

MÔN: HỆ ĐIỀU HÀNH

Bài thực hành số 4.1: LẬP LỊCH CHẠY CÁC PROCESS (THREAD)

I. Mục tiêu

- Giúp SV củng cố kiến thức về 1 số giải thuật lập lịch đã học.

II. Nội dung :

- Xây dựng các biểu đồ và kịch bản chạy của hệ thống theo các giải thuật lập lịch khác nhau.

III. Chuẩn đầu ra

- Sinh viên nắm vững 1 số giải thuật lập lịch đã học.

IV. Qui trình

1. Nhắc lại lý thuyết

Giải thuật lập lịch cho phép chọn 1 process trong danh sách các process đang ready để giao CPU chạy tiếp nó. Giải thuật lập lịch có thể hay hoàn toàn không dùng thông tin quyền ưu tiên của các process.

Với giải thuật lập lịch không cho phép ngắt, process được chọn sẽ giữ CPU cho đến khi nó kết thúc, nó không thể bị ngắt vì bất kỳ lý do gì (do process mới xuất hiện, do timeout của timer đếm giờ). Thí dụ giải thuật lập lịch SJN (Shortest Job Next) hay FCFS (First Come First Serve).

Với giải thuật trung dụng lại CPU nhưng không cho phép tuần hoàn, chỉ khi process mới với quyền ưu tiên cao hơn process đang chiếm CPU mới có thể ngắt process hiện hành để chiếm CPU, lúc này process bị dừng sẽ quay về đầu hàng chờ để được chạy tiếp khi process mới với quyền ưu tiên cao nhất hiện tại chạy xong. Thí dụ giải thuật lập lịch PSJN (Pre-emptive shortest job next).

Trong hệ thống phân chia thời gian, mỗi process chỉ có thể chạy tối đa 1 khe nhỏ thời gian (quantum) rồi phải dừng chờ để được chọn sau đó.

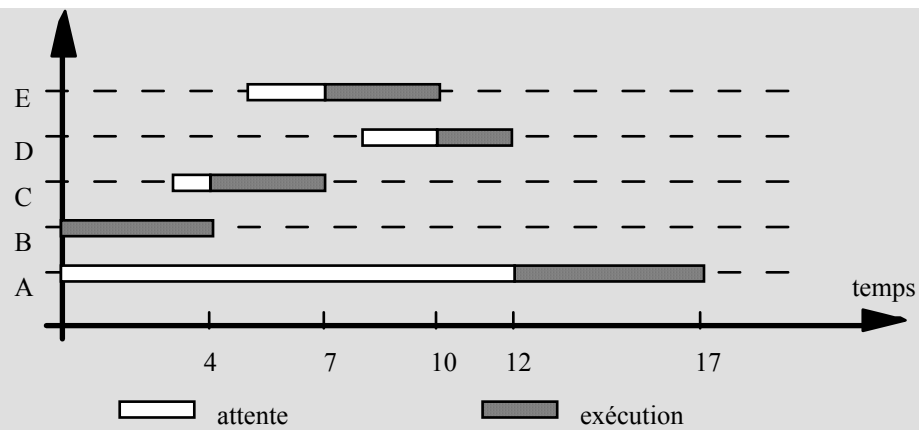
Một giải thuật lập lịch tốt cần phải tránh được nạn đói (starvation), nghĩa là không để 1 process nào bị chờ vô hạn vì không bao giờ được chọn để chạy, như vậy nó sẽ không bao giờ hoàn thành công việc của mình.

2. Lập lịch trong chế độ lô (batch)

Ta hãy xét thời điểm kích hoạt và thời gian cần chạy của các process sau đây :

process	A	B	C	D	E
Thời điểm kích hoạt	0	0	3	8	5
Thời gian chạy	5	4	3	2	3

2.1 Nếu dùng giải thuật lập lịch SJN (), hãy miêu tả việc thi hành các process dưới dạng biểu đồ. Hãy tính thời gian đáp ứng và tỉ lệ phạt của từng process. Có nguy cơ xuất hiện nạn đói trên các process không? Giải thích.



Thời gian đáp ứng = lúc kết thúc - lúc được kích hoạt chạy

A : 17 B : 4 C : 4 D : 4 E : 5

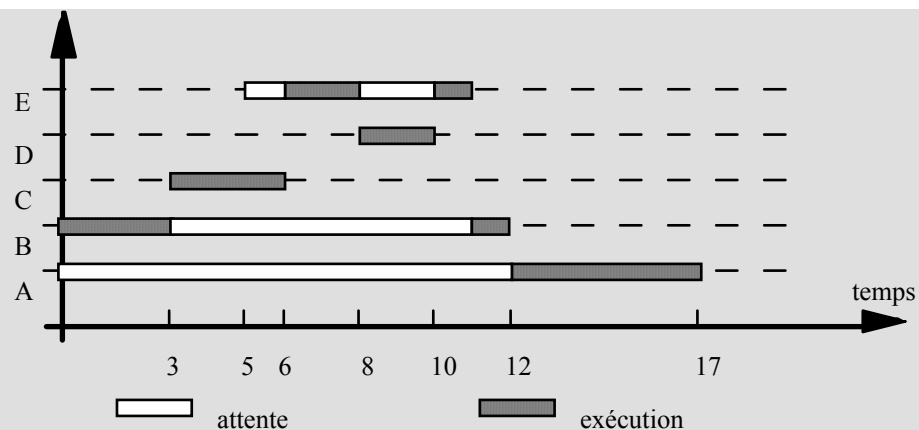
Lưu ý : thời gian đáp ứng không nhất thiết là độ đo tốt về hiệu năng làm việc của hệ thống vì nó không để ý tới nhu cầu dùng CPU của từng process. Thí dụ với các thông số trên, các process B, C, D có cùng thời gian đáp ứng nhưng chúng không cùng mức độ thỏa mãn vì chúng yêu cầu thời gian chạy khác nhau (4, 3, 2).

Tỉ lệ phạt = thời gian đáp ứng / thời gian được phục vụ

A : 3.4 B : 1 C : 1.33 D : 2 E : 1.66

Với giải thuật lập lịch SJN (hay PSJN), nạn đói chỉ xảy ra khi CPU bận 100% liên tục theo thời gian. Cụ thể nếu các process ngắn được kích hoạt đều đều, process dài sẽ không có cơ may được chạy. Tuy nhiên trình trạng này rất ít khi xảy ra trong thực tế.

2.2 Nếu dùng giải thuật lập lịch PSJN (), hãy miêu tả việc thi hành các process dưới dạng biểu đồ. Hãy tính thời gian đáp ứng và tỉ lệ phạt của từng process. Có nguy cơ xuất hiện nạn đói trên các process không? Giải thích.



Trong giải thuật lập lịch này, 1 process mới chỉ có thể ngắt process khác nếu nó có quyền ưu tiên cao hơn (tức có thời gian chạy ngắn hơn).

Thời gian đáp ứng : A : 17 B : 12 C : 3 D : 2 E : 6

Tỉ lệ phạt : A : 3.4 B : 3 C : 1 D : 1 E : 2

3. Lập lịch trong chế độ phân chia thời gian (time sharing)

Giải thuật Round-robin

Trong giải thuật lập lịch xoay vòng (**round-robin**), các process được xếp trong 1 hàng chờ duy nhất, ai tới trước sắp trước, ai tới sau sắp sau. Khi CPU rảnh, trình lập lịch sẽ cho CPU

chạy process ở đầu danh sách. Khi bị tước CPU (chờ I/O, hết khe chạy, thực hiện down semaphore,...), process sẽ được xếp vào cuối hàng chờ để được phục vụ sau. Trình lập lịch sẽ cho CPU chạy process ở đầu danh sách với điều kiện nó không bị blocked vì chờ I/O, nếu không trình lập lịch sẽ duyệt xem xét các process kế tiếp cho đến khi tìm được process ready để giao CPU cho nó chạy.

Trong bài thực hành này, ta giả sử máy chỉ có 1 CPU và 1 thiết bị I/O. Hãy xét các process sau :

	Times CPU	I/O	Thời gian I/O
T1	300 ms	Không có	
T2	30 ms	Sau mỗi 10 ms	250 ms
T3	200 ms	Không có	
T4	40 ms	Sau mỗi 20 ms	180 ms

Lưu ý T2 có 3 lần I/O, T4 có 2 lần I/O.

3.1 Miêu tả chính xác sự tiến triển của hệ thống : từng thời điểm, sự kiện gì xảy ra (chuyển ngữ cảnh - CNC, yêu cầu I/O, xong I/O), process được chọn, trạng thái hàng chờ. Với từng process, xác định thời điểm kết thúc chạy. Giả sử quantum = 100ms, các process lúc đầu đều ready và được xếp theo chỉ số của chúng, card điều khiển đĩa điều khiển 2 đĩa, 2 process T2 và T4 thực hiện I/O trên các đĩa khác nhau.

Trong bảng dưới đây, ký hiệu 1p nghĩa là process 1 ready, 2b nghĩa là process 2 blocked

Time	Sự kiện	Process được chọn	Trạng thái hàng chờ	Chú thích
0	Nạp chạy T1	T1	1p 2p 3p 4p	
100	Chuyển ngữ cảnh	T2	2p 3p 4p 1p	
110	I/O của T2, CNC	T3	3p 4p 1p 2b	
210	Chuyển ngữ cảnh	T4	4p 1p 2b 3p	
230	I/O của T4, CNC	T1	1p 2b 3p 4b	
330	Chuyển ngữ cảnh	T3	2b 3p 4b 1p	
360	I/O của T2	T3	2p 3p 4b 1p	
410	I/O của T4	T3	2p 3p 4p 1p	
430	Chuyển ngữ cảnh	T2	2p 4p 1p	T3 xong
440	I/O của T2, CNC	T4	4p 1p 2b	
460	I/O của T4, CNC	T1	1p 2b 4b	
560	Chuyển ngữ cảnh	Không có	2b 4b	T1 xong
640	xong I/O của T4	Không có	2b	T4 xong
690	xong I/O của T2	T2	2p	
700	I/O của T2, CNC	Không có	2b	
950	xong I/O của T2	Không có	Trống	T2 xong

3.2 Hãy nêu ưu khuyết điểm của giải thuật lập lịch round-robin. Nạn đói có thể xảy ra không? Vì sao?

Ưu : đơn giản.

Khuyết : không để ý tới các process cần khẩn cấp (ưu tiên hơn) các process khác.

Nguyên cơ nạn đói : Không, sau khi process vào hàng chờ, nó sẽ được chạy sau 1 khoảng thời gian hữu hạn. Cụ thể, chỉ có các process xếp trước nó mới có thể chạy tối đa 1 khe thời gian trước khi nó được chạy.

Giải thuật round-robin không cho phép để ý tới các process cần khẩn cấp (ưu tiên hơn) các process khác. Do đó ta cần giải thuật lập biểu khác mà xử lý quyền ưu tiên của các process.

Lập lịch với quyền ưu tiên cố định

Trong giải thuật này, mỗi process được cấp 1 quyền ưu tiên xác định lúc bắt đầu chạy và nó giữ quyền này cho đến khi thi hành xong toàn bộ.

Giả sử có 4 mức quyền ưu tiên, đánh số từ 0 - 3 (0 ưu tiên thấp nhất). Xét các process sau đây :

	Time CPU	I/O	Thời gian I/O	Mức ưu tiên
T1	300 ms	Không có		0
T2	30 ms	Sau mỗi 10 ms	250 ms	3
T3	200 ms	Không có		1
T4	40 ms	Sau mỗi 20 ms	180 ms	3
T5	300 ms	Không có		1

3.3 Miêu tả chính xác sự tiến triển của hệ thống : từng thời điểm, sự kiện gì xảy ra (chuyển ngữ cảnh - CNC, yêu cầu I/O, xong I/O), process được chọn, trạng thái hàng chờ. Với từng process, xác định thời điểm kết thúc chạy. Giả sử quantum = 100ms, các process lúc đầu đều ready và được xếp theo chỉ số của chúng, card điều khiển đĩa điều khiển 2 đĩa, 2 process T2 và T4 thực hiện I/O trên các đĩa khác nhau.

Ứng với mỗi mức quyền ưu tiên, có 1 hàng chờ các process cùng quyền ưu tiên ở mức này. Ta dùng giải thuật round-robin để chọn process chạy trong từng hàng chờ.

Trong bảng dưới đây, ta dùng dấu / để ngăn 4 hàng chờ của 4 mức ưu tiên khác nhau.

Time	Sự kiện	Process được chọn	Trạng thái hàng chờ	Chú thích
0	Nạp chạy T2	T2	2p 4p / / 3p 5p / 1p	
10	I/O của T2, CNC	T4	4p 2b / / 3p 5p / 1p	
30	I/O của T4, CNC	T3	2b 4b / / 3p 5p / 1p	
130	CNC	T5	2b 4b / / 5p 3p / 1p	
210	Xong I/O của T4	T5	2b 4p / / 5p 3p / 1p	
230	CNC	T4	4p 2b / / 3p 5p / 1p	
250	I/O của T4, CNC	T3	2b 4b / / 3p 5p / 1p	

260	Xong I/O của T2	T3	2p 4b / / 3p 5p / 1p	
350	CNC	T2	2p 4b / / 5p 3p / 1p	T3 xong
360	I/O của T2, CNC	T5	4b 2b / / 5p / 1p	
430	Xong I/O của T4	T5	2b / / 5p / 1p	T4 xong
460	CNC	T5	2b / / 5p / 1p	
560	CNC	T1	2b / / / 1p	T5 xong
610	Xong I/O của T2	T1	2p / / 5p / 1p	
660	CNC	T2	2p / / / 1p	
670	I/O của T2, CNC	T1	2b / / / 1p	
770	CNC	T1	2b / / / 1p	
870	CNC	Không có	2b / / /	T1 xong
920	Xong I/O của T2	Không có	Trống	T2 xong

3.4 Hãy nêu ưu khuyết điểm của giải thuật lập lịch round-robin. Nạn đói có thể xảy ra không? Vì sao?

Ưu : giải quyết khuyết điểm của giải thuật round-robin

Khuyết : cách giải quyết còn dở vì nó gây ra nạn đói. Cụ thể, process có quyền thấp phải chờ (vô tận) các process mới có quyền cao hơn mình được kích hoạt đều đều theo thời gian.

Lập lịch với quyền ưu tiên động (thay đổi theo thời gian và tính cách chạy process)

Trong giải thuật này, mỗi process được cấp 1 quyền ưu tiên xác định lúc bắt đầu chạy, nhưng sẽ bị thay đổi theo thời gian tùy tính cách chạy của nó như thế nào.

Giả sử ta dùng chiến lược giảm quyền ưu tiên cho các process dài. Khi process được kích hoạt, nó có quyền cao nhất, mỗi khi chạy được 1 lần, nó bị giảm 1 mức ưu tiên. Số mức ưu tiên là không hạn chế.

3.5 Miêu tả chi tiết khoảng thời gian 900ms đầu tiên của hệ thống khi dùng giải thuật lập lịch dùng quyền ưu tiên động được miêu tả ở trên với các process sau:

	Time CPU	Chu kỳ kích hoạt
T1	15 ms	Sau mỗi 150 ms
T2	200 ms	Sau mỗi 300 ms
T3	1000 ms	-

Ta giả sử lúc đầu, hàng chờ có 3 process T1, T2, T3, ở mỗi chu kỳ 300ms thì process loại T1 kích hoạt ngay trước process loại T2. Quantum = 100ms.

Time	Sự kiện	Process được chọn	Trạng thái hàng chờ
0	Nạp chạy 1.0	1.0	1.0 2.0 3.0
15	CNC	2.0	2.0 3.0
115	CNC	3.0	3.0 / 2.0
150	Process 1.1 chạy	3.0	1.1 / 2.0

215	CNC	1.1	1.1 / 2.0 3.0
230	CNC	2.0	/ 2.0 3.0
300	1.2 & 2.1 chạy	2.0	1.2 2.1 / 2.0 3.0
330	CNC	1.2	1.2 2.1 / 3.0
345	CNC	2.1	2.1 / 3.0
445	CNC	3.0	/ 3.0 2.1
450	1.3 chạy	3.0	1.3 / 3.0 2.1
545	CNC	1.3	1.3 / 2.1 / 3.0
560	CNC	2.1	/ 2.1 / 3.0
600	1.4 & 2.2 chạy	2.1	1.4 2.2 / 2.1 / 3.0
660	CNC	1.4	1.4 2.2 / / 3.0
675	CNC	2.2	2.2 / / 3.0
750	1.5 chạy	2.2	2.2 1.5 / / 3.0
775	CNC	1.5	1.5 / 2.2 / 3.0
790	CNC	2.2	/ 2.2 / 3.0
890	CNC	3.0	/ / 3.0