🙠🙟🕮🙝🙢

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐIỆN TỬ - VIỄN THÔNG**

---oOo---

Ảnh có chứa văn bản

Mô tả được tạo với mức tin cậy rất cao

**BÁO CÁO BÀI TẬP LỚN**

**Môn Hệ Điều Hành**

**Đề tài:**

GVHD: Phạm Doãn Tĩnh

Danh sách thành viên nhóm : 9

|  |  |  |
| --- | --- | --- |
| STT | Họ và Tên | MSSV |
| 1 | Lê Công Phương Nam | 20172706 |
| 2 | Trương Văn Minh | 20172695 |
| 3 | Đỗ Phương Thảo | 20172829 |
| 4 | Lê Quyết Anh |  |



Hà Nội, 11/2020

**Lời Nói Đầu**

**Chương 1. Tổng Quan**

1.1. Lý do chọn đề tài

Hệ điều hành là phần gắn bó trực tiếp với phần cứng và là môi trường để cho các chương trình ứng dụng khác chạy trên nó. Với chức năng quản lý và phân phối tài nguyên một cách hợp lý, đồng thời giả lập một máy tính mở rộng và tạo giao diện tiện lợi với người sử dụng, hệ điều hành là một thành phần then chốt không thể thiếu được trong mỗi một hệ thống máy tính điện tử.

Một trong những chức năng quan trọng của hệ điều hành là quản lý CPU. Trong môi trường xử lý đa chương, có thể xảy ra tình huống nhiều tiến trình đồng thời sẵn sàng để xử lý. Mục tiêu của các hệ phân chia thời gian (time-sharing) là chuyển đổi CPU qua lại giữa các tiến trình một cách thường xuyên để nhiều người sử dụng có thể tương tác cùng lúc với từng chương trình trong quá trình xử lý.

Để thực hiện được mục tiêu này, hệ điều hành phải lựa chọn tiến trình được xử lý tiếp theo. Bộ điều phối sẽ sử dụng một giải thuật điều phối thích hợp để thực hiện nhiệm vụ này. Một thành phần khác của hệ điều hành cũng tiềm ẩn trong công tác phối là bộ điều phối (dispatcher). Bộ phân phối sẽ chịu trách nhiệm chuyển đổi ngữ cảnh và trao CPU cho tiến trình được chọn bởi bộ điều phối để xử lý.

Vì những lợi ích lớn lao mà giải thuật điều phối CPU đem lại và để tìm hiểu kĩ hơn về nguyên tắc hoạt động của chúng, chúng em quyết định chọn đề tài: “”.

1.2. Mục tiêu của đề tài

* Tìm hiểu các giải thuật: First In First Out (FIFO), Round Robin (RR), Shortest Job First (SJF), Shortest Remain Time (SRT).
* Chỉ ra được ưu và nhược điểm của các giải thuật lập lịch cho CPU.
* Xây dựng chương trình mô phỏng các giải thuật đã tìm hiểu và kết quả demo.

**Chương 2. Cơ Sở Lý Thuyết**

2.1. Các khái niệm cơ bản

2.1.1. Tiến trình - Process là gì?

Tiến trình - Process là một khái niệm cơ bản trong bất kì một hệ điều hành nào. Một tiến trình có thể được định nghĩa là một thực thể chương trình đang được chạy trong hệ thống. Nó là một tập hợp các cấu trúc dữ liệu mô tả đầy đủ quá trình một chương trình được chạy trong hệ thống.

Một tiến trình cũng trải qua các quá trình như con người: Nó được sinh ra, nó có thể có một cuộc đời ít ý nghĩa hoặc nhiều ý nghĩa, nó có thể sinh ra một hoặc nhiều tiến trình con, và thậm chí, nó có có thể chết đi. Điều khác biệt nhỏ duy nhất là: Tiến trình không có giới tính. Mỗi tiến trình chỉ có một tiến trình cha (Hoặc có thể gọi là mẹ) duy nhất.

Dưới góc nhìn của kernel, tiến trình là một thực thể chiếm dụng tài nguyên của hệ thống (Thời gian sử dụng CPU, bộ nhớ…).

Khi một tiến trình con được tạo ra, nó hầu như giống hệt như tiến trình cha. Tiến trình con sao chép toàn bộ không gian địa chỉ, thực thi cùng một mã nguồn như tiến trình cha, và bắt đầu chạy tiếp những mã nguồn riêng cho tiến trình con từ thời điểm gọi hàm tạo tiến trình mới.

Mặc dù tiến trình cha và tiến trình con cùng chia sẻ sử dụng phần mã nguồn của chương trình, nhưng chúng lại có phần dữ liệu tách biệt nhau (stack và heap). Điều này có nghĩa là, những sự thay đổi dữ liệu của tiến trình con không ảnh hưởng đến dữ liệu trong tiến trình cha.

Một web server chạy trong thiết bị là một tiến trình, một chương trình soạn thảo văn bản đang chạy trong thiết bị cũng là một tiến trình. Cùng là một chương trình soạn thảo (Ví dụ chương trình quen thuộc Word), người dùng mở 16 chương trình Word để thao tác với 16 file khác nhau, chúng ta cho 16 tiến trình chạy trong hệ thống.

2.1.2. Các đặc điểm của tiến trình - Process

Điều phối hoạt động của các tiến trình là một vấn đề rất phức tạp, đòi hỏi hệ điều hành khi giải quyết phải xem xét nhiều yếu tố khác nhau để có thể đạt được những mục tiêu đề ra. Một số đặc tính của tiến trình cần được quan tâm như tiêu chuẩn điều phối:

* Tính hướng xuất/nhập của tiến trình: CPU chủ yếu để phát sinh nhập xuất, hoạt động của tiến trình bao gồm nhiều lượt sử dụng CPU, mỗi lượt trong một khoảng thời gian khá ngắn
* Tính hướng xử lí của tiến trình: Tiến trình có xu hướng dùng hết thời gian CPU dành cho nó, hoạt động như thế thường bao gồm một lượt ít sử dụng CPU nhưng một lượt trong một khoảng thời gian đủ dài.
* Tiến trình tương tác hay xử lí theo lô: Người sử dụng theo kiểu tương tác thường yêu cầu được hồi đáp tức thời đối với các yêu cầu của họ, trong khi các tiến trình của các tác vụ được xử lý theo lô nói chung có thể trì hoãn trong một thời gian chấp nhận được.
* Độ ưu tiên của tiến trình: Các tiến trình được phân các theo một số tiêu chí đánh giá nào đó, một cách hợp lí. Các tiến trình quan trọng hơn (Có độ ưu tiên cao hơn) cần được ưu tiên cao hơn.
* Thời gian đã sử dụng CPU của tiến trình: Một số quan điểm ưu tiên chọn những tiến trình đã sử dụng CPU nhiều thời gian nhất vì hy vọng chúng sẽ cần ít thời gian nhất để hoàn tất và rời khỏi hệ thống. Tuy nhiên cũng có quan điểm cho rằng các tiến trình nhận được CPU trong ít thời gian là những tiến trình đã phải chờ lâu nhất, do vậy ưu tiên chọn chúng.
* Thời gian còn lại để tiến trình hoàn tất: Có thể giảm thiểu thời gian chờ trung bình của các tiến trình bằng cách cho các tiền trình cần ít thời gian nhất để hoàn tất được thực hiên trước. Tuy nhiên rất hiếm khi biết cần bao nhiêu thời gian nữa để kết thúc tiến trình.

2.1.3. Lập lịch để làm gì?

Bộ điều phối không cung cấp cơ chế, mà đưa ra các quyết định. Các hệ điều hành xây dựng nhiều chiến lược khác nhau để thực hiện việc điều phối, nhưng tựu chung cần đạt được các mục tiêu sau:

* Sự công bằng (Fairness): Các tiến trình chia sẻ CPU một cách công bằng không có tiến trình nào phải đợi vô hạn để được cấp phát CPU.
* Tính hiệu quả (Efficiency): Hệ thống phải tận dụng được CPU 100% thời gian.
* Thời gian đáp ứng hợp lý (Response time): Cực tiểu hóa thời gian hồi đáp cho các tương tác của người sử dụng.
* Thời gian lưu lại trong hệ thống (Turnaround Time): Cực tiểu hóa thời gian hoàn tất các tác vụ xử lý theo lô.
* Thông lượng tối đa (Throughput): Cực đại hóa số công việc được xử lý trong một đơn vị thời gian.

Tuy nhiên thường không thể thỏa mãn tất cả các mục tiêu kể trên vì bản thân chúng có sự mâu thuẫn với nhau mà chỉ có thể thể dung hòa chúng ở mức độ nào đó.

2.1.4. Điều phối độc quyền và điều phối không độc quyền

Thuật toán điều phối cần xem xét và quyết định thời điểm chuyển đổi CPU giữa các tiến trình. Hệ điều hành có thể thực hiện cơ chế điều phối theo nguyên lý độc quyền hoặc không độc quyền.

Điều phối độc quyền: Nguyên lý điều phối độc quyền cho phép một tiến trình khi nhận được CPU sẽ có quyền độc chiếm CPU đến khi hoàn tất xử lý hoặc tự nguyện giải phóng CPU. Khi đó quyết định điều phối CPU sẽ xảy ra trong các tình huống sau:

* Khi tiến trình chuyển từ trạng thái đang xử lý (Running) sang trạng thái bị khóa Blocked (Ví dụ chờ một thao tác nhập xuất hay chờ một tiến trình con kết thúc…).
* Khi tiến trình kết thúc.

Các giải thuật độc quyền thường đơn giản và dễ cài đặt. Tuy nhiên chúng thường không thích hợp với các hệ thống tổng quát nhiều người dùng, vì nếu cho phép một tiến trình có quyền xử lý bao lâu tùy ý, có nghĩa là tiến trình này có thể giữ CPU một thời gian không xác định, có thể ngăn cản những tiến trình còn lại trong hệ thống có một cơ hội để xử lý.

Điều phối không độc quyền: Ngược với nguyên lý độc quyền, điều phối theo nguyên lý không độc quyền cho phép tạm dừng hoạt động của một tiến trình đang sẵn sàng xử lý. Khi một tiến trình nhận được CPU, nó vẫn được sử dụng CPU đến khi hoàn tất hoặc tự nguyện giải phóng CPU, nhưng một tiến trình khác có độ ưu tiên có thể dành quyền sử dụng CPU của tiến trình ban đầu. Như vậy là tiến trình có thể bị tạm dừng hoạt động bất cứ lúc nào mà không được báo trước, để tiến trình khác xử lý. Các quyết định điều phối xảy ra khi :

* Khi tiến trình chuyển từ trạng thái đang xử lý (Running) sang trạng thái bị khóa Blocked (Ví dụ chờ một thao tác nhập xuất hay chờ một tiến trình con kết thúc…).
* Khi tiến trình chuyển từ trạng thái đang xử lý (Running) sang trạng thái Ready (Ví dụ xảy ra một ngắt).
* Khi tiến trình chuyển từ trạng thái chờ (Blocked) sang trạng thái Ready (Ví dụ một thao tác nhập/xuất hoàn tất).
* Khi tiến trình kết thúc.

Các thuật toán điều phối theo nguyên tắc không độc quyền ngăn cản được tình trạng một tiến trình độc chiếm CPU, nhưng việc tạm dừng một tiến trình có thể dẫn đến các mâu thuẫn trong truy xuất, đòi hỏi phải sử dụng một phương pháp đồng bộ hóa thích hợp để giải quyết.

Trong các hệ thống sử dụng nguyên lý điều phối độc quyền có thể xảy ra tình trạng các tác vụ cần thời gian xử lý ngắn phải chờ tác vụ xử lý với thời gian rất dài hoàn tất! Nguyên lý điều phối độc quyền thường chỉ thích hợp với các hệ xử lý theo lô.

Đối với các hệ thống tương tác (Time sharing), các hệ thời gian thực (Real time), cần phải sử dụng nguyên lý điều phối không độc quyền để các tiến trình quan trọng có cơ hội hồi đáp kịp thời. Tuy nhiên thực hiện điều phối theo nguyên lý không độc quyền đòi hỏi những cơ chế phức tạp trong việc phân định độ ưu tiên, và phát sinh thêm chi phí khi chuyển đổi CPU qua lại giữa các tiến trình.

2.1.5. Giờ CPU & các trạng thái của tiến trình liên quan đến giờ CPU

CPU là một loại tài nguyên quan trọng của máy tính. Mọi tiến trình muốn hoạt động được đều phải có sự phục vụ của CPU (Để xử lý, tính toán...). Thời gian mà CPU phục vụ cho tiến trình hoạt động được gọi là giờ CPU.

Tại mỗi thời điểm, chỉ có một tiến trình duy nhất được phân phối giờ CPU để hoạt động (Thực hiện các lệnh của mình).

Trong chế độ đa chương trình, có ba trạng thái của tiến trình liên quan mật thiết đến giờ CPU bao gồm:

Ready

New

Running

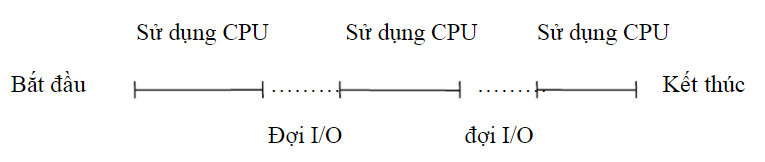
Halt

Waiting

Hình 2.1: Các trạng thái của tiến trình

* Sẵn sàng (Ready): Là trạng thái mà tiến trình được phân phối đầy đủ mọi tài nguyên cần thiết và đang chờ giờ CPU.
* Thực hiện (Running): Là trạng thái mà tiến trình được phân phối đầy đủ mọi tài nguyên cần thiết và giờ CPU.
* Đợi (Waiting): Là trạng thái mà tiến trình không được thực hiện được vì thiếu một vài điều kiện nào đó (Đợi dữ liệu vào/ra, đợi tài nguyên bổ sung...). Khi sự kiện mà nó chờ đợi xuất hiện, tiến trình sẽ quay lại trạng thái sẵn sàng.

Như vậy, trong suốt thời gian tồn tại của mình, các tiến trình sẽ tuân thủ theo sơ đồ thực hiện sau:



Hình 2.2: Sơ đồ thực hiện các tiến trình

Một tiến trình đang trong trạng thái thực hiện, nó có thể rời khỏi trạng thái bởi một trong ba lý do:

* Tiến trình đã hoàn thành công việc, khi đó nó trả lại giờ CPU và chuyển sang chờ xử lý kết thúc.
* Tiến trình tự ngắt: Khi tiến trình chờ đợi một sự kiện nào đó, tiến trình sẽ được chuyển sang trạng thái thực hiện khi có xuất hiện sự kiện nó đang chờ.
* Tiến trình sử dụng hết giờ CPU dành cho nó, khi đó nó sẽ được chuyển sang trạng thái sẵn sàng.

Việc chuyển tiến trình sang trạng thái sẵn sàng về bản chất là thực hiện vệc phân phối lại giờ CPU.

2.1.6. Khái niệm lập lịch cho CPU

Để điều khiển tiến trình ở nhiều trạng thái khác nhau, hệ thống thường tổ chức các từ trạng thái (Thực chất là các khối điều khiến tiến trình) để ghi nhận tình trạng sử dụng tài nguyên và trạng thái của tiến trình. Các trạng thái được tổ chức theo kiểu hàng đợi như sau:

Hình 2.3: Cách tổ chức các trạng thái

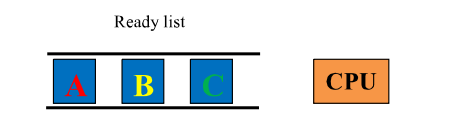
Như vậy lập lịch cho CPU có nghĩa là tổ chức một hàng đợi các tiến trình sẵn sàng để phân phối giờ CPU cho chúng dựa trên độ ưu tiên của các tiến trình, sao cho hiệu suất sử dụng CPU là tối ưu nhất.

Mỗi tiến trình ở trạng thái sẵn sàng sẽ được gắn với một thứ tự ưu tiên. Thứ tự ưu tiên này được xác định dựa vào các yếu tố như: Thời điểm hình thành tiến trình, thời gian thực hiện tiến trình, thời gian kết thúc tiến trình…

2.2. Các thuật toán lập lịch

2.2.1. First Come First Served (FCFS)

Trong thuật toán này, độ ưu tiên phục vụ tiến trình căn cứ vào thời điểm hình thành tiến trình. Hàng đợi các tiến trình được tổ chức theo kiểu FIFO (First In First Out – Vào trước ra trước, vào sau ra sau). Mọi tiến trình đều được phục vụ theo trình tự xuất hiện cho đến khi kết thúc hoặc bị ngắt.



Hình 2.4: Điều phối FCFS

Ưu điểm của thuật toán này là giờ CPU không bị phân phối lại (Không bị ngắt) và chi phí thực hiện thấp nhất (Vì không phải thay đổi thứ tự ưu tiên phục vụ, thứ tự ưu tiên là thứ tự của tiến trình trong hàng đợi).

Nhược điểm của thuật toán là thời gian trung bình chờ phục vụ của các tiến trình là như nhau (Không kể tiến trình ngắn hay dài), do đó dẫn tới ba điểm sau:

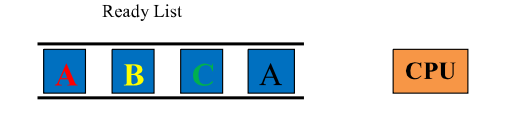
* Thời gian chờ trung bình sẽ tăng vô hạn khi hệ thống tiếp cận tới hạn khả năng phục vụ của mình.
* Nếu độ phát tán thời gian thực hiện tiến trình tăng thì thời gian chờ đợi trung bình cũng tăng theo.
* Khi có tiến trình dài, ít bị ngắt thì các tiến trình khác phải chờ đợi lâu hơn.

2.2.2. Round Robin (RR)

Giải thuật định thời luân phiên (Round - Robin - RR) được thiết kế đặc biệt cho hệ thống chia sẻ thời gian. Tương tự như định thời FIFO 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 gửi 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 cuối của hàng đợi sẵn sàng. Sau đó, bộ định thời biểu CPU sẽ chọn quá trình tiếp theo trong hàng đợi sẵn sàng.



Hình 2.5: Điều phối RR

Ưu điểm:

* Các quá trình sẽ được luân phiên cho CPU xử lý nên thời gian chờ đợi sẽ ít.
* Đối với các quá trình liên quan đến nhập/xuất người dùng thì rất hiệu quả.
* Việc cài đặt không quá phức tạp.

Nhược điểm:

* Thời gian chờ đợi trung bình của thuật toán RR thường là quá dài.
* Nếu thời gian định mức cho việc xử lý quá lớn thì RR thành FIFO.
* Nếu thời gian quá ngắn so với thời gian xử lý của một tiến trình trong danh sách hàng đợi thì việc chờ đợi và xử lý luân phiên sẽ nhiều.
* Qui tắc là định mức thời gian nên dài hơn 80% chu kỳ CPU.

2.2.3. Shortest Job First (SJF)

Một tiếp cận khác đối với việc định thời CPU là giải thuật định thời công việc ngắn nhất trước (Shortest-Job-First - SJF). 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 FIFO 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.

Ưu điểm:

* Giải thuật được xem là tối ưu, thời gian chờ đợi trung bình giảm.
* Tận dụng hết năng lực của CPU.

Nhược điểm:

* Cài đặt thuật toán phức tạp,tốn nhiều xử lý cho quá trình quản lý.
* Mặc dù SJF là tối ưu nhưng nó không thể được cài đặt tại cấp định thời CPU ngắn vì không có cách nào để biết chiều dài chu kỳ CPU tiếp theo.
* Giải thuật SJF có thể trưng dụng hoặc không trưng dụng CPU, dẫn tới giải thuật này có nhiều dị bản khác nhau và sẽ tối ưu hay không tối ưu phụ thuộc vào trưng dụng CPU.

2.2.4. Shortest Remain Time (SRT)

Tương tự như SJF nhưng trong thuật toán này, độ ưu tiên thực hiện các tiến trình dựa vào thời gian cần thiết để thực hiện nốt tiến trình (Bằng tổng thời gian trừ đi thời gian đã thực hiện). Như vậy, trong thuật toán này cần phải thường xuyên cập nhật CPU cũng phải được áp dụng nếu không sẽ làm mất tình ưu việc của thuật toán.

Ưu điểm:

* Thời gian chờ đợi, tồn tại trong hệ thống của mỗi tiến trình đều ngắn.
* Thuật toán tối ưu nhất.

Nhược điểm:

* Việc cài đặt thuật toán khá phức tạp.
* Cần quản lý chặt chẽ việc điều phối các tiến trình.
* Quản lý thời gian đến của mỗi tiến trình.