



Bộ nhớ ảo

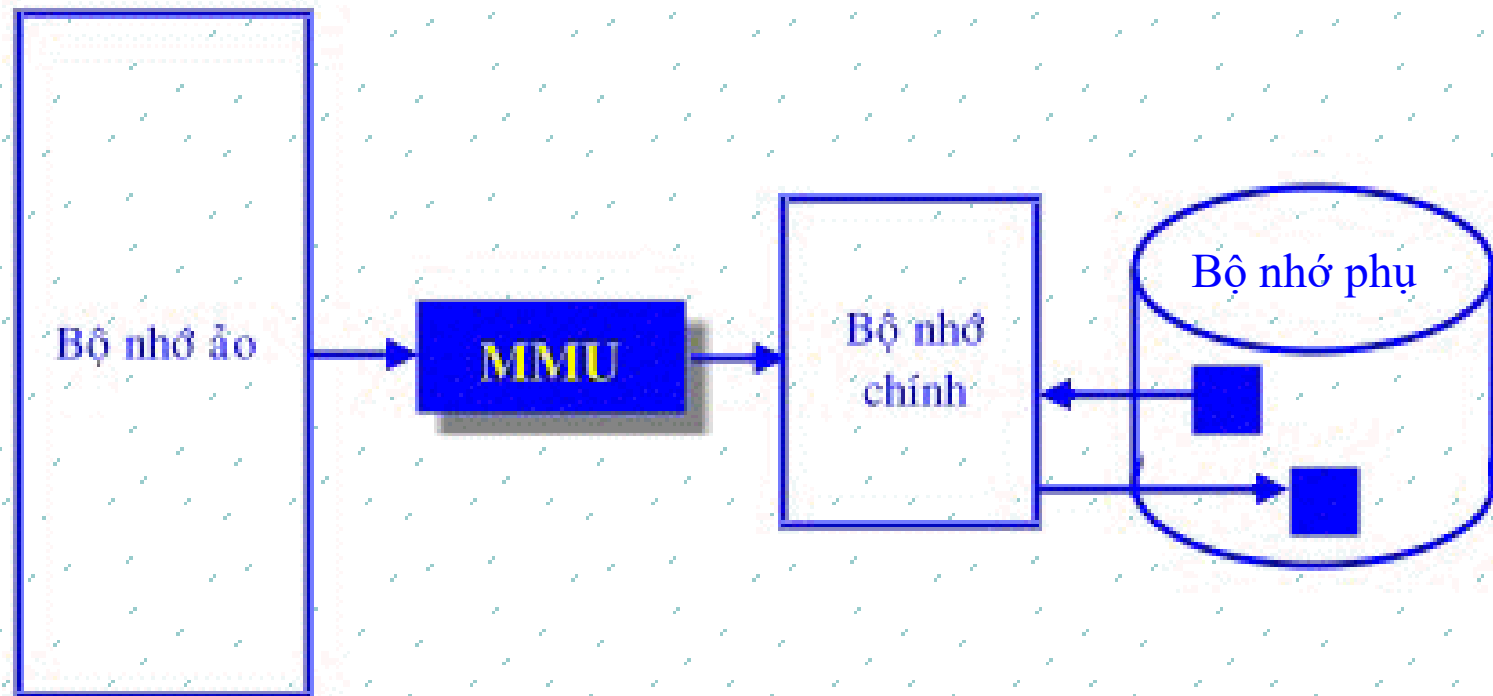
Cơ sở thiết lập bộ nhớ ảo

- Sự phân biệt rạch ròi giữa không gian địa chỉ luận lý và không gian địa chỉ vật lý.
- Một chương trình lớn không nhất thiết phải được nạp toàn bộ vào bộ nhớ rồi mới thực thi.
- Có thể dùng các thiết bị lưu trữ ngoài (đĩa cứng) để chứa tạm các đoạn tiến trình chưa dùng đến

Mục tiêu thiết lập bộ nhớ ảo

- Làm cho lập trình viên không cần quan tâm đến bộ nhớ vật lý (vốn khác nhau giữa các máy).
- “Mở rộng” dung lượng bộ nhớ vật lý.
- Tận dụng bộ nhớ phụ trong việc xử lý tiến trình.

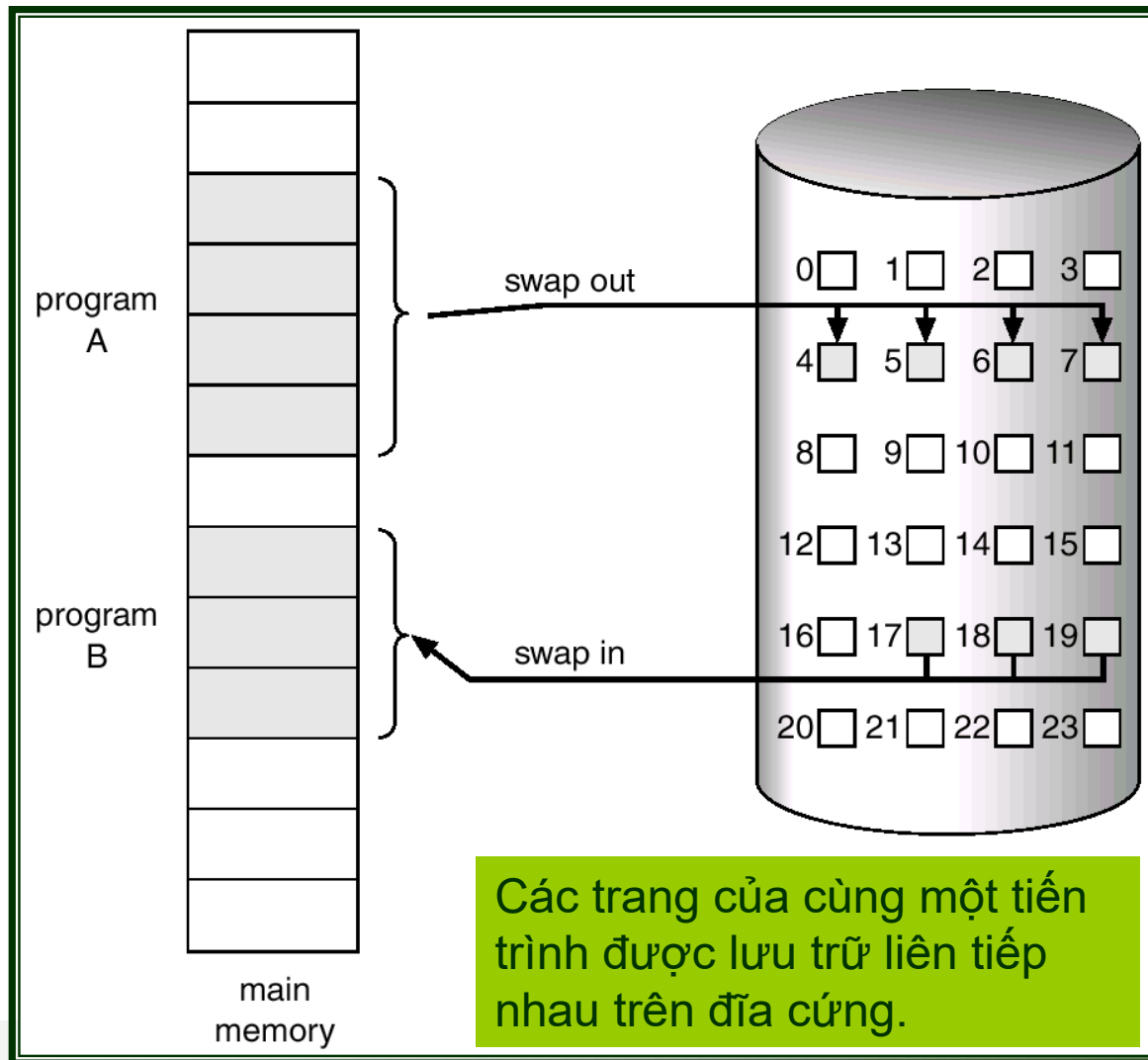
Cơ chế của bộ nhớ ảo



Bộ nhớ ảo được xây dựng dựa trên 2 cơ chế:

- Phân trang theo yêu cầu.
- Phân đoạn theo yêu cầu.

Cơ chế của bộ nhớ ảo



Cơ chế phần cứng

Một bit đặc biệt trong bảng trang cho biết trang đã nạp vào khung tương ứng trong bộ nhớ vật lý chưa.



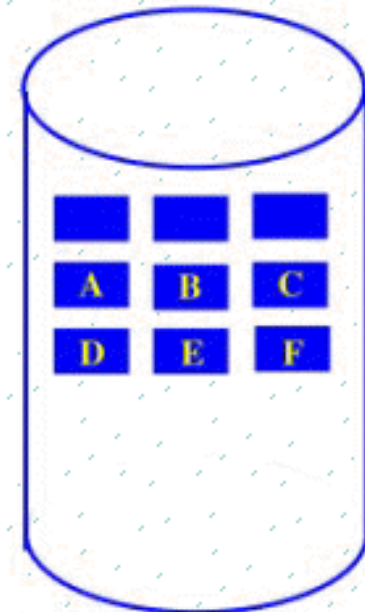
Bộ nhớ ảo



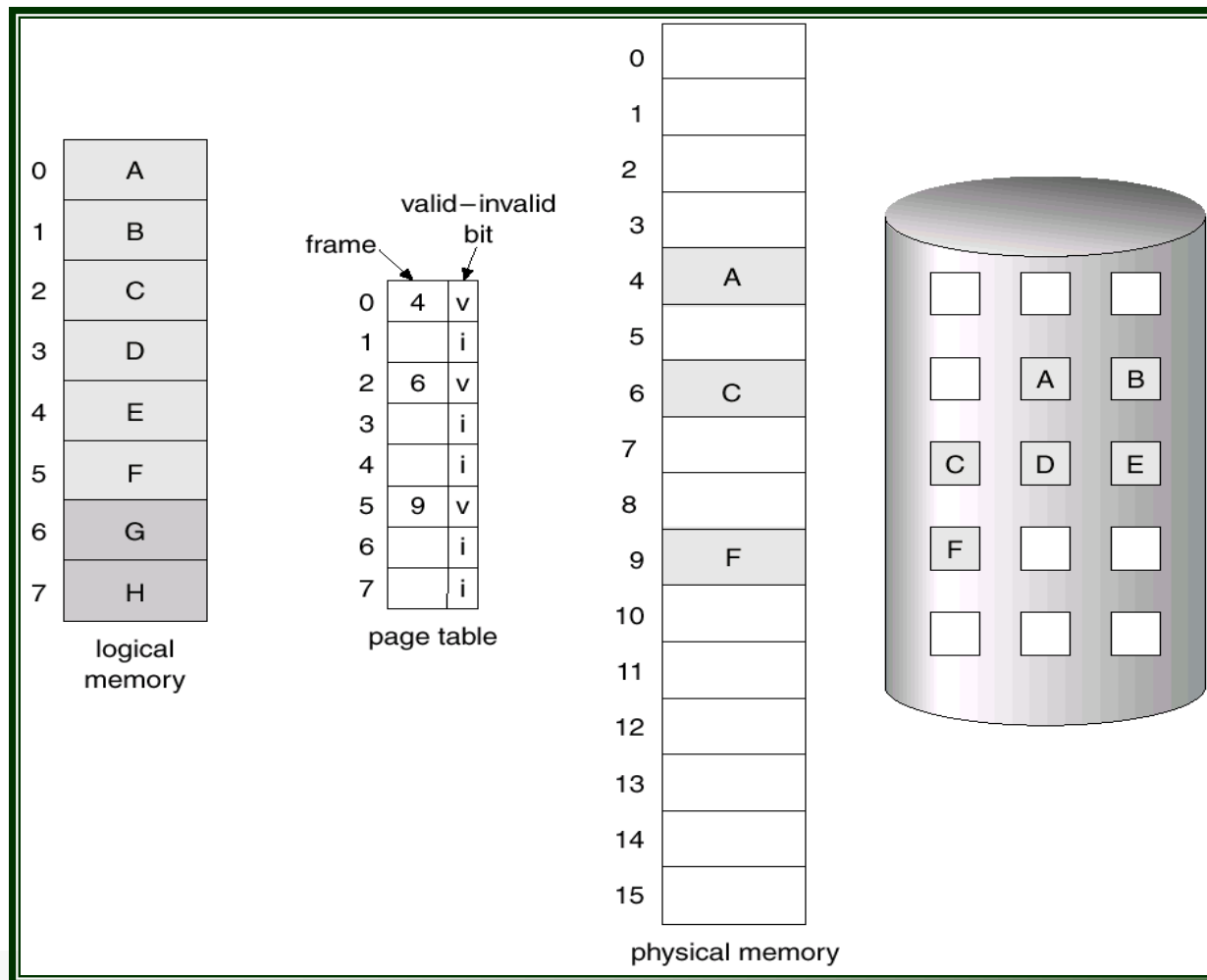
Bảng trang



Bộ nhớ vật lý



Cơ chế phần cứng



Lỗi trang (page fault)

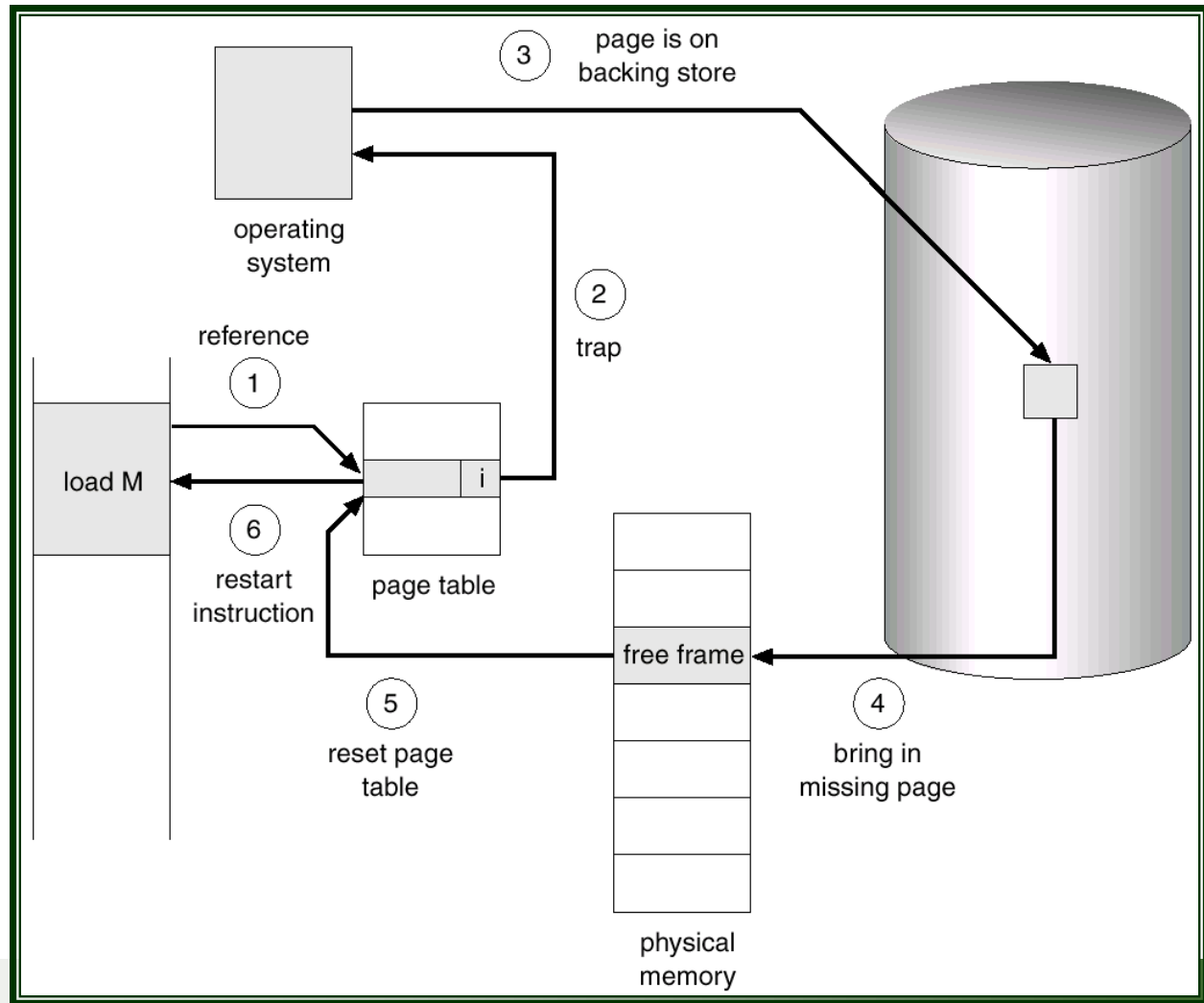
- Khi CPU truy xuất đến trang không có sẵn trong bộ nhớ vật lý -> lỗi trang. 2 lý do gây lỗi trang:
 - Truy xuất đến một địa chỉ không hợp lệ -> kết thúc tiến trình và báo lỗi.
 - Địa chỉ hợp lệ nhưng trang chưa sẵn sàng -> tạm dừng tiến trình và cập nhật trang tương ứng

Xử lý lỗi trang

- Nếu lỗi trang có nguyên nhân thuộc trường hợp thứ 2:
 - Tìm trang đang truy xuất trên đĩa cứng.
 - Tìm khung trống trên bộ nhớ chính, nếu có nạp trang từ đĩa cứng lên khung đó.
 - Nếu không còn khung trống, chọn một trang nào đó để “hy sinh”.
 - Tiếp tục tiến trình.

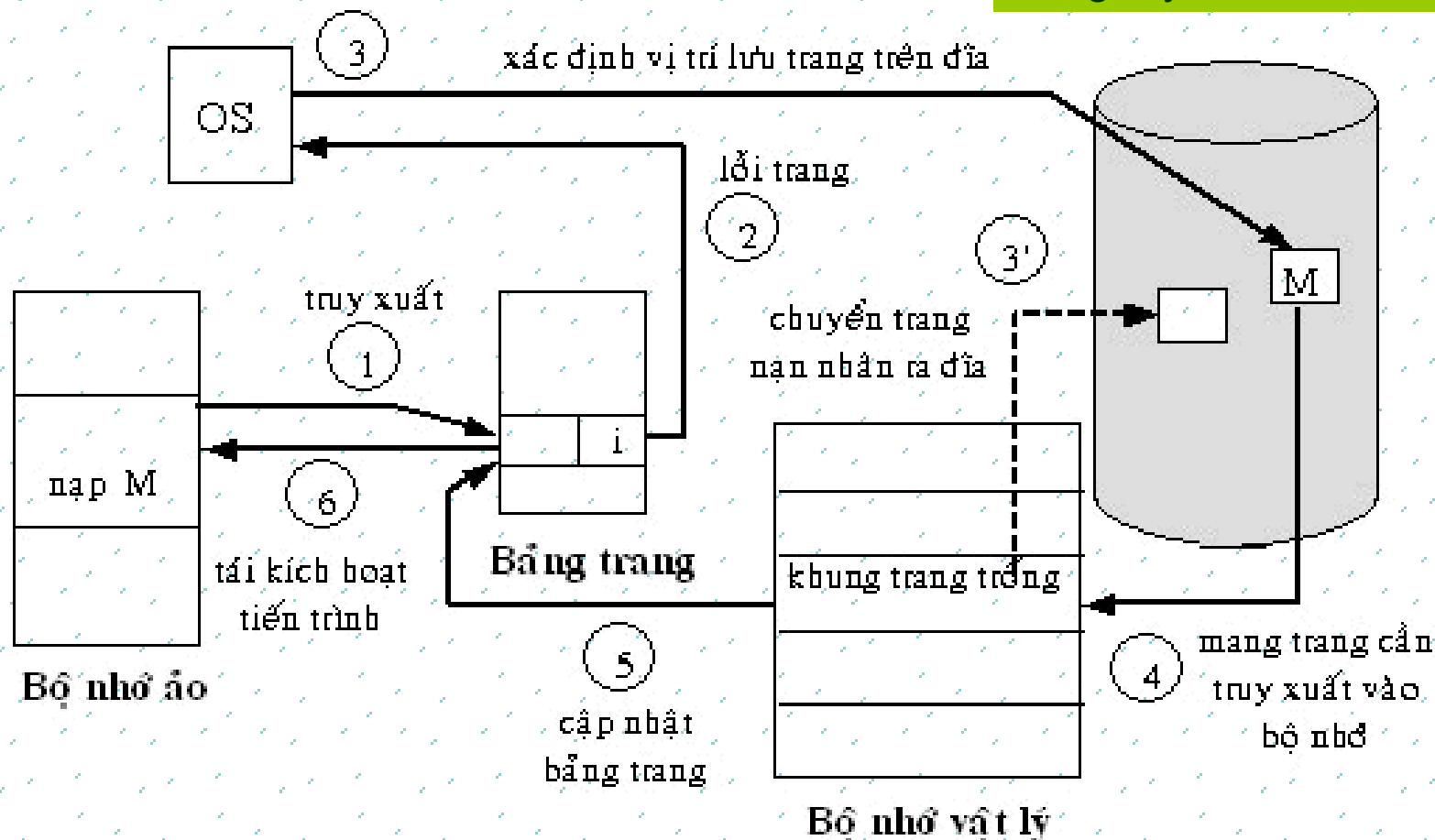
Xử lý lỗi trang

Trường hợp
có sẵn khung
trống trong bộ
nhớ vật lý.



Xử lý lỗi trang

Trường hợp không có khung trống, phải chọn trang “hy sinh”



Dirty bit

- Khi chọn một trang “nạn nhân” để thay thế, nếu trang đó chưa bị thay đổi gì thì chỉ đơn giản xóa sự hiện diện của nó trong bộ nhớ vật lý.
- Nếu trang đã có thay đổi thì cần lưu lại bản mới nhất.
- Dirty bit dùng để đánh dấu trang có bị thay đổi không.

Hiệu suất của cơ chế phân trang

- Gọi p là tỉ lệ lỗi trang ($0 \leq p \leq 1$)
- EAT (Effective Access Time) =
 $(1 - p) * MA +$
 $p * \text{page fault overhead} +$
 $[\text{swap out}] +$
 $\text{swap in} +$
 restart overhead

Cần duy trì tỉ lệ lỗi trang thấp

Các thuật toán thay thế trang

- Tìm trang thích hợp nhất để làm “nạn nhân” (victim).
- Biểu diễn chuỗi truy xuất:
 - Thứ tự địa chỉ truy xuất: 0100, 0432, 0101, 0162, 0102, 0103, 0104, 0101, 0611, 0102, 0103, 0104, 0101, 0610, 0102, 0103, 0104, 0101, 0609, 0102, 0105.
 - Giả sử kích thước trang là 100 byte, thứ tự truy xuất trang ???

Các thuật toán thay thế trang

- Thuật toán FIFO
- Chiến lược thay thế tối ưu
- Thuật toán LRU (Least Recently Used)
- Thuật toán xấp xỉ LRU

Thuật toán FIFO

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
	0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
		1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1
*	*	*	*		*	*	*	*	*	*			*	*			*	*	*

Ghi chú : * : có lỗi trang

Trang nào được nạp vào trước nhất sẽ được chọn làm nạn nhân

Thuật toán FIFO

Sử dụng 3 khung trang, sẽ có 9 lỗi trang phát sinh

Nghịch lý Belady

1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	4	4	4	5	5	5	5	5	5
	2	2	2	1	1	1	1	1	3	3	3
		3	3	3	2	2	2	2	2	4	4
*	*	*	*	*	*	*			*	*	

Sử dụng 4 khung trang, sẽ có 10 lỗi trang phát sinh

1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1	1	1	5	5	5	5	4	4
	2	2	2	2	2	2	1	1	1	1	5
		3	3	3	3	3	3	2	2	2	2
			4	4	4	4	4	4	3	3	3
*	*	*	*			*	*	*	*	*	*



Chiến lược thay thế tối ưu

- Trang nào lâu cần đến nhất thì thay thế!

Thuật toán LRU

sử dụng 3 khung trang, khởi đầu đều trống:

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	2	2	2	4	4	4	0	0	0	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	0	3	3	3	3	3	3	0	0	0	0	0
		1	1	1	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7
*	*	*	*		*		*	*	*	*			*		*		*		

Dựa vào thời điểm truy xuất gần nhất của từng trang

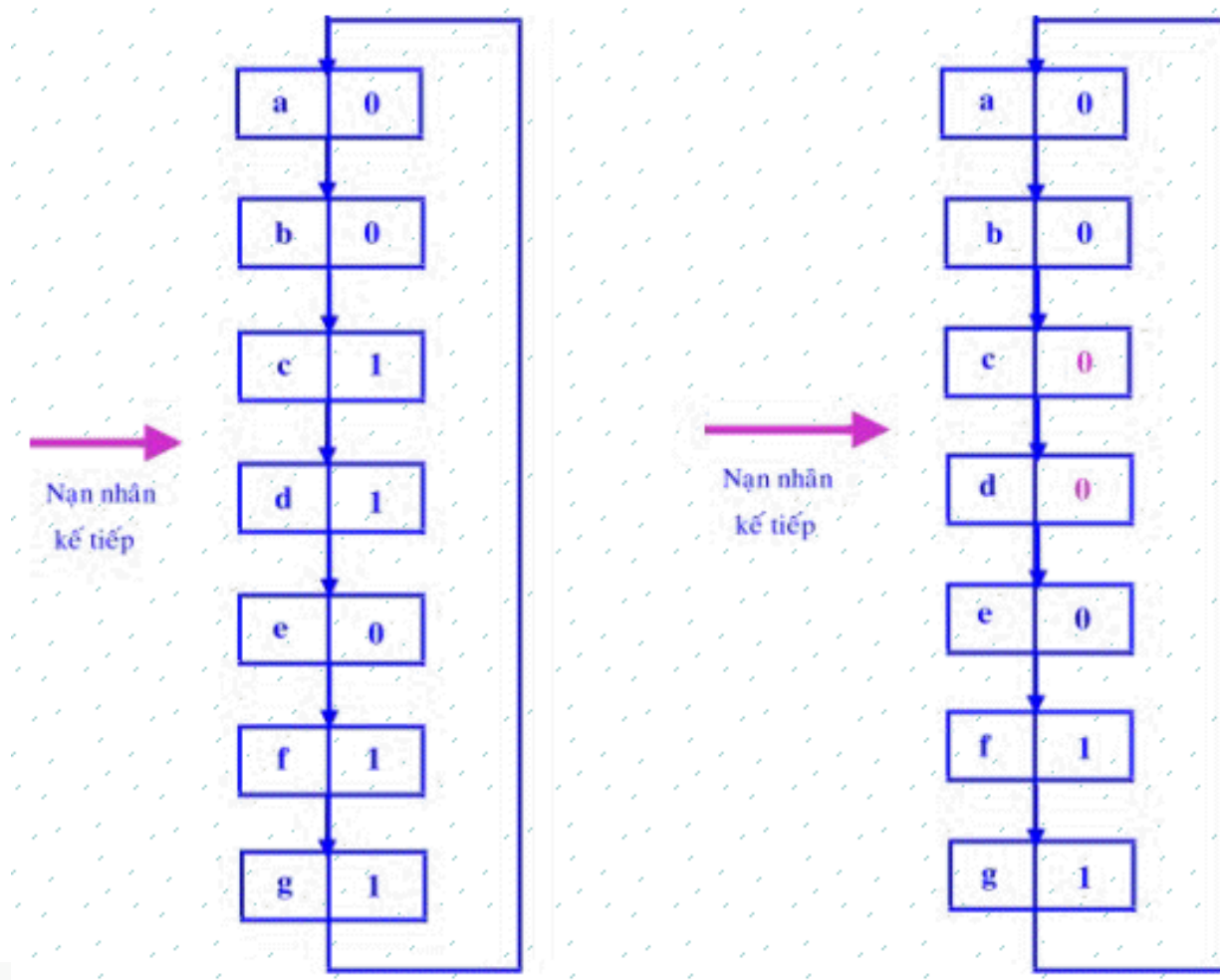
Các thuật toán xấp xỉ LRU

- Dùng bit reference để xác định trang đã được truy xuất chưa.
- Tạo thêm “cơ hội thứ 2”
- Thuật toán NRU (Not Recently Used)

Thuật toán cơ hội thứ 2

- Nếu giá trị của bit reference là 0, thay thế trang đã chọn.
- Ngược lại, cho trang này một cơ hội thứ hai, và chọn trang tiếp theo theo thuật toán FIFO.
- Khi một trang được cho cơ hội thứ hai, giá trị của bit reference được đặt lại là 0, và thời điểm vào Ready List được cập nhật lại là thời điểm hiện tại.

Thuật toán cơ hội thứ 2



Thuật toán NRU

- Kết hợp bit reference và bit dirty:
 - Lớp 1: (0,0) đây là trang tốt nhất để thay thế.
 - Lớp 2: (0,1)
 - Lớp 3: (1,0)
 - Lớp 4: (1,1)
- Lớp 1 có độ ưu tiên thấp nhất và lớp 4 có độ ưu tiên cao nhất.

Bài tập 1

Giả sử có một chuỗi truy xuất bộ nhớ có chiều dài p với n số hiệu trang khác nhau xuất hiện trong chuỗi. Giả sử hệ thống sử dụng m khung (khởi động trống). Với một thuật toán thay thế trang bất kỳ :

- Cho biết số lượng tối thiểu các lỗi trang xảy ra ?
- Cho biết số lượng tối đa các lỗi trang xảy ra ?

Bài tập 2

- Một máy tính 32-bit địa chỉ, sử dụng một bảng trang hai cấp. Địa chỉ ảo được phân bổ như sau : 9 bit dành cho bảng trang cấp 1, 11 bit cho bảng trang cấp 2, còn lại dành cho offset. Cho biết kích thước một trang trong hệ thống, và không gian địa chỉ ảo có bao nhiêu trang ?

Bài tập 3

Giả sử có một hệ thống sử dụng kỹ thuật phân trang theo yêu cầu. Bảng trang được lưu trữ trong các thanh ghi. Để xử lý một lỗi trang tốn 8 ms nếu có sẵn một khung trang trống, hoặc trang bị thay thế không bị sửa đổi nội dung, và tốn 20 ms nếu trang bị thay thế bị sửa đổi nội dung. Mỗi truy xuất bộ nhớ tốn 100 ns. Giả sử trang bị thay thế có xác suất bị sửa đổi là 70%. Tỷ lệ phát sinh lỗi trang phải là bao nhiêu để có thể duy trì thời gian truy xuất bộ nhớ (effective access time) không vượt quá 200 ns ?

Bài tập 4

- Một máy tính có 4 khung trang. Thời điểm nạp, thời điểm truy cập cuối cùng, và các bit reference (R), modify (M) của mỗi trang trong bộ nhớ được cho trong bảng sau :

Trang nào sẽ được chọn thay thế theo :

- a) thuật toán NRU
- b) thuật toán FIFO
- c) thuật toán LRU
- d) thuật toán "cơ hội thứ 2"

Trang	Nạp	Truy cập cuối	R	M
0	126	279	0	0
1	230	260	1	0
2	120	272	1	1
3	160	280	1	1

Bài tập 5

■ Xét chuỗi truy xuất bộ nhớ sau:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1
, 2, 3, 6

Có bao nhiêu lỗi trang xảy ra khi sử dụng các thuật toán thay thế sau đây, giả sử có 3 và 4 khung trang ?

a) LRU

b) FIFO

c) Cơ hội thứ hai