sang trạng thái sẵn sàng

D. Chuyển từ trạng thái đang chạy sang trạng thái chờ, Chuyển từ trạng thái đang chờ sang trạng thái sẵn sàng, Chuyển từ trạng thái đang chạy sang trạng thái sẵn sàng

Câu 9: *Chọn câu trả lời đúng:*

Lập lịch nào được kích hoạt theo điều kiện ưu tiên

A. Chuyển từ trạng thái chạy sang trạng thái sẵn sàng

B. Kết thúc

C. Chuyển từ trạng thái đang chạy sang trạng thái chờ

D. Chuyển từ trạng thái đang chờ sang trạng thái sẵn sàng

Câu 10: *Chọn câu trả lời đúng:*

Lập lịch nào được kích hoạt theo điều kiện không ưu tiên

A. Chuyển từ trạng thái chạy sang trạng thái sẵn sàng

B. Chuyển từ trạng thái chờ sang trạng thái sẵn sàng

C. Chuyển từ trạng thái đang chạy sang trạng thái chờ

D. Chuyển từ trạng thái sẵn sàng sang trạng thái chạy

Câu 11: *Chọn câu trả lời đúng:*

Modul điều phối trao quyền điều khiển CPU cho tiến trình được chọn bởi bộ lập lịch ngắn hạn; nó thực hiện công việc:

A. Chuyển đổi ngữ cảnh, Chuyển sang chế độ người dùng.

B. Chuyển sang chế độ người dùng, Nhảy đến vị trí chính xác trong chương trình người dùng để khởi động lại chương trình đó.

C. Chuyển đổi ngữ cảnh, Nhảy đến vị trí chính xác trong chương trình người dùng để khởi động lại chương trình đó.

D. Chuyển đổi ngữ cảnh, Chuyển sang chế độ người dùng, Nhảy đến vị trí chính xác trong chương trình người dùng để khởi động lại chương trình đó.

Câu 12: *Chọn câu trả lời đúng:*

Trong các tiêu chuẩn tối ưu về lập lịch, cần tận dụng tối đa ở tài nguyên nào

A. Tận dụng tối đa CPU và Tận dụng tối đa thời gian chờ

B. Tận dụng thông lượng tối đa và Tận dụng tối đa thời gian chờ

C. Tận dụng tối đa thời gian chờ

D. Tận dụng tối đa CPU và Tận dụng thông lượng tối đa

Bài tập

Bài 1: Cho bảng thông tin của các tiến trình. Thời gian chờ đợi trung bình theo giải thuật FCFS là:

Thứ tự tiến trình (Processes)	Thời điểm nạp (kích hoạt) (Arrival Time)	Thời gian hoạt động (Burst Time)
P1	0	3
P2	0	5
P3	0	4
P4	0	6

Bài 2: Cho bảng thông tin của các tiến trình. Thời gian chờ đợi trung bình theo giải thuật FCFS là:

Thứ tự tiến trình (Processes)	Thời điểm nạp (kích hoạt) (Arrival Time)	Thời gian hoạt động (Burst Time)
P1	0	3
P2	2	5
P3	3	4
P4	4	6

Bài 3: Cho bảng thông tin của các tiến trình. Thời gian chờ đợi trung bình theo giải thuật SJF độc quyền CPU là:

Thứ tự tiến trình (Processes)	Thời điểm nạp (kích hoạt) (Arrival Time)	Thời gian hoạt động (Burst Time)
P1	0	3
P2	2	5
P3	3	4
P4	4	6

Bài 4: Cho bảng thông tin của các tiến trình. Thời gian chờ đợi trung bình theo giải thuật SJF không độc quyền CPU là:

Thứ tự tiến trình (Processes)	Thời điểm nạp (kích hoạt) (Arrival Time)	Thời gian hoạt động (Burst Time)
P1	0	5
P2	1	1
P3	2	2
P4	3	4

Bài 5: Cho bảng thông tin của các tiến trình. Thời gian chờ đợi trung bình theo giải thuật độc quyền CPU có thứ tự ưu tiên là:

Thứ tự tiến trình (Processes)	Thời điểm nạp (kích hoạt) (Arrival Time)	Thời gian hoạt động (Burst Time)	Thứ tự ưu tiên
P1	0	3	1
P2	0	5	4
P3	0	4	3

Bài 6: Cho bảng thông tin của các tiến trình. Thời gian chờ đợi trung bình theo giải thuật Round Robin với thời gian lượng tử q=3:

Thứ tự tiến trình (Processes)	Thời điểm nạp (kích hoạt) (Arrival Time)	Thời gian hoạt động (Burst Time)
P1	0	5
P2	0	3
P3	0	6
P4	0	9

Bài 7: Cho bảng thông tin của các tiến trình. Thời gian chờ đợi trung bình theo giải thuật Round Robin với thời gian lượng tử q=3:

Thứ tự tiến trình (Processes)	Thời điểm nạp (kích hoạt) (Arrival Time)	Thời gian hoạt động (Burst Time)
P1	0	5
P2	1	3
P3	2	6
P4	3	9