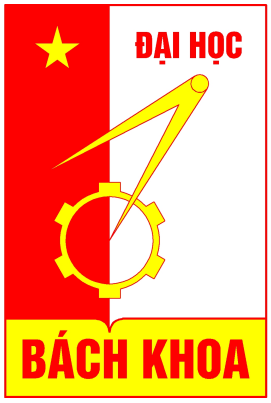
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
_____*******_____



BÁO CÁO
MÔN HỌC
Lập Trình Hệ Thống

Đề tài:

Quản lý tiến tình trong hệ điều hành Linux

Sinh viên thực hiện:

Họ và Tên	MSSV	Mã học phần
Đặng Quang Trung	20134145	IT3070

Giáo viên hướng dẫn:TS. Đỗ Quốc Huy

Hà Nội 10-05-2017

Mục lục

Chương 1

Hệ thống thời gian thực

1.1 Giới thiệu

Trong thế giới vật lý, mục đích của một hệ thống thời gian thực là có một thực hiện vật lý trong một khung thời gian đã chọn. Thông thường, một hệ thống thời gian thực bao gồm một hệ thống điều khiển(máy tính) và một hệ thống bị điều khiển (môi trường). Hệ thống điều khiển tương tác với môi trường của nó dựa trên thông tin có sẵn về môi trường. Trên một máy tính thời gian thực, điều khiển một thiết bị hoặc vi xử lý, các cảm biến sẽ cung cấp số lần đọc định kỳ và máy tính phải trả lời bằng cách gửi tín hiệu đến bộ truyền động. Có thể có các sự kiện không mong muốn hoặc bất thường và cũng phải nhận được phản hồi. Trong tất cả các trường hợp, sẽ có một khoảng thời gian mà trong đó phản hồi sẽ được gửi. Khả năng của máy tính để đáp ứng các yêu cầu này phụ thuộc vào khả năng thực hiện các phép tính cần thiết trong một khoảng thời gian nhất định. Nếu một số sự kiện xảy ra gần nhau, máy tính sẽ cần phải lên lịch tính toán sao cho mỗi phản hồi được cung cấp trong khoảng thời gian yêu cầu. Mỗi công việc xảy ra trong một hệ thống thời gian thực có một số thuộc tính thời gian. Các thuộc tính thời gian này cần được xem xét khi lập kế hoạch các nhiệm vụ trên một hệ thống thời gian thực. Các thuộc tính thời gian của một công việc nhất định xem các mục sau:

- Release time (or ready time): Thời gian mà công việc đã sẵn sàng để xử lý.
- Deadline: Thời gian thực hiện công việc phải được hoàn thành, sau khi tác vụ sẵn sàng.
- Minimum delay: Thời gian tối thiểu phải trôi qua trước khi thực hiện tác vụ được bắt đầu, sau khi tác vụ được giải phóng.
- Maximum delay: Khoảng thời gian tối đa cho phép trôi qua trước khi thực hiện tác vụ được bắt đầu, sau khi nhiệm vụ được giải phóng.
- Worst case execution time: Thời gian tối đa được thực hiện để hoàn thành nhiệm vụ, sau khi nhiệm vụ được giải phóng. Thời gian thực hiện trường hợp tồi tệ nhất còn được gọi là trường hợp thời gian phản ứng xấu nhất.

- Run time: Thời gian thực hiện mà không bị gián đoạn để hoàn thành nhiệm vụ, sau khi nhiệm vụ được giải phóng.
- Weight (or priority): Mức độ khẩn cấp tương đối của nhiệm vụ.

1.2 Hệ thống thời gian thực

Chúng ta có thể định nghĩa một hệ thống thời gia thực như sau. Xem xét một hệ thống chứa một tập các tác vụ, $T = \{\tau_1, \tau_2, \ldots, \tau_n\}$. Ở đây trường hợp xấu nhất thời gian thực hiện của mỗi tác vụ $\tau_i \in T$ là C_i. Hệ thống được cho là thời gia thực nếu có tồn tại ít nhất tác vụ $\tau_i \in T$, tác vụ rơi vào tình trạng:

1, Tác vụ τ_i là tác vụ **hard real-time**. Thời gian thực hiện tác vụ τ_i phải được hoàn thành bởi thời gian hết hạn D_i ($C_i \leq D_i$).

2, Tác vụ τ_i là tác vụ **soft real-time**. Tác vụ cuối cùng τ_i kết thúc tính toán của nó sau gian gian hết hạn D_i, sẽ bị phạt nặng hơn. Hàm phạt $P(\tau_i)$ được định nghĩa cho tác vụ. Nếu $C_i \leq D_i$ hàm phạt $P(\tau_i)$ bằng 0 trái lại $P(\tau_i) > 0$ giá trị này tăng theo $C_i - D_i$.

3, Tác vụ là **firm real-time**. tác vụ kết thúc công việc tính toán của nó sớm nhất trước thời hạn hết hạn D_i, sẽ nhận được thưởng. Hàm thưởng $R(\tau_i)$ được định nghĩa cho tác vụ. Nếu $C_i \geq D_i$, hàm $R(\tau_i) = 0$, trái lại $R(\tau_i) > 0$ giá trị này tăng theo $D_i - C_i$.

Một tập các tác vụ thời gian thực $T = \{\tau_1, \tau_2, \ldots, \tau_n\}$ có thể là hỗn hợp của hard, soft, firm các tác vụ thời gian thực.

Cho T_s là tập tất cả các tác vụ **soft real-time** trong T, ví dụ $T_s = \{\tau_{s,1}, \tau_{s,2}, \ldots, \tau_{s,l}\}$ với $\tau_{s,i} \in T$. Hàm lỗi của hệ thống sẽ là $P(T)$. Ở đây

$$P(T) = \sum_{i=1}^{l} P(\tau_{s,i})$$

Cho T_f là tập tất cả các tác vụ **soft real-time** trong T, ví dụ $T_f = \{\tau_{f,1}, \tau_{f,2}, \ldots, \tau_{f,k}\}$ với $\tau_{f,i} \in T$. Hàm thưởng của hệ thống sẽ là $R(T)$. Ở đây

$$R(T) = \sum_{i=1}^{k} R(\tau_{f,i})$$

1.3 Lập lịch thời gian thực

Cho một số công việc nhất định, vấn đề lập lịch yêu cầu sắp xếp các công việc sẽ được thực hiện sao cho các ràng buộc khác nhau được thỏa mãn. Thông thường, một công việc được đặc trưng bởi thời gian thực hiện của nó, thời

gian đã sẵn sàng, thời hạn, và các yêu cầu về tài nguyên. Việc thực hiện công việc có thể hoặc không thể bị gián đoạn (lập kế hoạch ưu tiên hoặc không bảo vệ). Trong tập hợp các công việc, có một mối quan hệ ưu tiên làm hạn chế thứ tự thực hiện. Đặc biệt, việc thực hiện một công việc không thể bắt đầu cho đến khi thực hiện tất cả các công việc trước đó của nó (theo mối quan hệ ưu tiên) được hoàn thành. Hệ thống mà công việc được thực hiện được đặc trưng bởi số lượng các tài nguyên sẵn có. Các mục tiêu sau đây cần được xem xét trong việc lập lịch trình một hệ thống thời gian thực:

- Đáp ứng các ràng buộc thời gian của hệ thống.
- Ngăn chặn truy cập đồng thời tới các tài nguyên và thiết bị dùng chung.
- Tận dụng mức độ sử dụng cao trong khi vẫn đáp ứng các ràng buộc thời gian của hệ thống; Tuy nhiên đây không phải là trình điều khiển chính.
- Giảm chi phí của thiết bị chuyển mạch ngữ cảnh gây ra bằng cách phạt.
- Giảm chi phí truyền thông trong các hệ thống phân phối thời gian thực; Chúng ta nên tìm cách tối ưu để phân tích ứng dụng thời gian thực thành các phần nhỏ hơn để có chi phí truyền thông tối thiểu giữa các phần tương hỗ (mỗi phần được gán cho máy tính).

Ngoài ra, các mục sau đây là mong muốn trong các hệ thống thời gian thực tiên tiến.

- Xem xét sự kết hợp của hard, firm, and soft real-time, có nghĩa là khả năng áp dụng các chính sách lập lịch trình động đáp ứng các tiêu chí tối ưu.
- Lập kế hoạch công việc cho một hệ thống thời gian thực có hành vi tự động thích nghi, có thể khôi phục, phản xạ và thông minh.
- Bao gồm độ tin cậy, an toàn và an toàn.

Với một hệ thống thời gian thực, cách tiếp cận lập kế hoạch thích hợp nên được thiết kế dựa trên các thuộc tính của hệ thống và các nhiệm vụ xảy ra trong đó. Những tính chất này như sau:

- **Soft/Hard/Firm real-time tasks**
- **Periodic/Aperiodic/Sporadic tasks**
- **Preemptive/Non-preemptive tasks**
- **Multiprocessor/Single processor systems**
- **Fixed/Dynamic priority tasks**
- **Flexible/Static systems**
- **Independent/Dependent tasks**

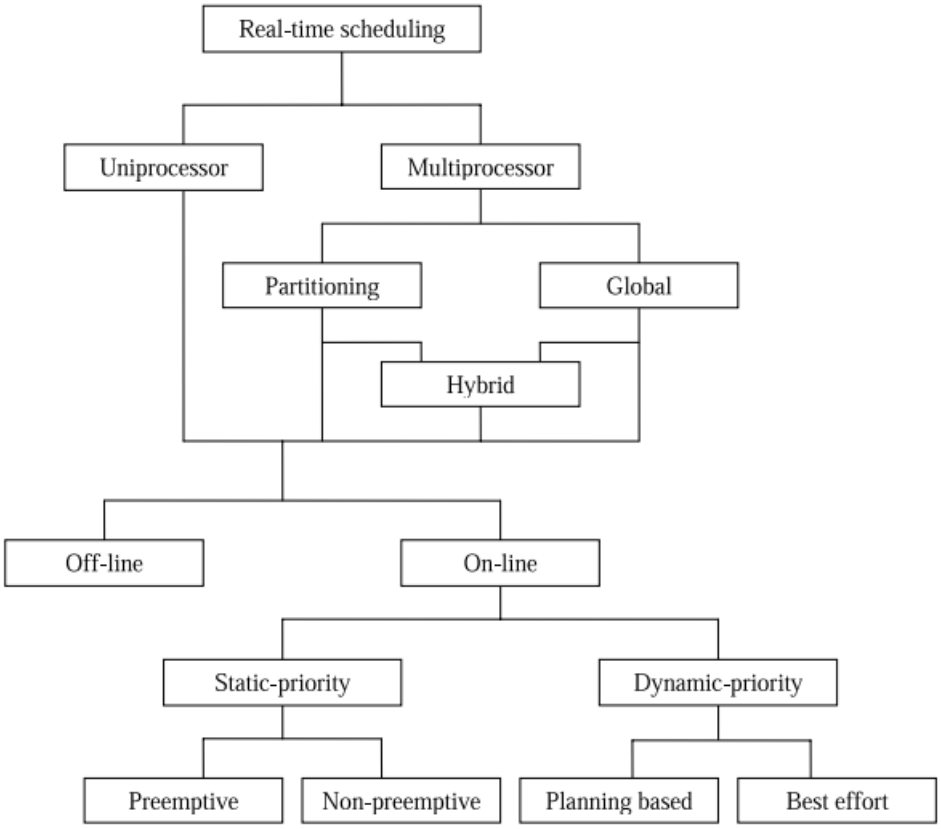
Chương 2

Thuật toán lập lịch cho hệ thống thời gian thực

2.1 Tổng quan

Một hệ thống thời gian thực là một hệ thống phải đáp ứng các ràng buộc hạn chế thời gian phản hồi rõ ràng, nếu không thì sẽ dẫn tới những hậu quả nghiêm trọng bao gồm cả sự thất bại. Thất bại xảy ra khi một hệ thống không thể đáp ứng được một hoặc nhiều yêu cầu đặt ra trong các đặc tả hệ thống chính thức.

Đối với một tập hợp các tác vụ nhất định, vấn đề lập kế hoạch tổng quát yêu cầu sắp theo đó các tác vụ sẽ được thực hiện sao cho các ràng buộc khác nhau được thỏa mãn. Đối với một bộ tác vụ thời gian thực, chúng ta được yêu cầu lập ra một phân bổ khả thi / lịch trình. Thời gian giải phóng, thời hạn và thời gian thực hiện các tác vụ là một số thông số cần được xem xét để lập kế hoạch. Thời hạn có thể là hard, soft hoặc firm. Các vấn đề khác cần được xem xét như sau. Đôi khi, một nguồn lực phải được duy nhất giữ bởi một tác vụ. Các tác vụ có thể có các ràng buộc ưu tiên. Một tác vụ có thể là định kỳ, không đều, hoặc rời rạc. Lịch trình có thể là ưu tiên hoặc không bắt buộc. Các nhiệm vụ ít quan trọng hơn phải được các cấp trên phê duyệt trước khi cần thiết để đến thời hạn. Đối với các hệ thống thời gian thực trong đó các nhiệm vụ đến rất nhiều, chúng ta phải sử dụng nhiều bộ xử lý để đảm bảo rằng các tác vụ được lên kế hoạch khả thi. Do đó, số bộ xử lý có sẵn là một tham số khác để xem xét. Các bộ xử lý có sẵn có thể giống hệt, thống nhất hoặc không liên quan.



Hình 2.1: Thuật toán lập trình thời gian thực

Ở đây trình bày 2 giải thuật Rate-Monotonic algorithm (RM) and Earliest Deadline First algorithm (EDF). Cả hai thuật toán RM và EDF là thuật toán lập lịch trình thời gian thực tối ưu. Thuật toán lập lịch trình thời gian thực tối ưu là một thuật toán không thể đáp ứng được thời hạn chỉ nếu khi không một thuật toán lập lịch trình khác có thể đáp ứng được thời hạn. Các giả định sau được thực hiện cho cả các thuật toán RM và EDF

(a) Không tác vụ nào có bất kì phần trưng dụng và giá của trưng dụng là không đáng kể.

(b) Chỉ có các yêu cầu về bộ nhớ, I/O là quan trọng, còn các yêu vâu tài nguyên khác là không đáng kể.

(c) Tất cả các tác vụ là độc lập và không có ràng buộc về ưu tiên.

2.2 Thuật toán lập lịch Rate-Monotonic(RM)

Thuật toán lập lịch RM là một trong những nghiên cứu và sử dụng rộng rãi nhất trong thực tiễn. Thuật toán lập lịch cho đơn bộ xử lý với ưu tiên static. Đối với thuật toán lập kế hoạch RM, ngoài các giả định (a) đến (c), chúng ta giả định rằng tất cả các tác vụ là có định kì và độ ưu tiên của tác vụ τ_i cao hơn độ ưu tiên của tác vụ τ_j Ở đây i $<$ j. Thuật toán lập lịch RM là một ví dụ về các thuật toán được ưu tiên điều khiển với sự phân cấp ưu

tiên tĩnh theo ý nghĩa rằng các ưu tiên của tất cả các trường hợp được biết đến ngay cả trước khi chúng đến. Các ưu tiên của tất cả các trường hợp cho mỗi tác vụ là như nhau. Chúng được xác định bởi chu kì của tác vụ. Một chu kì của tác vụ bao gồm một dãy vô hạn các trường hợp với số lần chu kì sẵn sàng. Ở đây thời hạn của một yêu cầu có thể nhỏ hơn, lớn hơn, hoặc bằng thời gian đã sẵn sàng của trường hợp thành công. Hơn nữa, thời gian thực thi của tất cả các trường hợp cả của tác vụ là như nhau. Một chu kì của tác vụ τ được đặc trưng bởi 3 tham số P_i là thời gian định kì, C_i là thời gian thực thi và D_i là thời gian hết hạn. Yếu tố sử dụng của một tập hợp n định kỳ tác vụ được định nghĩa bởi $\sum_{i-1}^{n} C_i/P_i$, ở đây $P_1, P_2, \ldots, P_n$ là các thời gian chu kì và $C_1, C_2, \ldots, C_n$ là thời gian thực thi của n tác vụ. Nếu $\sum_{i-1}^{n} C_i/P_i \leq n(2^{1/n} - 1)$, Ở đây n là số tác vụ cần lập lịch, sau đó thuật toán RM sẽ lập lịch các tác vụ để đáp ứng thời hạn của các tác vụ. Lưu ý rằng điều này là đủ, nhưng không phải là điều kiện cần thiết, vó thể có các tập tác vụ với mức sử dụng lớn hơn $n(2^{1/2} - 1)$ được lập lịch bởi thuật toán RM.

Cho mộ tập các tác vụ gọi là lập lịch được bởi thuật toán RM nếu thuật toán đưa ra một lập lịch gặp tất cả các thời gian hết hạn. Điều kiện cần và đủ cho tính khả thi của lập lịch RM như sau.

Tập gồm n chu kì của các tác vụ $\tau_1, \tau_2, \ldots, \tau_n$ có thời gian và thời gian thực hiện là $P_1, P_2, \ldots, P_n$ và $C_1, C_2, \ldots, C_n$ tương ứng, chúng ta giả sử τ_i hoàn thành ở t. Chúng ta có:

$$W_i(t) = \sum_{j=1}^{i} C_j \lceil \frac{t}{P_j} \rceil = t - idletime$$

$$L_i(t) = \frac{W_i(t)}{t}$$

$$L = \min_{0 \leq t \geq P_i} L_i(t)$$

Tác vụ τ_i có thể lập lịch khả thi sử dụng RM nếu và chỉ nếu $L_i(t) \leq 1$. Trong trường hợp này $\tau_1, \tau_2, \ldots, \tau_{i-1}$ là cũng khả thi.

Một nhược điểm của thuật toán RM là ưu tiên tác vụ được xác định bởi các chu kì của chúng. Đôi khi, chúng ta phải thay đổi các tác vụ ưu tiên để đảm bảo rằng tất cả các tác vụ quan trọng được hoàn thành. Giả sử chúng ta có một tập các tác vụ có chứa 2 tác vụ τ_i và τ_j ở đây $P_i < P_j$ nhưng τ_j là tác vụ quan trọng hơn và τ_i là tác vụ không quan trọng. Chúng ta kiểm tra tính khả thi của thuật toán RM cho các tác vụ $\tau_1, \tau_2, \ldots, \tau_n$. Giả sử nếu chúng ta thực hiện trường hợp tồi nhất của các tác vụ, Chúng ta không thể đảm bảo tính có thể lập trình của các tác vụ. Tuy nhiên, trong trường hợp trung bình, chúng đều có thể lập biểu cho RM. Vấn đề là làm thế nào để sắp xếp các tác vụ sao cho tất cả các tác vụ quan trọng đều đạt được thời hạn

của chúng theo thuật toán RM ngay cả trong trường hợp xấu nhất, trong khi các nhiệm vụ không quan trọng, như τ_i, đáp ứng thời hạn của chúng trong nhiều trường hợp khác. Có 1 trong 2 giải pháp sau:

- Chúng ta sẽ kéo dài thời gian của các tác vụ không quan trọn, ví dụ τ_i bởi yếu tố k. Tác vụ ban đầu cũng cần được thay thế bằng k tác vụ, từng giai đoạn theo chu kì thích hợp tham số k nên được chọn sao cho $P_i^{'} > P_j$
- Chúng ta sẽ giảm chu kì của tác vụ τ_j bởi yếu tố k. Sau đó chúng ta sẽ thay thế tác vụ ban đầu bằng môt thời gian thực thi giảm bởi yếu tố k. Tham số k nên được chọ sao cho chúng ta đạt được $P_i > P_j^{'}$

Cho đến nay, chúng ta đã giả định rằng thời hạn tương đối của một nhiệm vụ bằng khoảng thời gian của nó. Nếu chúng ta bỏ giả định này, thuật toán RM không còn là thuật toán lập lịch trình tối ưu static-priority. Khi $D_i \leq P_i$, nhiều nhất một tác vụ bắt đầu có thể tồn tại trong bất kì thời điểm. Tuy nhiên khi $d_i > P_i$ nó có thể bắt đầu với nhiều tác vụ như nhau tồn tại đồng thời. Do đó, kiểm tra khả năng định thời của RM đối với trường hợp $D_i > P_i$ là khó hơn nhiều so với trường hợp $D_i \leq P_i$. Thuât toán RM chạy với thời gian $O((N+\alpha)^2)$ trong trường hợp tồi nhất, ở đây N là tổng số request của mỗi hyper-period của e periodic tác vụ trong hê thống và α tổng số tác vụ không có định kì.

2.3 Thuật toán EDF

EDF là một thuật toán lập lịch ưu tiên trong đó ưu tiên cao hơn được gán cho yêu cầu có thời hạn sớm hơn, và yêu cầu ưu tiên cao hơn luôn luôn chiếm quyền của một ưu tiên thấp hơn. EDF còn được gọi là thuật toán lập lịch biểu đơn điệu deadline-monotonic. Giả sử mỗi khi một tác vụ đã sẵn sàng mới đến, nó được chèn vào một hàng đợi các tác vụ đã sẵn sàng, sắp xếp theo thời hạn của chúng. Nếu danh sách được sắp xếp, thuật toán EDF có độ phức tập $O((N+\alpha)^2)$ trong trường hợp tồi nhất, Ở đây N là tổng số yêu cầu trong mỗi hyper-period của n chu kì các tác vụ trong hệ thống và α là số tác vụ không định kì.

Đối với thuật toán EDF, chúng ta thực hiện tất cả các giả định mà chúng ta đã thực hiện cho thuật toán RM, ngoại trừ các nhiệm vụ không phải là định kỳ

Thuật toán EDF là tối ưu với lập trình đơn bộ xử lí. Nghĩa là, nếu EDF không thể lập lịch trình một nhiệm vụ trên một bộ xử lý đơn, thì không có thuật toán lập lịch trình nào có thể. Điều này có thể được chứng minh bằng cách sử dụng một kỹ thuật trao đổi slice thời gian. Sử dụng kỹ thuật này, chúng ta có thể chỉ ra rằng bất kỳ lịch trình hợp lệ nào cho bất kỳ nhóm tác vụ nào cũng có thể được chuyển thành lịch biểu EDF hợp lệ.

Nếu tất cả các tác vụ đều định kỳ và có thời hạn tương ứng với thời gian của chúng, thì lập lịch có thể khả thi bởi EDF nếu và chỉ nếu $\sum_{i-1}^{n} C_i/P_i \leq 1$. Không có kiểm tra tính khả thi đơn giản tương ứng với trường hợp các thời hạn tương đối không bằng các khoảng thời gian định kì. Trong trường hợp đó, chúng ta phải phát triển một lập lịch sử dụng thuật toán EDF để xem liệu tất cả các thời hạn có thể đạt được trong một khoảng thời gian nhất định hay không. Sau đây là kiểm tra tính khả thi cho EDF trong trường hợp này.

Định nghĩa $U = \sum_{i=1}^{n} C_i/P_i, D_{max} = \max_{1 \leq i \leq n} D_i$ và $P = lcm(P_1, \ldots, P_n)$, ở lcm là bội số chung nhỏ nhất. Xem xét $h(t)$ Là tổng thời gian thực hiện của tất cả các nhiệm vụ có thời hạn tuyệt đối nhỏ hơn t. Một tập các tac vụ n là không có khả thi vơi EDF nếu và chỉ nếu.

- $U < 1$ hoặc
- Tồn tại $t < minP + D_{max}, \frac{U}{1-U} \max_{1 \leq i \leq n} P_i - D_i$ sao cho $h(t) > t$

Mặc dù nhiều người đã làm việc về phân tích tính khả thi của các thuật toán đa thức, vẫn cần phải nghiên cứu thêm. Việc xác minh tính tối ưu của thuật toán lập lịch trình là một chủ đề khác cần được nghiên cứu sâu hơn.

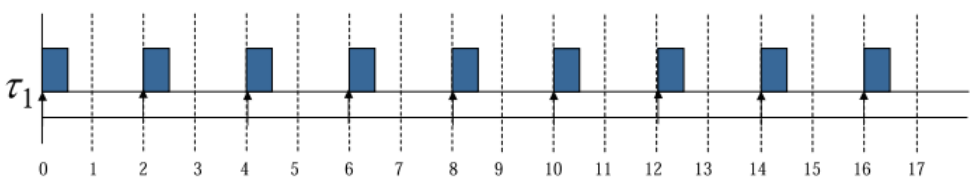
2.4 Ví dụ về lập lịch

Xem xét một hệ thống có chứa 3 tác vụ τ_1, τ_2, τ_3 vó thời gian lặp lại, thời gian tính toán, lần gọi đầu tiên, thời gian hết hạn được định nghĩa trong bảng hình 2.2.

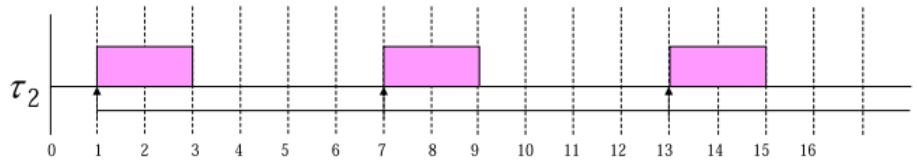
	Period	*Computation time*	*First invocation time*	*Deadline*
τ_1	2	0.5	0	2
τ_2	6	2	1	6
τ_3	10	1.8	3	10

Hình 2.2: Bảng minh họa

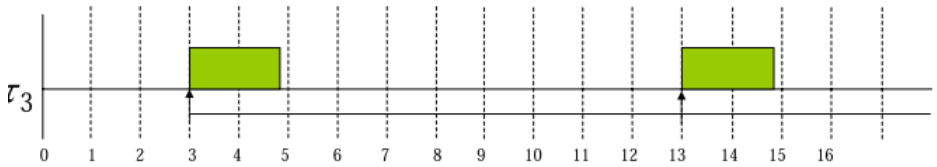
Các hình 2.3, 2.4, 2.5 biểu diễn sơ đồ thời gian của mỗi tác vụ τ_1, τ_2, τ_3 trước khi lập lịch



Hình 2.3: Sở đồ thời gian tác vụ τ_1



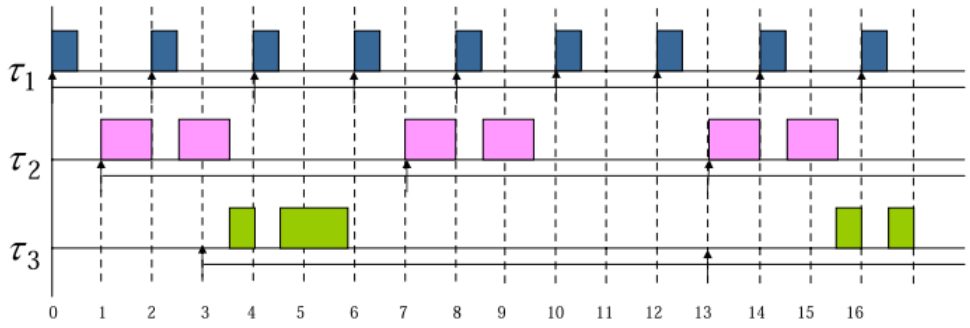
Hình 2.4: Sở đồ thời gian tác vụ τ_2



Hình 2.5: Sở đồ thời gian tác vụ τ_3

- *Thuật toán EDF*

hình 2.6 biểu diễn một phần của biểu đồ thời gian của lịch trình được cung cấp bởi thuật toán EDF về các tác vụ được xác định trong Bảng 2.2. Trong khoảng thời gian 0 và 17, chúng ta thấy rằng không có hạn chót nào bị bỏ qua.



Hình 2.6: Thuật toán ED,RM

- *Thuật toán RM*

  Nếu chúng ta lập kế hoạch cho các tác vụ được thiết lập bởi thuật toán RM, không có hạn chót nào được bỏ qua giữa khoảng thời gian 0 và 17.

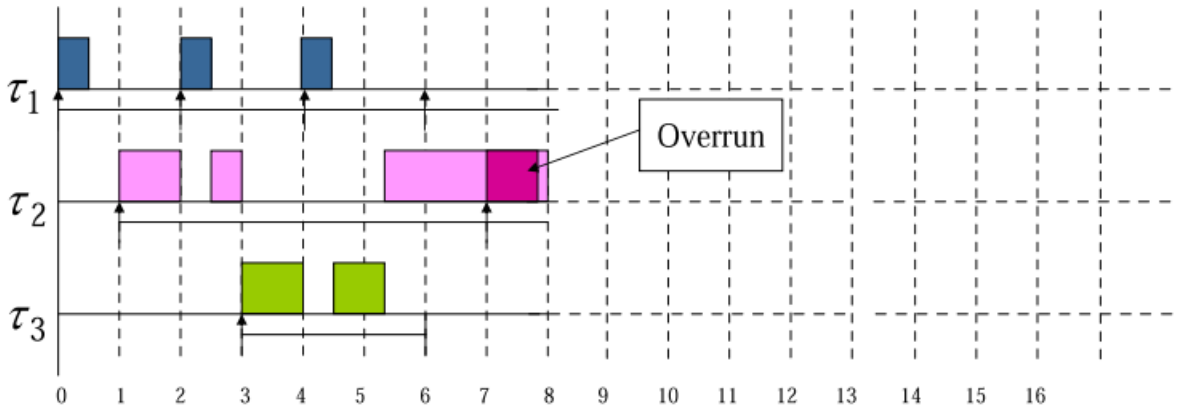
Xem xét một hệ thống có chứa 3 tác vụ τ_1, τ_2, τ_3 vó thời gian lặp lại, thời gian tính toán, lần gọi đầu tiên, thời gian hết hạn được định nghĩa trong bảng hình 2.7.

	Period	*Computation time*	*First invocation time*	*Deadline*
τ_1	2	0.5	0	2
τ_2	6	4	1	6
τ_3	3	1.8	3	10

Hình 2.7: Bảng số liêu

- *Thuật toán EDF*

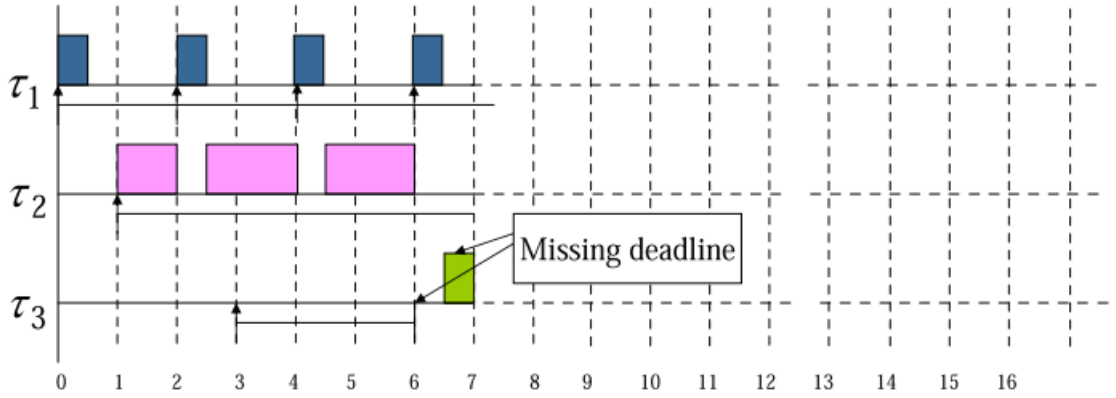
  Như được trình bày trong Hình 2.8, hệ thống thời gian thực đơn nhất bao gồm các tác vụ được xác định trong Bảng 2.7 không lập lịch bởi EDF, bởi vì trong khi việc thực hiện lời gọi đầu tiên của tác vụ τ_2 vẫn chưa kết thúc thì lời gọi mới của tác vụ đến. Nói cách khác, tình trạng vượt qua sẽ xảy ra.



Hình 2.8: Thuật toán EDF

- *Thuật toán RM*

  Như thể hiện trong hình 2.10, hệ thống thời gian thực thống nhất bao gồm các tác vụ được xác định trong Bảng 2.7 không thể lập lịch theo RM. Lý do là thời hạn của cuộc gọi đầu tiên của tác vụ τ_3 là bỏ qua. Việc thực hiện yêu cầu đầu tiên được yêu cầu phải hoàn thành theo thời gian 6, nhưng lịch trình không thể làm cho nó.



Hình 2.9: Thuật toán RM

2.5 Các vấn đề mở

Một danh sách các vấn đề mở cho quá trình lập lịch đa bộ xử lí thực như sau:

Xem xét 1 tập các tác vụ thời gian thực hard, soft và firm , $T = \{\tau_1, \tau_2, \ldots, \tau_n\}$ ở đây thời gian thực hiện trong trường hợp tồi nhất của mỗi tác vụ $\tau_i \in T$ là C_i

(1) Nếu các nhiệm vụ thời gian thực là hard, định kỳ, ưu tiên và có ưu tiên cố định, sau đó tìm số bộ vi xử lý tối thiểu cần thiết để đảm bảo rằng tất cả các thời hạn được đáp ứng. Một số thuật toán heuristic đã được đề xuất, tuy nhiên chúng ta tin rằng thuật toán tốt hơn với hiệu suất được cải thiện có thể được phát triển.

(2) Giả sử trong một hệ thống bao gồm k bộ xử lý giống nhau, tác vụ thời gian thực là ưu tiên và có các ưu tiên cố định. Tác vụ thời gian thực hard là định kỳ. Chi phí truyền thông là không đáng kể. Tìm một lịch làm giảm thiểu thời gian đáp ứng trung bình đồng thời đảm bảo rằng tất cả các thời hạn đều được đáp ứng.

(3) Giả sử có k bộ xử lý giống nhau, tác vụ thời gian thực là ưu tiên và có ưu tiên cố định, một chức năng phạt $P(\tau_i)$ được gán cho mỗi tác vụ thời gian thực soft, và một chức năng thưởng $R(\tau_i)$ được xác định cho mỗi tác vụ thời gian thực của firm. Chi phí truyền thông là không đáng kể. Tìm một lịch trình đảm bảo tất cả các thời hạn được đáp ứng và $\frac{P(T)}{R(T)}$được giảm thiểu.

(4) v.v.

Tài liệu tham khảo

[1] Technical Report No. 2005-499 Scheduling Algorithms for Real-Time Systems of *Arezou Mohammadi and Selim G. Akl*

[2] Scheduling Algorithms for Real-Time Systems of *Fredrik Lindh and Thomas Otnes and Jessica Wennerström*

[3]