

BÀI TẬP CHƯƠNG 5 – ĐỊNH THỜI CPU

5.1 Thuật toán lập lịch CPU xác định thứ tự thực hiện các tiến trình đã sẵn sàng. Cho n tiến trình được lên lịch trên một vi xử lý, có bao nhiêu lịch trình khác nhau có thể sắp? Đưa ra một công thức chung cho n .

5.2 Giải thích sự khác biệt giữa lập lịch giữ quyền ưu tiên (preemptive) và lập lịch không giữ quyền ưu tiên (non-preemptive).

5.3 Giả sử rằng các tiến trình sau đây đến để thực thi tại thời điểm chỉ ra ở cột Arrival Time. Mỗi tiến trình sẽ chạy trong khoảng thời gian được liệt kê ở cột Burst Time. Sử dụng lập lịch không giữ quyền ưu tiên, và kết luận dựa trên thông tin tại thời điểm ra kết luận đó.

Process	Arrival Time	Burst Time
P1	0.0	8
P2	0.4	4
P3	1.0	1

Bảng 1. Dữ liệu bài tập 5.3

a. Thời gian quay vòng trung bình (average turnaround time) của các tiến trình này là bao nhiêu với thuật toán lập lịch FCFS?

b. Thời gian quay vòng trung bình của các tiến trình này là bao nhiêu với thuật toán lập lịch SJF?

c. Thuật toán SJF được cho là có cải thiện hiệu suất, nhưng chú ý rằng chúng ta đã chọn chạy tiến trình P1 tại thời điểm 0 vì chúng ta không biết rằng hai tiến trình ngắn hơn sẽ sớm đến hàng đợi lập lịch. Tính thời gian quay vòng trung bình sẽ là bao nhiêu nếu CPU không hoạt động trong 1.0 đơn vị thời gian đầu tiên và sau đó giải thuật lập lịch SJF được sử dụng. Hãy nhớ rằng quá trình P1 và P2 đang chờ trong thời gian nhàn rỗi này, vì vậy thời gian chờ của chúng có thể gia tăng. Thuật toán này có thể được gọi là lập lịch với thông tin nhìn trước (future-knowledge scheduling).

5.4 Xem tập các tiến trình sau đây, với thời gian cần chạy ở cột Burst Time được cho ở đơn vị mili giây.

Process	Burst Time	Priority
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

Bảng 2. Dữ liệu bài tập 5.4

Cho rằng các tiến trình đến theo thứ tự P1, P2, P3, P4, P5, tại thời điểm 0.0

- Vẽ bốn biểu đồ Gantt minh họa việc thực thi các tiến trình này lần lượt sử dụng các thuật toán lập lịch sau: FCFS, SJF, Mức độ ưu tiên không giữ quyền ưu tiên (non-preemptive priority) (số ưu tiên lớn hơn hàm ý độ ưu tiên cao hơn) và RR (một lượng thời gian = 2).
- Thời gian quay vòng của mỗi quá trình cho mỗi thuật toán lập lịch trình tại phần a là bao nhiêu?
- Thời gian chờ đợi của mỗi tiến trình trong từng giải thuật tại phần a là bao nhiêu?
- Những thuật toán nào dẫn đến thời gian đợi trung bình nhỏ nhất (tính trên tất cả các tiến trình)?

5.5 Các tiến trình sau đây được lập lịch với giải thuật RR

Process	Priority	Burst Time	Arrival
P1	40	20	0
P2	30	25	25
P3	30	25	30
P4	35	15	60
P5	5	10	100
P6	10	10	105

Bảng 3. Dữ liệu bài tập 5.5

Mỗi tiến trình được gán mức ưu tiên bằng một con số, giá trị cao hơn thể hiện quyền ưu tiên cao hơn. Ngoài các tiến trình được liệt kê dưới đây, hệ thống cũng có một tác vụ nhàn rỗi (nó không tiêu tốn tài nguyên CPU và được đặt tên là Pidle). Nhiệm vụ này có mức độ ưu tiên thấp nhất và được lên lịch khi hệ thống không có tiến trình sẵn sàng để chạy. Độ dài của một lượng thời gian là 10 đơn vị. Nếu một tiến trình bị chiếm quyền ưu tiên bởi một tiến trình có độ ưu tiên cao hơn, tiến trình nạn nhân sẽ được đặt ở cuối hàng đợi.

- Hiện thị thứ tự lập lịch của các tiến trình bằng biểu đồ Gantt.
- Thời gian quay vòng cho mỗi quá trình là bao nhiêu?
- Thời gian chờ đợi cho mỗi tiến trình là bao nhiêu?
- Tỷ lệ sử dụng CPU là bao nhiêu?

5.6 Lợi điểm có thể đạt được là gì khi hệ thống hàng đợi đa mức sử dụng các lượng thời gian (cho một lần mà một tiến trình sử dụng CPU) khác nhau tại mỗi mức?

5.7 Nhiều thuật toán lập lịch CPU được tham số hóa. Ví dụ: Thuật toán RR yêu cầu một tham số để chỉ ra thời lượng một lần chạy. Hàng đợi đa cấp có phản hồi yêu cầu tham số để chỉ ra số lượng hàng đợi, các thuật toán lập lịch cho mỗi hàng đợi, các tiêu chí được sử dụng để di chuyển

các tiến trình giữa các hàng đợi, v.v.

Do đó, các thuật toán này thực sự là các bộ thuật toán (ví dụ: bộ thuật toán RR cho tất cả các lát cắt thời gian, v.v.). Một bộ thuật toán có thể bao gồm một cái khác (ví dụ: thuật toán RR khi sử dụng giá trị rất lớn cho lượng thời gian thì nó tương tự như giải thuật FCFS). Chỉ ra mối quan hệ (nếu có) giữa các cặp thuật toán sau đây?

- Ưu tiên và SJF

- b. Hàng đợi phản hồi đa cấp và FCFS
- c. Ưu tiên và FCFS
- d. RR và SJF

5.8 Giả sử rằng một thuật toán lập lịch CPU thiên vị các tiến trình đã sử dụng ít thời gian xử lý nhất trong lần chạy gần đây. Tại sao điều này sẽ dẫn đến sự thiên vị các tiến trình có xu hướng nhập / xuất và cũng không gây ra sự thiếu thốn vĩnh viễn cho các tiến trình có xu hướng tính toán CPU.

5.9 Phân biệt giữa lập lịch PCS và SCS.

5.10 Bộ lập lịch UNIX truyền thống sử dụng mối quan hệ nghịch đảo giữa số ưu tiên và độ ưu tiên: số càng cao, sự ưu tiên càng thấp¹. Bộ lập lịch tính toán lại các ưu tiên của quy trình một lần mỗi giây

sử dụng chức năng sau:

$$\text{Độ Ưu tiên} = (\text{mức sử dụng CPU gần đây} / 2) + \text{cơ sở}$$

Trong đó cơ sở = 60 và mức sử dụng CPU gần đây là một giá trị cho biết cách mà một tiến trình đã sử dụng CPU từ lúc độ ưu tiên được tính toán lần cuối gần đó nhất.

Giả sử rằng mức sử dụng CPU gần đây cho tiến trình P1 là 40, cho tiến trình P2 là 18, và cho tiến trình P3 là 10. Ưu tiên mới cho 3 tiến trình này sẽ là bao nhiêu khi được tính toán lại? Dựa trên thông tin này, bộ lập lịch UNIX truyền thống tăng hay giảm độ ưu tiên tương đối với một tiến trình có xu hướng tính toán CPU.

5.11 Trong hai loại chương trình này:

- a. Xu hướng I / O
- b. Xu hướng tính toán CPU

Thì nhóm nào có nhiều khả năng có các chuyển ngữ cảnh tự nguyện (non-preemptive), và cái nào có nhiều khả năng có các chuyển ngữ cảnh không tự nguyện (preemptive)? Giải thích.

5.12 Thảo luận về các cặp tiêu chí lập lịch sau đây với cùng các thiết lập khác.

- a. Mức độ sử dụng CPU và Thời gian đáp ứng
- b. Thời gian quay vòng trung bình và thời gian chờ tối đa
- c. Mức độ sử dụng thiết bị I / O và mức độ sử dụng CPU

5.13 Một giải thuật để thực hiện việc **lập lịch xổ số** là ấn định các vé xổ số cho các tiến trình

xử lý, con số mà được sử dụng để phân bổ thời gian CPU. Khi nào giải thuật cần quyết định, một vé xổ số được chọn ngẫu nhiên, và tiến trình đang giữ vé đó sẽ được dùng CPU. Hệ điều hành BTV hiện thực lập lịch xổ số bằng cách tổ chức xổ số 50 lần mỗi giây, với mỗi tiến trình trúng xổ số nhận được 20 mili giây sử dụng CPU (20 mili giây \times 50 = 1 giây). Hãy giải thích vì sao bộ lập lịch BTV có thể đảm bảo rằng các tiến trình có độ ưu tiên cao hơn nhận được nhiều sự chú ý hơn từ CPU hơn các tiến trình có độ ưu tiên thấp hơn.

¹ Windows cũng vậy, độ ưu tiên = 0 là ưu tiên cao nhất.

5.14 Hầu hết các thuật toán lập lịch duy trì một hàng đợi “run”, liệt kê các tiến trình đủ điều kiện để chạy trên một bộ xử lý. Trên các hệ thống đa nhân, có hai tùy chọn: (1) mỗi nhân xử lý có hàng đợi chạy riêng hoặc (2) một hàng đợi chạy được chia sẻ bởi tất cả các nhân xử lý. Những ưu điểm và nhược điểm của từng phương pháp tiếp cận này là gì?

5.15 Xem xét công thức trung bình theo cấp số nhân được sử dụng để dự đoán độ dài của lượt sử dụng CPU tiếp theo. Ý nghĩa của việc gán các giá trị cho các tham số được sử dụng bởi thuật toán là gì?

- a. $\alpha = 0$ và $\tau_0 = 100$ mili giây
- b. $\alpha = 0,99$ và $\tau_0 = 10$ mili giây

5.16 Một biến thể của lập lịch RR là RR hồi quy. Bộ lập lịch này gán cho mỗi tiến trình một lượng thời gian và một độ ưu tiên. Giá trị ban đầu của một lượng thời gian là 50 mili giây. Tuy nhiên, mỗi khi một tiến trình được phân bổ CPU và sử dụng toàn bộ thời gian của nó lượng tử (không chặn I / O), 10 mili giây được thêm vào thời gian của nó lượng tử, và mức độ ưu tiên của nó được tăng cường. (Lượng tử thời gian cho một quá trình có thể tăng lên tối đa 100 mili giây.) Khi một khối xử lý trước khi sử dụng toàn bộ lượng tử thời gian của nó, lượng tử thời gian của nó là giảm 5 mili giây, nhưng mức độ ưu tiên của nó vẫn giữ nguyên. Loại của quá trình (ràng buộc CPU hoặc ràng buộc I / O) thực hiện vòng tròn hồi quy lập lịch ủng hộ? Giải thích.

5.17 Xem tập các tiến trình sau đây, với thời gian cần chạy ở cột Burst Time được cho ở đơn vị mili giây.

Process	Burst Time	Priority
P1	5	4
P2	3	1
P3	1	2
P4	7	2
P5	4	3

Bảng 4. Dữ liệu bài tập 5.17

Cho rằng các tiến trình đến theo thứ tự P1, P2, P3, P4, P5, tại thời điểm 0.0

- a. Vẽ bốn biểu đồ Gantt minh họa việc thực thi các tiến trình này lần lượt sử dụng các thuật toán lập lịch sau: FCFS, SJF, Mức độ ưu tiên không giữ quyền ưu tiên (non-preemptive priority) (số ưu tiên lớn hơn hàm ý độ ưu tiên cao hơn) và RR (một lượng thời gian = 2).
- b. Thời gian quay vòng của mỗi quá trình cho mỗi thuật toán lập lịch trình trong phần a là bao nhiêu?
- c. Thời gian chờ đợi của mỗi quá trình cho mỗi tiến trình trong từng giải thuật là bao nhiêu?
- d. Những thuật toán nào dẫn đến thời gian đợi trung bình nhỏ nhất (tính trên tất cả các tiến trình)?

5.18 Các tiến trình sau đây được lập lịch với giải thuật RR, có xét đến độ ưu tiên của tiến trình.

Process	Priority	Burst Time	Arrival
P1	8	15	0
P2	3	20	0
P3	4	20	20
P4	4	20	25
P5	5	5	45
P6	5	15	55

Bảng 5. Dữ liệu bài tập 5.18

Mỗi tiến trình được chỉ định độ ưu tiên bằng số, với giá trị càng lớn thì độ ưu tiên càng cao. Bộ lập lịch sẽ thực hiện tiến trình ưu tiên cao nhất. Đối với các tiến trình có cùng độ ưu tiên, bộ lập lịch RR sẽ được sử dụng với lượng thời gian là 10 đơn vị. Nếu một tiến trình bị chiếm quyền ưu tiên bởi một tiến trình ưu tiên cao hơn, tiến trình nạn nhân được đặt ở cuối hàng đợi.

- Hiển thị thứ tự lập lịch của các tiến trình bằng biểu đồ Gantt.
- Thời gian quay vòng cho mỗi tiến trình là bao nhiêu?
- Thời gian chờ đợi cho mỗi tiến trình là bao nhiêu?

5.19 Lệnh nice được sử dụng để đặt giá trị đẹp của tiến trình trên Linux, cũng như trên các hệ thống UNIX khác. Giải thích tại sao một số hệ thống cho phép bất kỳ người dùng nào gán cho một tiến trình giá trị đẹp ≥ 0 nhưng chỉ cho phép root (hoặc quản trị viên) gán các giá trị đẹp < 0 .

5.20 Thuật toán lập lịch nào sau đây có thể dẫn đến một vài tiến trình bị thiếu thôn tài nguyên?

- First-come, first-served
- Shortest job first
- Round robin
- Priority

5.21 Xem xét một biến thể của thuật toán lập lịch RR trong đó thực thể trong hàng đợi đã sẵn sàng là con trỏ tới PCB.

- Điều gì sẽ có tác dụng của việc đặt hai con trỏ cùng chỉ đến một tiến trình trong hàng đợi sẵn sàng?
- Chỉ ra 2 ưu điểm và 2 khuyết điểm của cách tiếp cận này.
- Làm thế nào để sửa đổi thuật toán RR cơ bản để đạt được ưu điểm tương tự mà không có con trỏ trùng lặp?

5.22 Hãy xem xét một hệ thống chạy mười tác vụ có xu hướng nhập / xuất và một tác vụ có xu hướng tính toán CPU. Giả sử rằng các tác vụ có xu hướng nhập / xuất thực hiện thao tác nhập / xuất một lần mỗi 1 mili giây và mỗi thao tác nhập / xuất mất 10 mili giây để hoàn thành. Cũng giả sử rằng chuyển ngữ cảnh có tổng chi phí là 0,1 mili giây và tất cả các tiến trình đều là các tác vụ dài hạn.

Mô tả việc sử dụng CPU cho bộ lập lịch RR khi:

a. Time slice / Lượng thời gian là 1 mili giây.

b. Time slice / Lượng thời gian là 10 mili giây.

5.23 Xem xét một hệ thống hiện thực lập lịch với hàng đợi đa mức. Người dùng máy tính có thể sử dụng chiến lược gì để tối đa hóa thời lượng CPU trong một lần phân bổ cho tiến trình người dùng?

5.24 Xem xét thuật toán lập lịch ưu tiên có giữ quyền ưu tiên dựa trên độ ưu tiên biến động. Số ưu tiên lớn hơn ngụ ý độ ưu tiên cao hơn. Khi một quá trình đang chờ CPU (trong hàng đợi sẵn sàng, nhưng không chạy), độ ưu tiên của nó thay đổi theo tỷ lệ α . Khi nó đang chạy, độ ưu tiên của nó thay đổi với tỷ lệ β . Tất cả các quy trình được gán độ ưu tiên là 0 khi chúng vào hàng đợi sẵn sàng. Các tham số α và β có thể được thiết lập để tạo ra nhiều thuật toán lập lịch khác nhau.

a. Thuật toán kết quả từ $\beta > \alpha > 0$ là gì?

b. Thuật toán kết quả từ $\alpha < \beta < 0$ là gì?

5.25 Giải thích cách các thuật toán lập lịch sau đã đối xử ủng hộ hoặc chống lại các tiến trình ngắn:

a. FCFS

b. RR

c. Hàng đợi đa cấp có phản hồi.

5.26 Mô tả lý do tại sao hàng đợi sẵn sàng chia sẻ có thể bị ảnh hưởng bởi các vấn đề hiệu suất trong môi trường SMP.

5.27 Xem xét thuật toán **cân bằng tải** (load-balancing algorithm) để đảm bảo rằng mỗi hàng đợi có xấp xỉ cùng số lượng tiến trình, không phụ thuộc vào mức độ ưu tiên. Thuật toán lập lịch dựa trên ưu tiên sẽ xử lý hiệu quả như thế nào tình huống này nếu một hàng đợi thứ nhất có tất cả các tiến trình có độ ưu tiên cao và một hàng đợi thứ hai có tất cả các tiến trình có độ ưu tiên thấp?

5.28 Giả sử rằng một hệ thống SMP có hàng đợi chạy riêng cho mỗi bộ xử lý. Khi một tiến trình mới được tạo ra, nó có thể được đặt trong cùng một hàng đợi với tiến trình cha hoặc một hàng đợi riêng biệt.

a. Lợi ích của việc đặt một tiến trình mới vào trong cùng một hàng đợi với tiến trình cha nó là gì?

b. Những lợi ích của việc đặt tiến trình mới trong một hàng đợi khác (với hàng đợi của tiến trình cha) là gì?

5.29 Giả sử rằng một tiến trình đã bị chặn để thao tác mạng và đã đủ điều kiện để chạy lại. Mô tả lý do tại sao thuật toán lập lịch nhận biết NUMA (NUMA-aware scheduling algorithm) nên lên lịch lại cho tiến trình đó lên cùng nhân CPU mà nó đã chạy trước đó.

5.30 Sử dụng thuật toán lập lịch Windows, xác định trị số độ ưu tiên của mỗi tiến trình sau đây:

a. Một tiến trình trong lớp REALTIME PRIORITY CLASS với độ ưu tiên liên quan là NORMAL.

b. Một tiến trình trong lớp ABOVE NORMAL PRIORITY CLASS với độ ưu tiên liên quan là HIGHEST.

c. Một tiểu trình trong lớp BELOW NORMAL PRIORITY CLASS với độ ưu tiên liên quan là ABOVE NORMAL.

5.32 Xem xét thuật toán lập lịch trong hệ điều hành Solaris cho tiến trình chia sẻ thời gian.

- Lượng thời gian (tính bằng mili giây) cho một tiểu trình với độ ưu tiên là 15? Với độ ưu tiên là 40?
- Giả sử rằng một tiểu trình có độ ưu tiên là 50 đã sử dụng toàn bộ lượng thời gian của nó mà không chặn. Bộ lập lịch sẽ gán giá trị ưu tiên mới nào cho tiểu trình này?
- Giả sử rằng một trình có độ ưu tiên là 20 bị block (do có thao tác nhập xuất) trước khi nó dùng hết lượng thời gian được cấp. Bộ lập lịch sẽ gán giá trị ưu tiên mới nào cho tiểu trình này?

5.33 Giả sử rằng hai tác vụ A và B đang chạy trên hệ thống Linux. Giá trị nice của A và B lần lượt là -5 và +5. Sử dụng bộ lập lịch CFS như một hướng dẫn, mô tả cách mà các giá trị tương ứng của runtime khác nhau giữa hai tiến trình được đưa ra cho mỗi tình huống sau:

- Cả A và B đều là tiến trình xu hướng tính toán CPU.
- A có xu hướng nhập / xuất và B có xu hướng tính toán CPU.
- A có xu hướng tính toán CPU và B có xu hướng nhập / xuất.

5.34 Cung cấp một tình huống cụ thể cho thấy lập lịch tỷ lệ đơn điệu kém hơn so với lập lịch hạn chót sớm nhất trước khi mà các hạn chót của các tiến trình thời gian thực trùng nhau?

5.35 Xem xét hai tiến trình, P_1 và P_2 , trong đó $p_1 = 50$, $t_1 = 25$, $p_2 = 75$ và $t_2 = 30$.

a. Hai tiến trình này có thể được lên lịch bằng cách sử dụng lập lịch tỷ lệ đơn điệu không? Minh họa câu trả lời của bạn bằng biểu đồ Gantt, ví dụ như các hình sau:

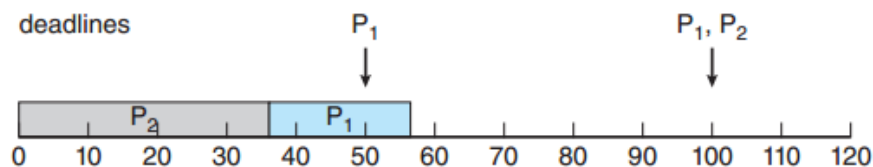


Figure 5.21 Scheduling of tasks when P_2 has a higher priority than P_1 .

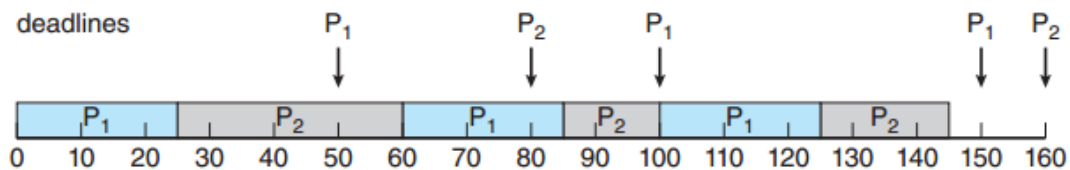


Figure 5.24 Earliest-deadline-first scheduling.

b. Minh họa lịch trình của hai tiến trình này bằng cách sử dụng lập lịch hạn chót gần nhất trước (EDF).

5.36 Giải thích tại sao thời gian ngắt và thời gian trễ khi điều phối phải bị giới hạn trong một hệ thống thời gian thực cứng?

5.37 Mô tả những lợi ích của việc sử dụng đa xử lý không đồng nhất trong một hệ thống di động.