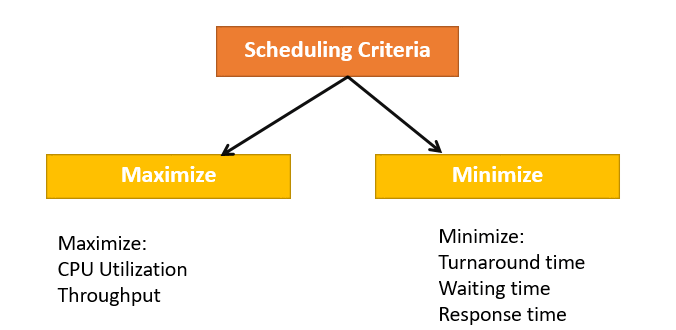
1. **Tiêu Chí Lập Lịch**

* Trong Tiêu Chí lập lịch CPU sẽ có 2 tiêu chí: tối đa hóa và giảm thiểu



* **Tối đa hóa gồm những gì ???**
* Sử dụng CPU: là nhiệm vụ chính mà hệ điều hành cần có, để đảm bảo rằng CPU hoạt động tối đa công suất có thể. Nó có thể nằm trong khoảng từ 0% đến 100%. Tuy nhiên, đối với **Hệ điều hành thời gian thực**, nó có thể dao động từ 40% đối với hệ thống cấp thấp và 90% đối với hệ thống cao cấp.
* Thông lượng(**Throughput**): Nó là tổng số quá trình được hoàn thành trên một đơn vị thời gian hay nói đúng hơn là tổng khối lượng công việc được thực hiện trong một đơn vị thời gian, và khi CPU đang bận thực hiện tiến trình, tại thời điểm đó, công việc đang được thực hiện và công việc được hoàn thành trên một đơn vị thời gian được gọi là Thông lượng.
* **Giảm Thiểu gồm những gì ???**
* Thời gian chờ đợi: là khoảng thời gian cụ thể cần phải có sẵn trong hàng đợi, để sẵn sàng được kiểm soát và sử dụng trên CPU
* Thời gian đáp ứng(thời gian phản hồi): là khoảng thời gian từ lúc yêu cầu gửi đi, cho đến khi phản hồi được đáp ứng(tạo ra), chú ý: đây là khoảng thời gian đến khi bắt đầu tiến chình chứ không phải là thời gian hoàn thành tiến trình
* Thời gian quay vòng: là khoảng thời gian để nó thực hiện 1 tiến trình cụ thể, tức là Khoảng thời gian từ khi gửi quy trình đến thời điểm hoàn thành quy trình
* Nhìn chung, việc sử dụng CPU và Thông lượng được tối đa hóa và các yếu tố khác được giảm bớt để tối ưu hóa phù hợp.
* **Interval Timer(bộ hẹn giờ khoảng):**
* Ngắt hẹn giờ là một phương pháp có liên quan mật thiết đến quyền ưu tiên.
* Khi một quy trình bất kì được phân bổ bởi CPU, khi đó một bộ đếm thời gian có thể được đặt thành một khoảng thời gian nhất định.
* Cả bộ hẹn giờ và chế độ ưu tiên sẽ buộc một quá trình trả lại CPU khi tiến trình xảy ra lỗi
* Hầu hết hệ điều hành nó sẽ sử dụng một số dạng đếm thời gian để ngăn một số tiến trình hoạt động mãi mãi
* **Dispatcher(điều phối viên):**
* Điều phối viên là gì: nó là một mô-đun cung cấp quyền kiểm soát CPU đối với các tiến trình
* Điều phối viên thì nó phải nhanh, để nó có thể chạy tên mọi ngữ cảnh, và độ trễ của nó là lượng thời gian cần thiết để bộ lập lịch CPU dừng lại một tiến trình này và bắt đầu một tiến trình khác
* Các Chức năng của Dispatcher thực hiện:

+ Chuyển đổi ngữ cảnh

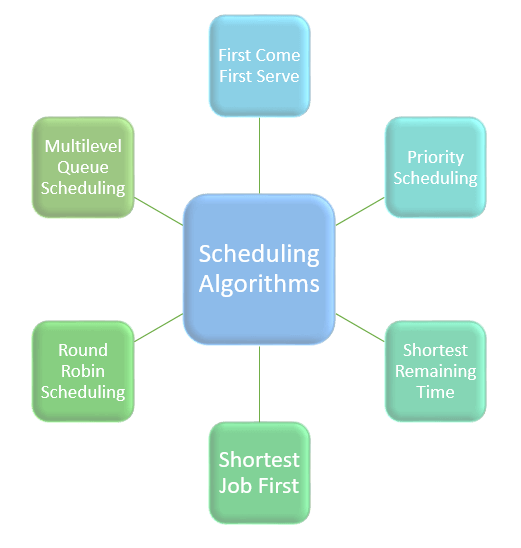
+ Chuyển đổi sang chế độ người dùng

+ Di chuyển đến vị trí chính xác trong chương trình mới được tải.

1. **Lập lịch CPU**

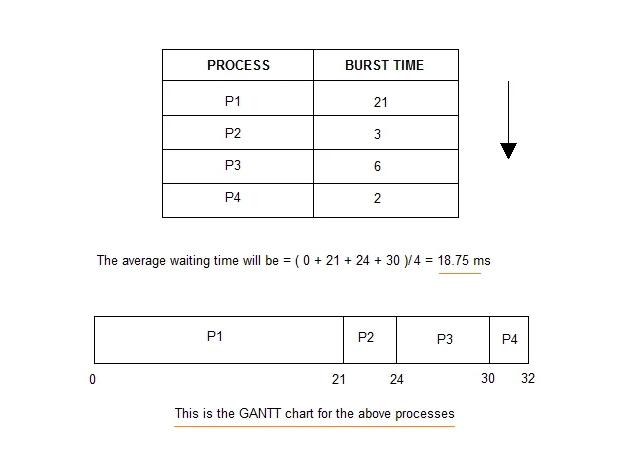
* **Mục đích:**
* Cải thiện hiệu quả của CPU
* Giúp nó phân bổ tài nguyên giữa các tiến trình cạnh tranh nhau
* Việc sử dụng CPU tối đa có thể đạt được với đa lập trình
* Các quy trình thực thi sẽ được nằm sẵn trong hàng đợi.
* **Các thuật toán lập lịch CPU**
* Có 6 loại thuật toán lập lịch phổ biến:

1. First Come First Serve (FCFS)
2. Shortest-Job-First (SJF) Scheduling
3. Shortest Remaining Time
4. Priority Scheduling
5. Round Robin Scheduling
6. Multilevel Queue Scheduling



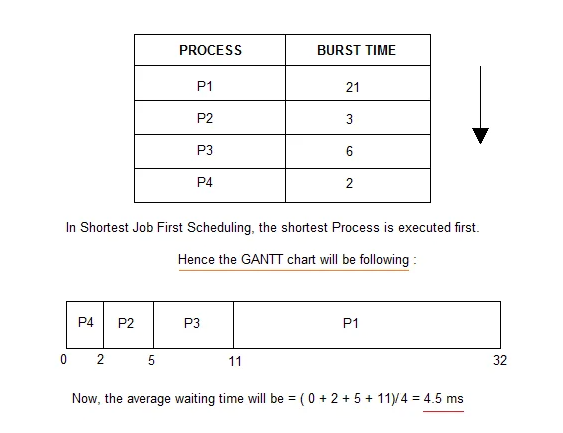
1. **First Come First Serve (FCFS)**
   1. Khái Niêm

* Thuật toán lập lịch “First come first serve”. Giống như tên thì thuật toán này yêu cầu: quá trình nào đến trước thì sẽ thực thi trước quá trình nào đến sau thì sẽ thực thi sau.
* First come first serve thì nó cũng giống như cấu trúc dữ liệu: Hàng đợi.
* Ví dự thực tế của thuật toán First come first serve đó là mua vé tại quầy bán vé.
  1. Các vấn đề về lập lịch First come first serve
* Thuật toán Non Pre-emptive hay thuật toán về mức độ ưu tiên sẽ không thể sử dụng, mức độ ưu tiên của quy trình không còn quan trọng.
* Chính vì thế hệ thống sẽ dễ bị sập, chỉ vì lập kế hoạch quy trình không đúng.
* Thời gian trung bình không tối ưu
* Việc sử dụng tài nguyên song song là không thể. Vì thế sẽ dẫn đến hiệu ứng “**Convoy**”. Do đó việc sử dụng tài nguyên (CPU, I/O,v.v) kém.
  1. Hiệu ứng Convoy là gì ???
* Hiệu ứng Convoy xảy ra khi có nhiều tiến trình diễn ra, và nó cần sử dụng tài nguyên trong một thời gian ngắn nhưng lại bị chặn bởi một tiến trình đang giữ tài nguyên đó trong một thời gian dài !
* Điều này dẫn đến việc sử dụng nhiều tài nguyên và hiệu suất kém
  1. Sơ đồ minh họa hoạt động:

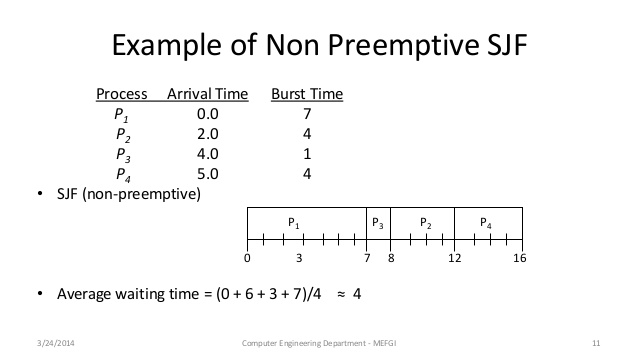


1. **Shortest-Job-First (SJF) Scheduling**
   1. Đặc điểm

* Nó hoạt đọng theo nguyên lý với thời gian hoặc khoảng thời gian ngắn nhất sẽ chạy trước tiên.
* Sẽ giảm thiểu tối đa thời gian chờ đợi.
* Sử dụng trong hệ thống hàn loạt
* Để thực hiện thành công thì buộc bộ sử lý phải biết trước được thời gian thực thi của các tiến trình. Chính vì thế điều này thực tế thì khá thiếu khả thi.
* Thuật toán lập lịch này là tối ưu nếu tất cả các tiến trình sẵn sàng thực thi cùng một lúc.
  1. Nguyên lý hoạt động



* 1. Vấn đề của Shortest-Job-First với Non Pre-emptive
* Khi thời gian sẵn sàng hoạt động của các tiến trình là khác nhau, hay các tiến trình nó không có sẵn tại hàng đợi tại thời điểm 0. Trong những tình huống như vậy, đôi khi các tiến trình mới có thời gian ngắn phải đợi quá trình thực thi kết thúc thì mới thực hiện được. Bởi vì trong SJF không ưu tiên khi xuất hiện một tiến trinhg mới,
* Chính vì vậy nó dẫn đến việc thiếu tài nguyên, và một tiến trình ngắn hơn phải đợi một tiến trình dài hơn thực hiện



1. **Shortest Remaining Time**
   1. Khái Niệm

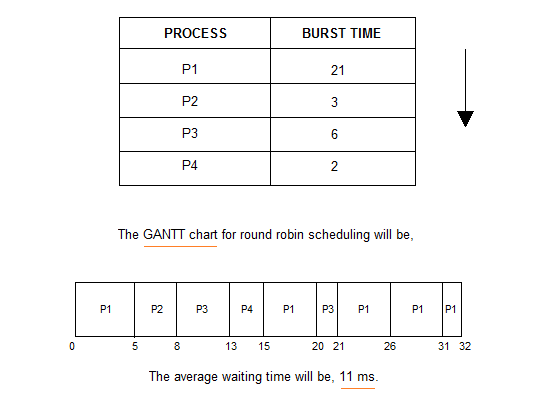
* Trong Phương pháp này các tiến trình sẽ được sắp xếp sao cho các tiến trình sắp hoàn thành sẽ được phân chạy trước và các tiến trình chưa chạy sẽ được phân chạy sau. Điều này sẽ ngăn các tiến trình mới thực thi và nó sẽ thực thi các tiến trình cũ hơn
  1. Đặc điểm
* Phương pháp này hầu heeys được áp dụng trong môi trường hoàng loạt, nơi ưu tiên các công việc ngắn hạn
* Đây thì không phải là một phương pháp lý tưởng để thực hiện nó trong một hệ thống chia sẻ, mà thời gian CPU cần thiết không xác định.
* Tuy nhiên thì đây là một thuật toán giúp lập lịch trình với thời gian ngắn nhất có thể.

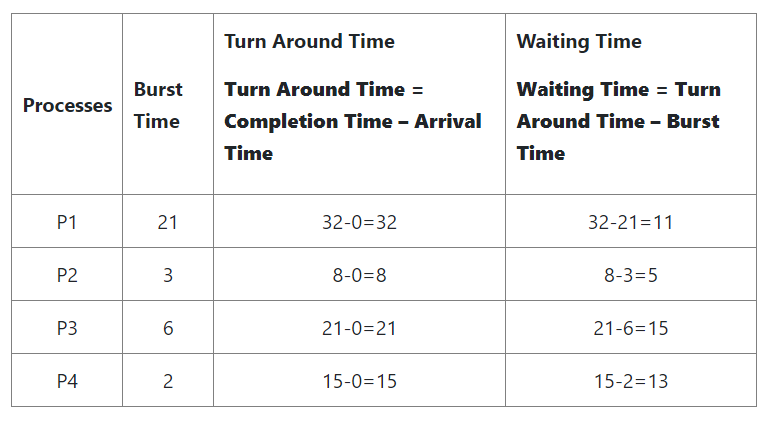
1. **Priority Scheduling**

* Đây là phương pháp lập lịch dự trên mức độ ưu tiên. Khi đó bộ lập lịch chọn các tác vụ, tiến trình để làm việc theo mức độ ưu tiên
* Vì vậy các công việc có độ ưu tiên cao hơn sẽ được thực hiện sớm hơn. Và các công việc có mức độ ưu tiên ngang nhau sẽ được thực hiện theo Round-Robin hoặc First Come First Serve.

1. **Round-Robin Scheduling.**
   1. Khái Niệm

* Thuật toán lập lịch Round Robin chủ yếu được thiết kế cho các hệ thống chia sẻ thời gian.
* Thuật toán này tương tự FCFS nhưng trong thuật toán lập lịch Round Robin quyền ưu tiên sẽ được thêm vào
* Thuật toán này sẽ khắc phục tình trạng Convoy ở thật toán FCFS. Vì tắt cả các tiến trình thì đều có một phần CPU công bằng.
  1. Nguyên lý hoạt động

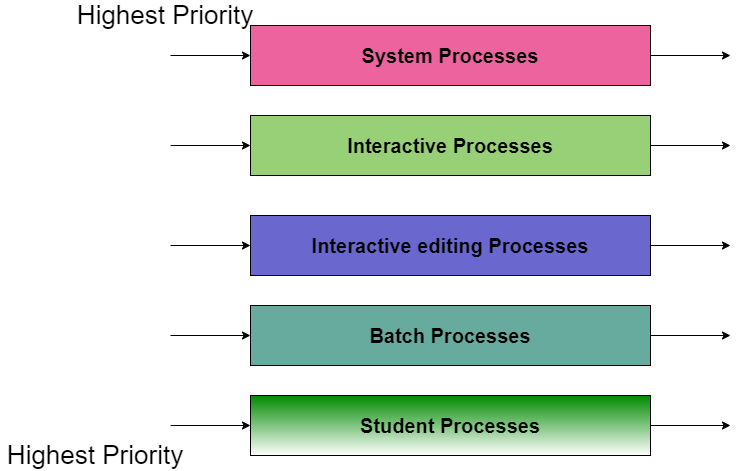




* 1. Một số ưu, nhực điểm:
* Ưu điểm:
* Về thời gian phản hồi trung bình bì thuật toán này cho hiệu suất tốt nhất
* Công việc sẽ được CPU phân bổ hợp lý
* Sử lý được tất cả các tiến trình mà không cần bất kì sự ưu tiên nào.
* Mỗi tiến trình sẽ có cơ hội lên lịch lại sau một thời gian cụ thể.
* Nhược điểm:
* Thuật toán dành nhiều thời gian cho việc chuyển đổi giữa các tiến trình và lập kế hoạch
* Cung cấp thời gian chờ và phản hồi lớn

1. **Multilevel Queue Scheduling**
   1. Khái niệm

* Thuật toán này, nó sẽ tách hàng đợi sẵn sàng thành nhiều hàng đợi riêng biệt khác nhau. Các quá trình gán cho một hàng đợi dựa trên một thuộc tính cụ thể của tiến trình, như : độ ưu tiên, kích thước bộ nhớ, vv
* Tuy nhiên đây không phải là một thuật toán lập lịch độc lập mà nó cần sử dụng các thuật toán khác để lập lịch công việc.
  1. Ví dụ mô tả:



Nó sẽ chia hàng đợi có sẵn thành các hàng đợi:

* Quy trình hệ thống (FCFS)
* Quy trình tương tác (SJF)
* Quy trình chỉnh sửa tương tác (SJF)
* Quy trình hàng loạt (RR)
* Quy trình học sinh (lên lịch ưu tiên)
  1. Ưu nhược điểm của tiến trình
* Ưu điểm:
* Ta có thể áp dụng nhiều loại lập lịch khác nhau
* Nhược điểm:
* Nó sẽ làm cho các tiến trình thấp cấp không được thực thi hoặc nếu có thì phải đợi trong một thời gian dài.