

Thuật toán thay trang

Bài chữa trên lớp

Đề bài:

Cho tiến trình có 5 trang: 1,2,3,4,5. OS cấp cho 3 frame (tối đa có 3 frame trong bộ nhớ tại một thời điểm)

Các trang xuất hiện theo trình tự: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

Dùng thuật toán (FIFO, OPT, LRU) để xác định số lỗi trong quá trình phân trang

Note: Số lỗi đúng, nhưng bảng sai → ko có điểm.

Giải – Thuật toán FIFO = thay cái ở lâu nhất:

Lúc đầu tiên OS cấp cho tiến trình n frame (VD: 3 frame)

• Bước 1:

Tiến trình cần trang 1 trong bộ nhớ, nhưng các khung đều trống (ko có trang cần trong các khung được cấp) → lỗi, đánh F vào dòng FAULT (bước 1) → sửa lỗi bằng cách, tải trang 1 vào frame (VD: vào frame 1)

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1												
	F2												
	F3												
	FAULT	F											

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1											
	F2												
	F3												
	FAULT	F											

• Bước 2:

frame 1 vẫn ở đó, cần trang 2 trong bộ nhớ, nhưng chưa có → lỗi, đánh F vào dòng FAULT (bước 2) → Tải trang 2 vào frame (VD: vào frame 2)

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1										
	F2		2										
	F3												
	FAULT	F	F										

• Bước 3:

Tương tự (note: đánh lỗi F vào FAULT bước 3 trước khi sửa lỗi)

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1									
	F2		2	2									
	F3			3									
	FAULT	F	F	F	+								

• Bước 4:

Tương tự (note: đánh lỗi F vào FAULT bước 4 trước khi sửa lỗi).

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	+								
	F2		2	2									
	F3			3									
	FAULT	F	F	F	F								

Sửa lỗi đưa page 4 vào bộ nhớ, thay page 4 vào một trang đang có trong bộ nhớ.

Xác định trang 1,2,3 xem cái nào ở bộ nhớ lâu nhất → đưa nó ra, thay page 4 vào.

Ở đây thay page 4 vào vị trí page 1, còn những cái khác ko bị thay nên giữ nguyên

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	4								
	F2		2	2	+								
	F3			3									
	FAULT	F	F	F	F								

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	4								
	F2		2	2	2								
	F3			3	3								
	FAULT	F	F	F	F	+							

• Bước 5:

Cái dở của FIFO là vừa thay page 1 ra thì bước sau lại phải nạp lại page 1 → Báo lỗi → sửa bằng cách thay page 2 (do page 2 đã ở lâu nhất) bằng page 1.

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	4	4							
	F2		2	2	2	1							
	F3			3	3	3	+						
	FAULT	F	F	F	F	F	F						

• Bước 6:

Lại cần trang 2 → Báo lỗi → Sửa bằng cách thay page 3 bằng page 2 (do page 3 đã ở lâu nhất)

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	4	4	4						
	F2		2	2	2	1	1						
	F3			3	3	3	2						
	FAULT	F	F	F	F	F	F	+					

• Bước 7:

Tương tự, có lỗi

FIFO		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	4	4	4	5	5				
	F2		2	2	2	1	1	1	+				
	F3			3	3	3	2	2					
	FAULT	F	F	F	F	F	F	F					

- Bước 8, 9:

Ko có lỗi → Để trống (Ko ghi F và giữ nguyên)

- Bước 10:

Có lỗi → Ghi F và sửa lỗi

FIFO			1	2	3	4	1	2	5	1	2	3	4	5
		F1	1	1	1	4	4	4	5	5	5			
		F2		2	2	2	1	1	1	1	1			
		F3			3	3	3	2	2	2	2			
		FAULT	F	F	F	F	F	F	F			F		

- Bước 11:

Có lỗi → Ghi F và sửa lỗi

FIFO			1	2	3	4	1	2	5	1	2	3	4	5
		F1	1	1	1	4	4	4	5	5	5	5		
		F2		2	2	2	1	1	1	1	1	3		
		F3			3	3	3	2	2	2	2	2		
		FAULT	F	F	F	F	F	F	F			F	F	

- Bước 12:

Tương tự, ko có lỗi

FIFO			1	2	3	4	1	2	5	1	2	3	4	5
		F1	1	1	1	4	4	4	5	5	5	5	5	5
		F2		2	2	2	1	1	1	1	1	3	3	3
		F3			3	3	3	2	2	2	2	2	4	4
		FAULT	F	F	F	F	F	F	F			F	F	

- Kết luận:

9 lỗi

FIFO			1	2	3	4	1	2	5	1	2	3	4	5
9 LỖI		F1	1	1	1	4	4	4	5	5	5	5	5	5
		F2		2	2	2	1	1	1	1	1	3	3	3
		F3			3	3	3	2	2	2	2	2	4	4
		FAULT	F	F	F	F	F	F	F			F	F	

Giải – Thuật toán OPT = từ điểm xét ở dòng trên cùng, nhìn về bên phải → thay cái xa nhất, đang có trong bộ nhớ:

Lúc đầu tiên OS cấp cho tiến trình n frame (VD: 3 frame)

- Bước 1 - 4:

Ba bước đầu tiên giống như trên, sang bước 4 vẫn bị lỗi → điền F trước

OPTIMAL			1	2	3	4	1	2	5	1	2	3	4	5
		F1	1	1	1									
		F2		2	2									
		F3			3									
		FAULT	F	F	F	F								

Áp dụng quy tắc: từ điểm xét → nhìn thấy 3 là xa nhất → thay 3 bằng 4.

OPTIMAL		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	1	1							
	F2		2	2	2	2							
	F3			3	4								
	FAULT	F	F	F	F								

- Bước 5 - 7:

Tiếp đến bước 5, 6 ko bị lỗi. Nhưng sang bước 7 bị lỗi.

OPTIMAL		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	1	1	1						
	F2		2	2	2	2	2						
	F3			3	4	4	4						
	FAULT	F	F	F	F			F					

Áp dụng quy tắc: từ điểm xét → nhìn thấy 4 là xa nhất → thay 4 bằng 5.

OPTIMAL		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	1	1	1	1	1	1			
	F2		2	2	2	2	2	2	2	2			
	F3			3	4	4	4	5	5				
	FAULT	F	F	F	F			F			F		

- Bước 8 - 10:

Tiếp đến bước 8, 9 ko bị lỗi. Nhưng sang bước 10 bị lỗi.

Dùng quy tắc: từ điểm xét → ko thấy 1, 2 (hay nói khác đi 1, 2 ko xuất hiện nữa, hay nó là ở rất xa, xa vô cùng) → **Quy định**, thay cái có giá trị bé nhất (thay 1 bằng 3)

OPTIMAL		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	1	1	1	1	1	1	3		
	F2		2	2	2	2	2	2	2	2	2		
	F3			3	4	4	4	5	5	5	5		
	FAULT	F	F	F	F			F			F		

- Bước 11:

Bị lỗi, điền F → ko thấy 2, 3. **Quy định**, thay cái có giá trị bé nhất (thay 4 bằng 2)

OPTIMAL		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	1	1	1	1	1	1	3	3	
	F2		2	2	2	2	2	2	2	2	2	4	
	F3			3	4	4	4	5	5	5	5	5	
	FAULT	F	F	F	F			F			F	F	

- Bước 12:

Ko lỗi.

- Kết luận:

7 lỗi

OPTIMAL		1	2	3	4	1	2	5	1	2	3	4	5
7 LỖI		1	1	1	1	1	1	1	1	1	3	3	3
	F2		2	2	2	2	2	2	2	2	2	4	4
	F3			3	4	4	4	5	5	5	5	5	5
	FAULT	F	F	F	F			F			F	F	

Giải – Thuật toán LRU = từ điểm xét ở dòng trên cùng, nhìn về bên trái → thay cái xa nhất, đang có trong bộ nhớ:

Lúc đầu tiên OS cấp cho tiến trình n frame (VD: 3 frame)

- Bước 1 - 4:

Ba bước đầu tiên giống như trên, sang bước 4 vẫn bị lỗi → điền F trước

OPTIMAL		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1									
	F2		2	2									
	F3			3									
	FAULT	F	F	F	F								

Áp dụng quy tắc: từ điểm xét → nhìn thấy 1 là xa nhất → thay 1 bằng 4.

LRU		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	4								
	F2		2	2	2								
	F3			3	3								
	FAULT	F	F	F	F	F							

- Bước 5:

Lỗi → điền F

Áp dụng quy tắc: từ điểm xét (là 1) → nhìn thấy 2 là xa nhất (điểm gần 1 nhất về phía tay trái là 4, đến 3, rồi 2) → thay 2 bằng 1.

LRU		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	4	4							
	F2		2	2	2	1							
	F3			3	3	3							
	FAULT	F	F	F	F	F	F						

- Bước 6 - 7:

Lỗi → điền F

Áp dụng quy tắc: từ điểm xét (là 2) → nhìn thấy 3 là xa nhất (điểm gần 2 nhất về phía tay trái là 1, đến 4, rồi 3) → thay 2 bằng 3. Làm tiếp với điểm xét là 5

LRU		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	3	4	4	4	5					
	F2		2	2	2	1	1	1					
	F3			3	3	3	2	2					
	FAULT	F	F	F	F	F	F	F					

- Bước 8 - 9:

Tiếp tục với điểm xét lần lượt là 1, 2. Ko lỗi

LRU		1	2	3	4	1	2	5	1	2	3	4	5
	F1	1	1	1	4	4	4	5	5	5			
	F2		2	2	2	1	1	1	1	1			
	F3			3	3	3	2	2	2	2			
	FAULT	F	F	F	F	F	F	F			F		

- Bước 10-12:

Điểm xét là 3, 4, 5. Lỗi → điền F

• Kết luận:

10 lỗi

LRU			1	2	3	4	1	2	5	1	2	3	4	5
10 LỖI		F1	1	1	1	4	4	4	5	5	5	3	3	3
		F2		2	2	2	1	1	1	1	1	1	4	4
		F3			3	3	3	2	2	2	2	2	2	5
		FAULT	F	F	F	F	F	F	F			F	F	F

Note:

- Thuật toán FIFO, LRU chỉ có 1 bảng đúng
- Thuật toán OPT có nhiều bảng, nhưng theo quy định → chỉ có 1 bảng đúng (tính điểm theo quy định)
- LRU nhìn về quá khứ, OPT nhìn về tương lai
- OPT có ít lỗi nhất
- Số lỗi đúng, nhưng bảng sai → ko có điểm.

Bài chữa trên lớp – SV tự làm (1)

Đề bài:

Cho tiến trình có 5 trang: 1,2,3,4,5. OS cấp cho 4 frame.

Các trang xuất hiện theo trình tự: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

Dùng thuật toán (FIFO, OPT, LRU) để xác định số lỗi trong quá trình phân trang

FIFO: 10 lỗi			1	2	3	4	1	2	5	1	2	3	4	5
		Frame 1	1	1	1	1	1	1	5	5	5	5	4	4
		Frame 2		2	2	2	2	2	2	1	1	1	1	5
		Frame 3			3	3	3	3	3	3	2	2	2	2
		Frame 4				4	4	4	4	4	4	3	3	3
		Faults	x	x	x	x			x	x	x	x	x	x
OPT: 6 lỗi			1	2	3	4	1	2	5	1	2	3	4	5
		Frame 1	1	1	1	1	1	1	1	1	1	1	4	4
		Frame 2		2	2	2	2	2	2	2	2	2	2	2
		Frame 3			3	3	3	3	3	3	3	3	3	3
		Frame 4				4	4	4	5	5	5	5	5	5
		Faults	x	x	x	x			x				x	
LRU: 8 lỗi			1	2	3	4	1	2	5	1	2	3	4	5
		Frame 1	1	1	1	1	1	1	1	1	1	1	1	5
		Frame 2		2	2	2	2	2	2	2	2	2	2	2
		Frame 3			3	3	3	3	5	5	5	5	4	4
		Frame 4				4	4	4	4	4	4	3	3	3
		Faults	x	x	x	x			x			x	x	x

Bài chữa trên lớp – SV tự làm (2) (tùy chọn 1 trong 2)

Đề bài:

Cho tiến trình có 7 trang: 1,2,3,4,5,6,7. OS cấp cho 4 frame.

Các trang xuất hiện theo trình tự: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

Dùng thuật toán (FIFO, OPT, LRU) để xác định số lỗi trong quá trình phân trang

FIFO: 14 lỗi			1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
	Frame 1	1	1	1	1	1	1	1	5	5	5	5	5	3	3	3	3	3	1	1	1	1
	Frame 2		2	2	2	2	2	2	2	6	6	6	6	6	7	7	7	7	7	7	3	3
	Frame 3			3	3	3	3	3	3	3	2	2	2	2	2	6	6	6	6	6	6	6
	Frame 4				4	4	4	4	4	4	1	1	1	1	1	1	2	2	2	2	2	2
	Faults	x	x	x	x				x	x	x	x		x	x	x		x	x		x	

OPT: 8 lỗi			1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
	Frame 1	1	1	1	1	1	1	1	1	1	1	1	1	1	7	7	7	7	1	1	1	1
	Frame 2		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	Frame 3			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	Frame 4				4	4	4	4	5	6	6	6	6	6	6	6	6	6	6	6	6	6
	Faults	x	x	x	x				x	x					x				x			

LRU: 10 lỗi			1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
	Frame 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6
	Frame 2		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	Frame 3			3	3	3	3	3	5	5	5	5	5	3	3	3	3	3	3	3	3	3
	Frame 4				4	4	4	4	4	6	6	6	6	6	7	7	7	7	1	1	1	1
	Faults	x	x	x	x				x	x				x	x	x			x			

Bài tập SV làm nộp

Đề bài

Một chương trình có không gian địa chỉ dài 7 trang. Hệ điều hành quản lý bộ nhớ ảo phân trang theo yêu cầu. Bộ nhớ được cấp phát cho chương trình gồm 3 khung trang. Giả sử {1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 6, 7} là chuỗi truy cập bộ nhớ của chương trình. Hãy xác định chuỗi trạng thái các trang trong bộ nhớ và số lượng lỗi trang ứng với các phương pháp FIFO, tối ưu OPT và LRU?