**Lập lịch trình**

**Giới thiệu về lập lịch trình**

* **Sử dụng CPU tối đa có được với đa chương trình**
* **Chu kì bùng nổ I/O của CPU - Quá trình thực thi bao gồm chu kì thực thi CPU và chờ I/O**
* **Phân phối bùng nổ CPU**

* **Sự bùng nổ của việc sử dụng CPU xen kẽ với khoảng thời gian chờ đợi I / O**
* **Quy trình ràng buộc CPU:** có nghĩa là tốc độ tiến trình của quá trình bị giới hạn bởi tốc độ của CPU.
* **Quy trình ràng buộc I/O:** nghĩa là tốc độ tiến trình của một quá trình bị giới hạn bởi tốc độ của hệ thống con I / O.

* **Lập lịch ba cấp**: chia quá trình [lập lịch](https://de.wikipedia.org/wiki/Scheduling) thành 3 quá trình phụ:
* Bộ lập lịch nhập học
* Bộ lập lịch bộ nhớ
* Bộ lập lịch CPU

Khi bắt đầu quá trình lập lịch trình này, có một [hàng đợi đầu vào](https://de.wikipedia.org/wiki/Warteschlange_(Datenstruktur)) . Đây chỉ đơn giản là một cấu trúc dữ liệu tương tự như một [danh sách](https://de.wikipedia.org/wiki/Datenstruktur#Grundlegende_Datenstrukturen) trong đó các “công việc” mới đến được lưu trữ tạm thời.  Bộ lập *lịch tuyển đầu vào* đã có hiệu lực. nó quyết định công việc nào đi vào hệ thống xa hơn (chính xác hơn là vào bộ nhớ). Các công việc sau đó được tải vào bộ nhớ [chính](https://de.wikipedia.org/wiki/Arbeitsspeicher) bởi bộ lập lịch nhập. Đây là nơi công việc của *bộ lập lịch bộ nhớ* bắt đầu. Vì mọi công việc đều yêu cầu dung lượng trong bộ nhớ và có thể xảy ra một số công việc trong bộ nhớ, bộ lập lịch bộ nhớ sẽ quyết định công việc nào còn lại trong bộ nhớ chính và công việc nào được hoán đổi trên đĩa cứng. Bộ lập *lịch CPU* bây giờ nhận một công việc từ bộ nhớ và xử lý nó

* Lựa chọn trong số các quy trình trong bộ nhớ đã sẵn sàng thực thi và phân bổ CPU cho một trong số chúng
* Các quyết định lập lịch trình CPU có thể diễn ra khi một quá trình:

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

2. Chuyển từ trạng thái đang chạy sang trạng thái sẵn sàng

3. Chuyển từ trạng thái chờ đợi hoặc quy trình mới được tạo để sẵn sàng

4. Chấm dứt

* Thuật toán lập lịch trình không ưu tiên: chọn quy trình và để nó chạy cho đến khi nó chặn hoặc cho đến khi nó tự nguyện giải phóng CPU
* Thuật toán lập lịch ưu tiên: chọn quy trình và để nó chạy trong thời gian khắc phục tối đa

**Trạng thái của Process**

Khi 1 process chạy, nó thay đổi trạng thái. Trạng thái của process được định nghĩa là 1 phần của hoạt động hiện tại. Process sẽ có 1 trong những trạng thái sau:

1. **New** - Process được khởi tạo
2. **Running** - các đoạn code được chạy
3. **Waiting**- Process đang đợi các sự kiện như I/O...
4. **Ready** - Process đang đợi để được gán vào processor
5. **Terminated** - Process kết thúc

Trạng thái của tiến trình tại một thời điểm được xác định bởi hoạt động hiện thời của tiến trình tại thời điếm đó.

Trong quá trình sống một tiến trình thay đổi trạng thái do nhiều nguyên nhân như: phải chờ một sự kiện nào đó xảy ra ,hay đợi một thao tác nhập xuất hoàn tất, buộc phải dừng hoạt động do hết thời gian xử lý……

**Tiêu chí lập lịch trình**

**Sử dụng CPU** - giữ cho CPU bận rộn nhất có thể

**Thông lượng** - # quy trình hoàn thành việc thực thi chúng trên mỗi đơn vị thời gian

**Thời gian quay vòng** - lượng thời gian để thực hiện một quy trình cụ thể

**Thời gian chờ** - khoảng thời gian mà một quá trình đã được chờ đợi trong hàng đợi sẵn sàng

**Thời gian phản hồi** - lượng thời gian cần từ khi gửi yêu cầu cho đến khi phản hồi đầu tiên được tạo, không phải đầu ra (đối với môi trường chia sẻ thời gian)

**Tiêu chí tối ưu hóa:**

Sử dụng CPU tối đa

Thông lượng tối đa

Thời gian quay vòng tối thiểu

Thời gian chờ tối thiểu

Thời gian phản hồi tối thiểu

**Mục tiêu của thuật toán lập lịch**

**Tất cả các hệ thống**

**Công bằng** - chia sẻ công bằng cho mỗi quá trình của CPU

**Thực thi chính sách** - thấy rằng chính sách đã nêu được thực hiện

**Cân bằng** - giữ cho tất cả các bộ phận của hệ thống bận rộn

**Hệ thống hàng loạt**

**Thông lượng** - tối đa hóa công việc mỗi giờ

**Thời gian quay vòng** - giảm thiểu thời gian từ khi gửi đến khi kết thúc

**Sử dụng CPU** - giữ cho CPU luôn bận rộn

**Hệ thống tương tác**

**Thời gian phản hồi** - phản hồi yêu cầu nhanh chóng

**Tương xứng** - đáp ứng mong đợi của người dùng

**Hệ thống thời gian thực**

**Thời hạn họp**- tránh mất dữ liệu

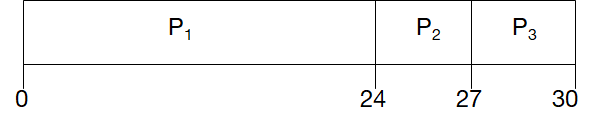
**Khả năng dự đoán** - tránh suy giảm chất lượng trong các hệ thống đa phương tiện

**Lập lịch trình trong hệ thống hàng loạt**

**Lập lịch cho người đến trước, người phục vụ trước (FCFS)**

Giả sử rằng các quá trình đến theo thứ tự: P1, P2, P3 Biểu đồ Gantt cho lịch trình là:





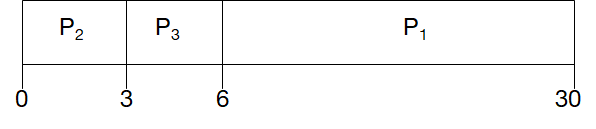
Thời gian chờ P1 = 0; P2 = 24; P3 = 27

Thời gian chờ trung bình: (0 + 24 + 27) / 3 = 17

Giả sử rằng các quy trình đến theo thứ tự

P2, P3, P1

Biểu đồ Gantt cho lịch trình là:



Thời gian chờ P1 = 6; P2 = 0; P3 = 3

Thời gian chờ trung bình: (6 + 0 + 3) / 3 = 3

Tốt hơn nhiều so với trường hợp trước

Hiệu ứng đoàn xe quá trình ngắn đằng sau quá trình dài

**Lập lịch trình công việc ngắn nhất (SJF)**

\*Liên kết với mỗi quá trình độ dài của lần bùng nổ CPU tiếp theo của nó. Sử dụng các độ dài này để lập  lịch cho quá trình với thời gian ngắn nhất

\*Hai phưong án:

-**không ưu tiên** - một khi CPU được cung cấp cho quá trình, nó không thể được ưu tiên cho đến khi hoàn thành đợt bùng nổ CPU của nó

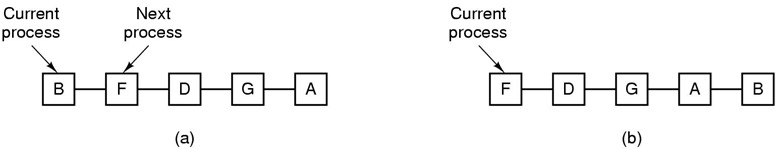
-**ưu tiên** - nếu một quá trình mới đến với thời lượng bùng nổ CPU ít hơn thời gian còn lại của quy trình thực thi hiện tại. Lược đồ này gọi là thời gian đầu tiên còn lại ngắn nhất (SRTF)

\*SJF là tối ưu - cung cấp thời gian chờ trung bình tối thiểu cho một tập hợp các quy trình nhất định



Một ví dụ về lập lịch trình đầu tiên cho công việc ngắn nhất

**Lập lịch trong Hệ thống tương tác**



Lập lịch vòng quay

danh sách các quy trình có thể chạy được (a)

danh sách các quy trình có thể chạy sau khi B sử dụng hết lượng tử của nó (b)

**Vòng quay (RR)`**

Mỗi quá trình nhận được một đơn vị thời gian CPU nhỏ (lượng tử thời gian), thường là 10 - 100 mili giây. Sau khi thời gian này trôi qua, quy trình được ưu tiên và thêm vào cuối hàng đợi sẵn sàng.

Nếu có n quy trình trong hàng đợi sẵn sàng và lượng tử thời gian là q, thì mỗi quy trình sẽ nhận được 1 / n thời gian CPU với nhiều nhất là q đơn vị thời gian cùng một lúc. Không có quá trình nào chờ nhiều hơn (n-1) q đơn vị thời gian.

Hiệu suất

*q lớn ⇒ FIFO*

*q nhỏ ⇒ q phải lớn đối với chuyển đổi ngữ cảnh, nếu không thì chi phí quá cao*

Ví dụ về RR với Lượng tử thời gian = 20

Quá trình Thời gian bùng nổ

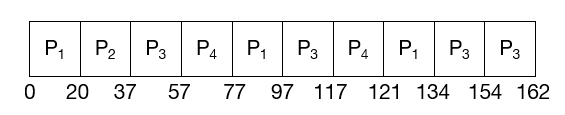
P1 53

P2 17

P3 68

P4 24

Biểu đồ Gantt là:



Thông thường, quay vòng trung bình cao hơn SJF, nhưng phản hồi tốt hơn

**Lập lịch ưu tiên :** Số ưu tiên (số nguyên) được liên kết với mỗi quy trình  -  CPU được phân bổ cho quá trình với mức ưu tiên cao nhất

Ưu tiên

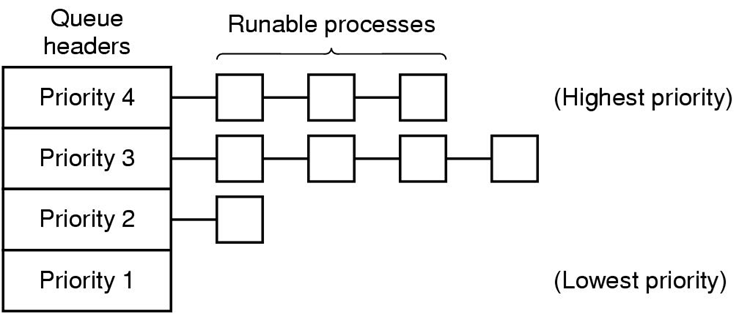
không ưu tiên

SJF là lập lịch ưu tiên trong đó ưu tiên là thời gian bùng nổ CPU tiếp theo được dự đoán

**Vấn đề Đói** - quá trình ưu tiên thấp có thể không bao giờ thực hiện

**Giải pháp tuổi** - theo thời gian tiến triển làm tăng mức độ ưu tiên của quy trình

Một thuật toán lập lịch với bốn lớp ưu tiên



**Lập lịch trong Hệ thống thời gian thực**

**Hệ thống thời gian thực cứng** - cần thiết để hoàn thành một nhiệm vụ quan trọng trong một khoảng thời gian được bảo đảm

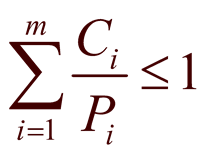
**Điện toán thời gian thực mềm** - yêu cầu các quy trình quan trọng nhận được ưu tiên hơn so với các quy trình kém may mắn hơn

**Hệ thống thời gian thực có thể lập lịch**

m sự kiện định kỳ

sự kiện i xảy ra trong khoảng thời gian Pi và yêu cầu Ci giây

Sau đó, tải chỉ có thể được xử lý nếu:



**Chính sách so với Cơ chế**

Phân tách những gì được phép thực hiện với cách nó được thực hiện

-một quá trình biết chủ đề con nào là quan trọng và cần ưu tiên

Thuật toán lập lịch tham số

-cơ chế trong kernel

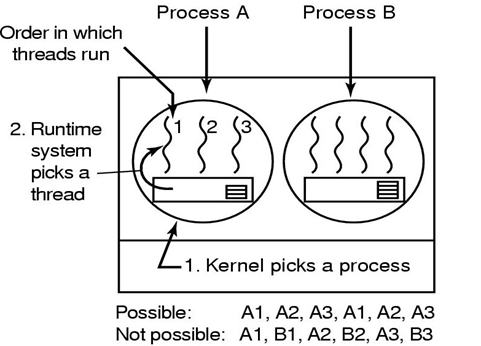
Các tham số được điền bởi các quy trình người dùng

-chính sách được thiết lập bởi quy trình người dùng

**Lập lịch chuỗi**

**Lập lịch cục bộ** - cách thư viện chủ đề quyết định chủ đề nào sẽ được đưa vào

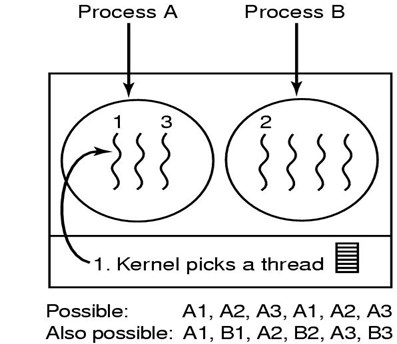
**Lập lịch toàn cục** - Cách kernel quyết định luồng kernel nào sẽ chạy tiếp theo



**Lập lịch có thể cho các chủ đề cấp người dùng**

Quy trình lượng tử 50 msec

luồng chạy 5 msec / CPU bùng nổ



**Lập lịch có thể của các luồng cấp hạt nhân**

\*Lượng tử quá trình 50-msec

\*luồng chạy 5 msec / CPU bùng nổ