Lý thuyết hệ điều hành

Giảng viên: TS. Hà Chí Trung

Bộ môn: Khoa học máy tính

Khoa: Công nghệ thông tin

Học viện Kỹ thuật quân sự

Email: hct2009@yahoo.com

Mobile: 01685.582.102

Chương 3. Quản lý bộ nhớ

3.11. Bộ nhớ ảo

3.11.1. Khái niệm về bộ nhớ ảo

3.11.2. Các thuật toán thay thế trang

3.11.3. Thrashing và nguyên lý cục bộ

Chương 3. Quản lý bộ nhớ

3.11. Bộ nhớ ảo

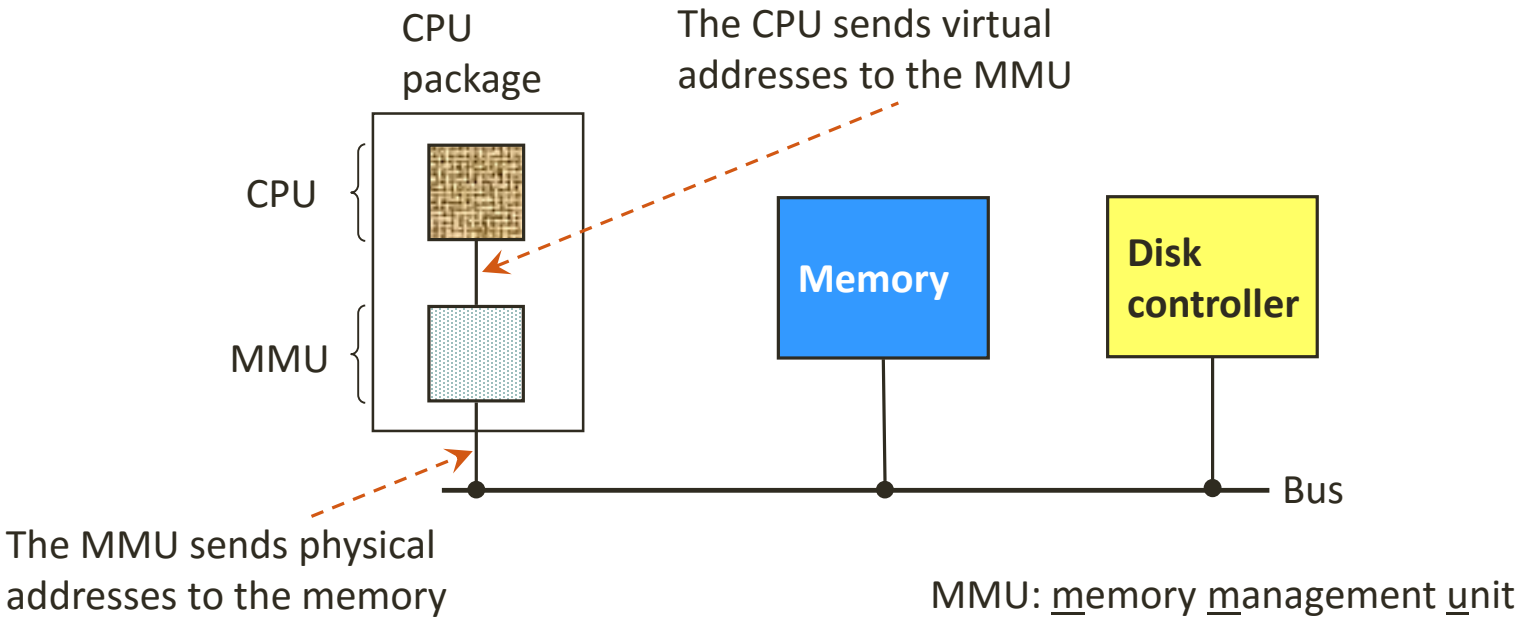
3.11.1. Khái niệm về bộ nhớ ảo

3.11.2. Các thuật toán thay thế trang

3.11.3. Thrashing và nguyên lý cục bộ

3.11.1 Khái niệm về bộ nhớ ảo (VM)

- Trong quá trình tiến trình chạy, các tham chiếu đến bộ nhớ (logical address) được chuyển thành địa chỉ thực (physical address)



- Tiến trình gồm các trang hay đoạn được nạp vào không liên tục trong bộ nhớ

3.11.1 Khái niệm về bộ nhớ ảo (VM)

- Các phần của một tiến trình không nhất thiết phải nạp vào bộ nhớ tại cùng một thời điểm, VD:
  - Đoạn mã điều khiển các lỗi hiếm khi xảy ra
  - Các arrays, list, tables được cấp phát bộ nhớ (cấp phát tĩnh) nhiều hơn yêu cầu cần thiết
  - Một số tính năng ít khi được dùng của một chương trình
- Không gian VM: bao gồm MM và bộ nhớ thứ cấp (SM). Thông thường SM tham gia vào VM được lưu trữ ở một vùng đặc biệt gọi là không gian hoán đổi (swap space). VD: swap partition trong Unix/Linux, file pagefile.sys trong W2K/XP

3.11.1 Khái niệm về bộ nhớ ảo (VM)

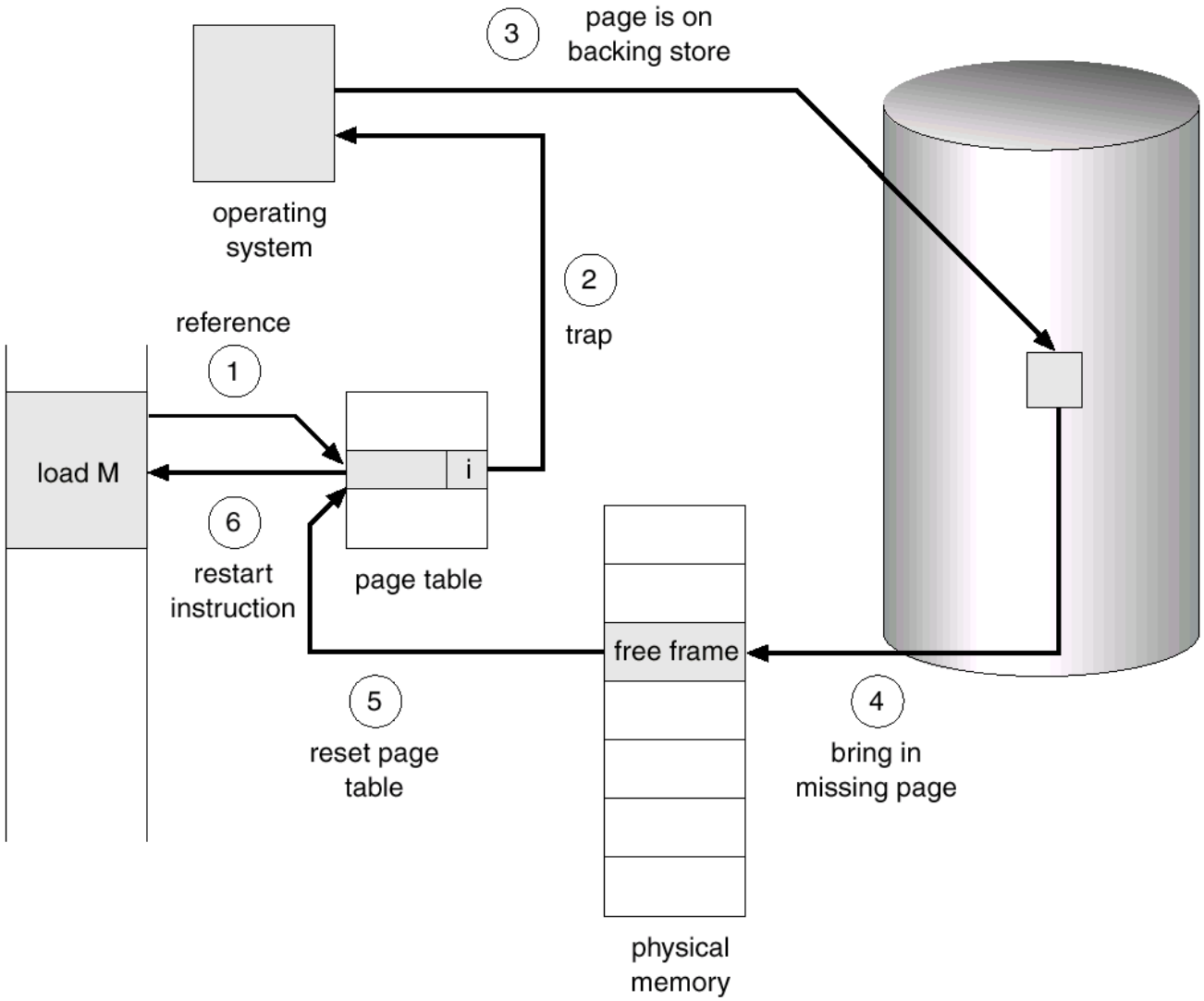
- Các page table có thêm các trường bit trạng thái:
  - **Present bit**: = 1 nếu trang được yêu cầu hợp lệ và có trong bộ nhớ, ngược lại = 0;
  - **Modified bit**: = 1 báo hiệu trang bị thay đổi kể từ khi được nạp vào bộ nhớ

- Yêu cầu phân trang (**Demand Paging**): Các trang chỉ được nạp vào bộ nhớ khi được yêu cầu.

3.11.1 Khái niệm về bộ nhớ ảo (VM)

- Khi có một lệnh tham chiếu đến phần chương trình chưa có trong bộ nhớ chính, page-fault service routine (**PFSR**) của OS kích hoạt một ngắt mềm gọi là **memory fault** (**page fault, segmentation fault – lỗi trang**)
  1. chuyển tiến trình về trạng thái **blocked**
  2. Phát ra yêu cầu đọc đĩa để nạp phần chương trình được tham chiếu vào bộ nhớ chính. Trong khi đó, một tiến trình khác sẽ chiếm được CPU để thực thi.
  3. Sau khi đọc ghi đĩa hoàn tất, đĩa gây ra một ngắt mềm báo cho hệ điều hành để chuyển tiến trình tương ứng trở lại trạng thái sẵn sàng trong **ready queue.**

3.11.1 Khái niệm về bộ nhớ ảo (VM)



Chương 3. Quản lý bộ nhớ

3.11. Bộ nhớ ảo

3.11.1. Khái niệm về bộ nhớ ảo

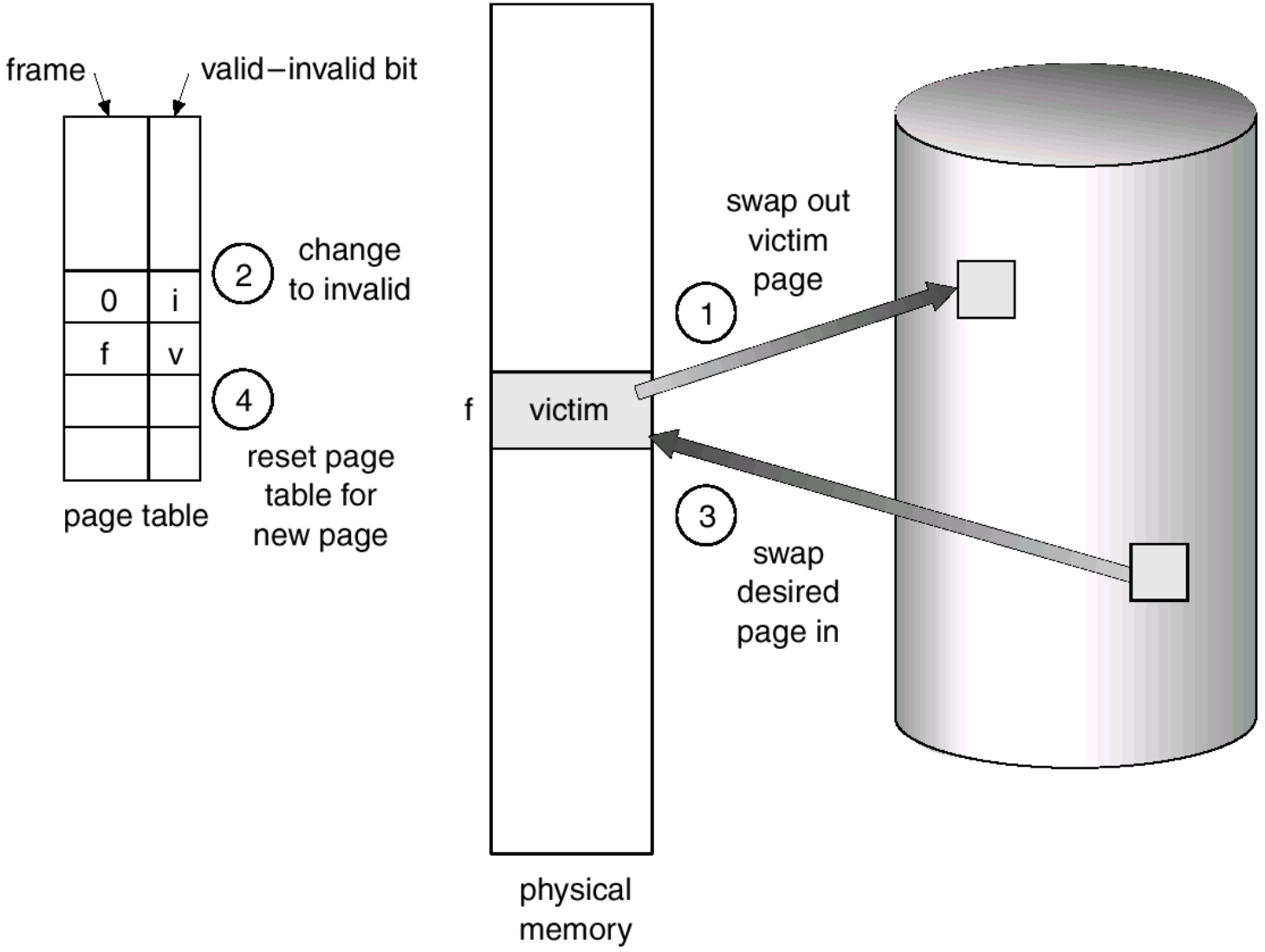
3.11.2. Các thuật toán thay thế trang

3.11.3. Thrashing và nguyên lý cục bộ

3.11.2 Các thuật toán thay thế trang

- Tại bước 2 của PFSR, để xử lý được cả trường hợp không tìm thấy frame còn trống:
  1. Xác định vị trí trên đĩa của page đang cần nạp
  2. Tìm 1 frame còn trống:
     a) Nếu tìm thấy thì dùng nó
     b) Nếu không, sử dụng thuật toán thay thế trang để tìm một trang hi sinh (victim page)
     c) Ghi victim page lên đĩa, cập nhật page table
  3. Đọc trang cần vào frame trống
  4. Cập nhật lại page table.

3.11.2 Các thuật toán thay thế trang



3.11.2 Các thuật toán thay thế trang

- Frame-allocation algorithm: Cấp cho mỗi tiến trình bao nhiêu frame?
- Page-replacement algorithm: Tối thiểu số page fault
- Nguyên tắc tối ưu: Chọn trang thay thế là trang không còn dùng nữa hoặc sẽ không dùng lại trong thời gian xa nhất
- Các tiêu chuẩn (thực tế) để chọn trang thay thế
  - Các trang không bị thay đổi
  - Các trang không bị khóa
  - Các trang không thuộc quá trình nhiều page fault
  - Các trang không thuộc tập làm việc của quá trình
  - Các giải thuật thay thế trang phụ thuộc vào resident set (số frame cấp cho mỗi process)

3.11.2 Các thuật toán thay thế trang

1. Giải thuật tối ưu (**MIN**) hay optimal (**OPT**)
2. Giải thuật thay thế trang "lâu nhất chưa sử dụng" **Least Recently Used** (**LRU**)
3. Giải thuật **FIFO**
4. Giải thuật thay thế trang FIFO cải tiến (**Second Chance**)
5. Giải thuật thay thế trang ít được sử dụng gần đây (**NRU – Not Recently Used**)

3.11.2 Các thuật toán thay thế trang

- Giải thuật tối ưu (**MIN**) hay optimal (**OPT**)
  - Thay thế trang nhớ được tham chiếu trễ nhất trong tương lai
  - Mỗi trang nhớ được gắn nhãn là một số có giá trị bằng số lệnh sẽ được thực thi trước khi tham chiếu đến trang đó. Các trang sẽ không được truy cập tiếp kề từ vị trí hiện thời sẽ có nhãn là vô cùng lớn.
  - Trang nhớ có nhãn lớn nhất sẽ bị thay thế.
  - Tối ưu số page faults
  - Không thể hiện thực được. Vì sao?

3.11.2 Các thuật toán thay thế trang

- Giải thuật tối ưu (**MIN**) hay optimal (**OPT**)
- VD: Thứ tự các trang nhớ được tham chiếu như sau và Giả sử resident set = 3 – số frame của mỗi tiến trình được hiện thực dạng quay vòng
- Chú ý: 2 cột gần cuối cùng: - Thay trang 3 vào vị trí trang 2 vì trang 2 ko được truy cập nữa.

Thời điểm t	0	1	2	3	4	5	6	7	8	9	10	11
	1	2	3	4	1	2	5	1	2	3	4	5
Bộ nhớ thực có 3 frames	1	1	1	1	1	1	1	1	1	1	1	1
		2	2	2	2	2	2	2	2	3	4	4
			3	4	4	4	5	5	5	5	5	5

7 page fault

3.11.2 Các thuật toán thay thế trang

- Giải thuật thay thế trang "lâu nhất chưa sử dụng" Least Recently Used (**LRU**)
  - Trong bảng phân trang, mỗi trang được ghi nhận thời điểm được tham chiếu.
  - Chọn trang thay thế là trang đã không được tham khảo trong thời gian lâu nhất

Thôøi ñieåm t	0	1	2	3	4	5	6	7	8	9	10	11
Chuoãi tham khaûo	1	2	3	4	1	2	5	1	2	3	4	5
Boänhôù thöïc coù 3 frame	1	1	1	4	4	4	5	5	5	3	3	5
		2	2	2	1	1	1	1	1	1	4	4
			3	3	3	2	2	2	2	2	2	2

3.11.2 Các thuật toán thay thế trang

- Giải thuật **FIFO**
  - Xem các frame được cấp phát cho process như là circular buffer
  - Trang nhớ cũ nhất sẽ được thay thế: first-in, first-out

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

Boänhôù thöïc coù 3 frame

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

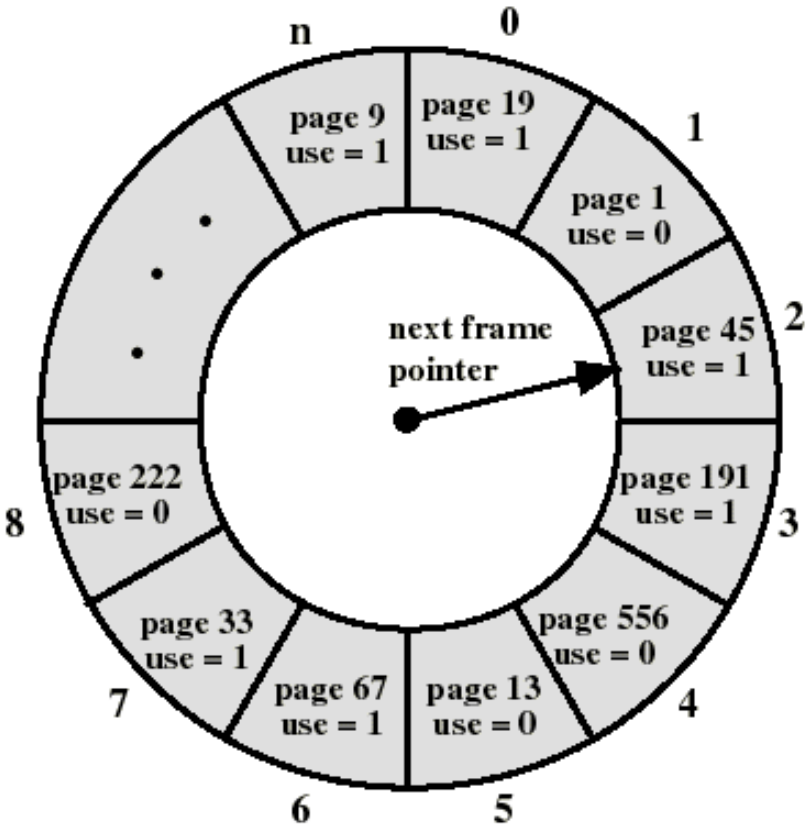
9 page fault

3.11.2 Các thuật toán thay thế trang

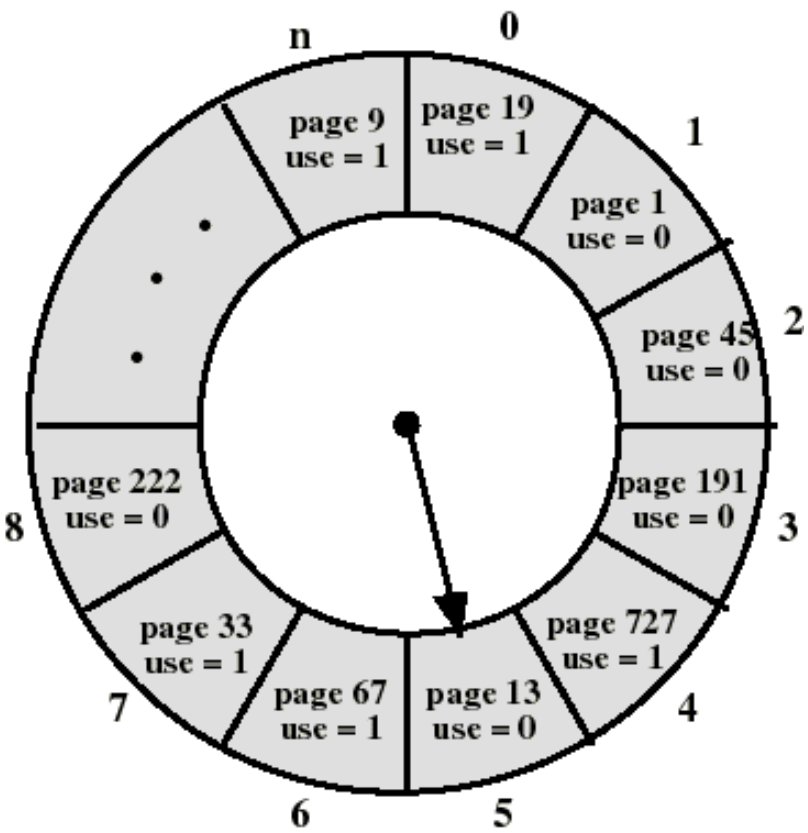
- Giải thuật thay thế trang FIFO cải tiến (**Second Chance**) còn gọi là giải thuật đồng hồ (clock) vì dùng bộ đệm quay vòng giống đồng hồ. Tương tự FIFO
  1. Khi một trang được thay thế, con trỏ sẽ chỉ đến frame kế tiếp trong bộ đệm quay vòng
  2. Mỗi frame có một use-bit. Bit này được thiết lập trị 1 khi
     - Trang nhớ được nạp lần đầu vào frame
     - Có tham chiếu tới địa chỉ thuộc trang chứa trong frame
  3. Trang được chọn xét thay thế theo kiểu FIFO
     - Khi cần thay thế một trang nhớ, trang nhớ nằm trên frame đầu tiên có use bit bằng 0 sẽ được thay thế.
     - Trong suốt quá trình tìm trang nhớ thay thế, giải thuật clock sẽ reset về giá trị 0 các use-bit của frame trên đường đi qua.

3.11.2 Các thuật toán thay thế trang

- VD: tham chiếu trang 727



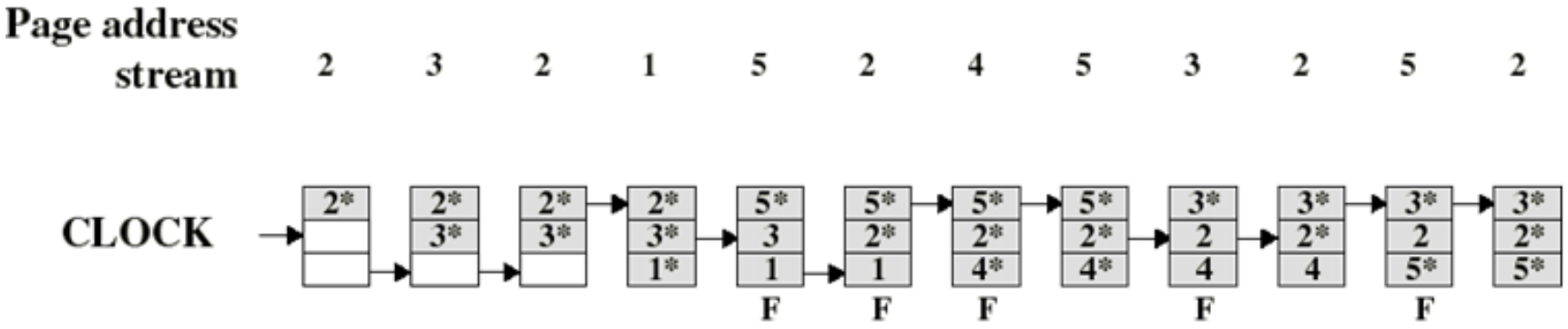
(a) State of buffer just prior to a page replacement



(b) State of buffer just after the next page replacement

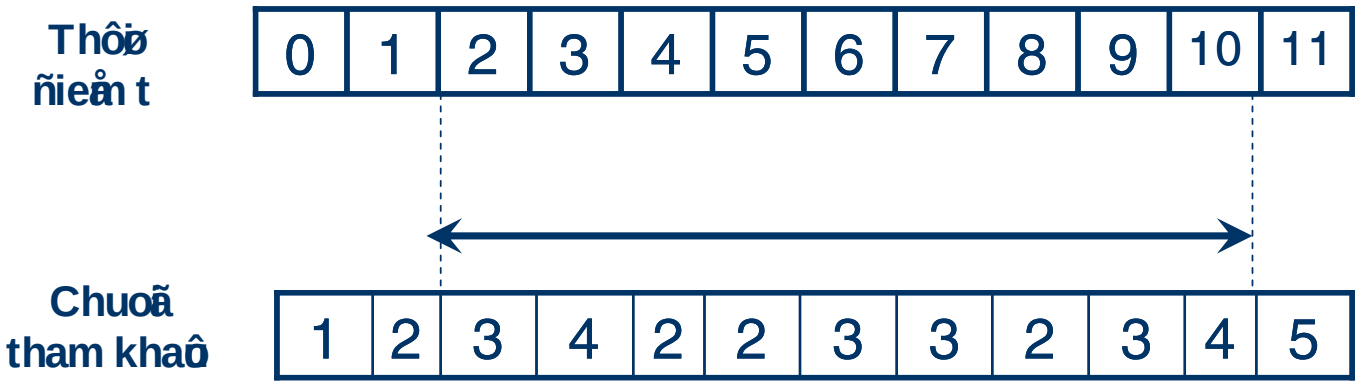
3.11.2 Các thuật toán thay thế trang

- Dấu *: use bit tương ứng được thiết lập trị 1
- Giải thuật Clock bảo vệ các trang thường được tham chiếu bằng cách thiết lập use bit bằng 1 với mỗi lần tham chiếu
- Một số kết quả thực nghiệm cho thấy clock có hiệu suất gần với LRU



3.11.2 Các thuật toán thay thế trang

- **Giải thuật thay thế trang ít được sử dụng gần đây (NRU - Not Recently Used)** Là giải thuật xấp xỉ LRU
- Chọn trang thay thế là trang có tần suất được tham khảo là nhỏ nhất trong 1 khoảng thời gian nhất định
- Tại t=11, nếu trong bộ nhớ còn 3 trang 2, 3, 4 ta sẽ chọn trang 4 để thay thế



3.11.2 Các thuật toán thay thế trang

- **Giải thuật thay thế trang ít được sử dụng gần đây (NRU - Not Recently Used)**
  - Mỗi mục trong page table có thêm 2 bit là M (modified) và R (referenced: read, write)
  - Khởi đầu: R = M = 0
  - Khi trang nhớ được tham chiếu thì thiết lập R = 1
  - Khi có thay đổi nội dung trang nhớ thì thiết lập M = 1
  - Khi có page fault xảy ra, hệ điều hành xem xét tất cả trang nhớ và chia thành 4 loại dựa trên giá trị của R và M
  - Loại 1: không tham chiếu (R=0), không cập nhật (M=0)
  - Loại 2: không tham chiếu (R=0), có cập nhật (M=1)
  - Loại 3: có tham chiếu (R=1), không cập nhật (M=0)
  - Loại 4: có tham chiếu (R=1), có cập nhật (M=1)
  - NRU sẽ thay trang nhớ đầu tiên ở loại nhỏ hơn trước.

Chương 3. Quản lý bộ nhớ

3.11. Bộ nhớ ảo

3.11.1. Khái niệm về bộ nhớ ảo

3.11.2. Các thuật toán thay thế trang

3.11.3. Thrashing và nguyên lý cục bộ

3.11.3. Thrashing và nguyên lý cục bộ

- **VM** hoạt động trên nguyên lý cục bộ (***locality principle***):
  - Cục bộ về thời gian (***temporal locality***): Các sự việc xảy ra ở thời điểm t rất có thể là đã hoặc sẽ xảy ra ở các thời điểm lân cận (t – dt, t + dt).
  - Cục bộ về không gian (***spatial locality***): Biến cố xảy ra ở một vùng nhớ rất có thể là đã hoặc sẽ xảy ra ở các vùng lân cận

  - **Ý nghĩa:** Nguyên lý cục bộ là cơ sở trong các giải thuật thay thế trang

3.11.3. Thrashing và nguyên lý cục bộ

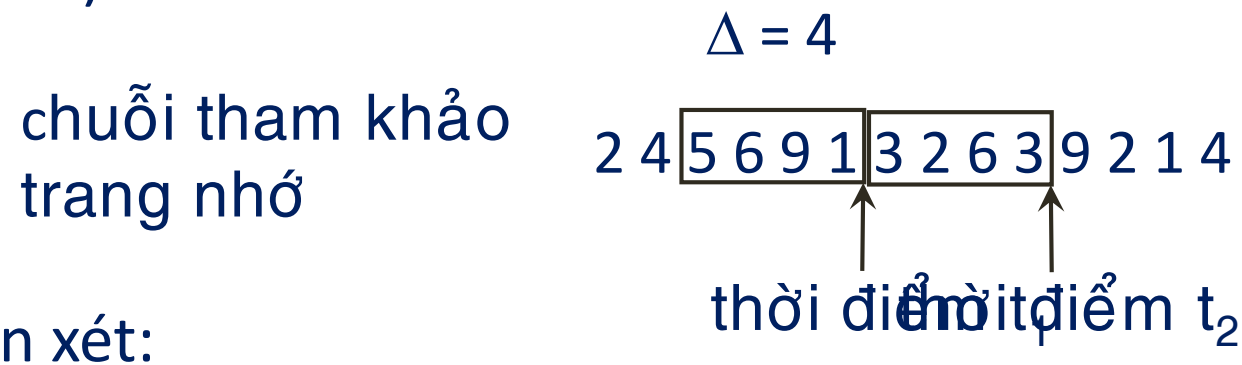
- Hiện tượng **thrashing**: các trang nhớ của 1 tiến trình bị vào ra liên tục. Làm giảm hiệu suất hoạt động của CPU.
  - VD: một vòng lặp N lần, mỗi lần tham chiếu đến địa chỉ nằm trong 4 trang nhớ, trong khi process chỉ được cấp 3 frame. Chuỗi tham chiếu trang: 012301230123
- Để tránh thrashing, OS phải cấp đủ frame cho process. Bao nhiêu frame là đủ?

3.11.3. Thrashing và nguyên lý cục bộ

- **Frame-allocation algorithm:**
- Cấp bao nhiêu:
  - Cấp ít: nhiều page fault
  - Cấp nhiều: giảm mức độ đa nhiệm (multiprogramming)
- Cấp phát tĩnh (fixed allocation): số frame cấp cho mỗi tiến trình không đổi:
  - Cấp bằng nhau
  - Cấp theo tỉ lệ;
- Cấp phát động: Thay đổi khi tiến trình chạy, OS phải mất thêm chi phí tính toán, VD:
  - Tỉ lệ page fault cao, tăng số frame, ngược lại giảm.

3.11.3. Thrashing và nguyên lý cục bộ

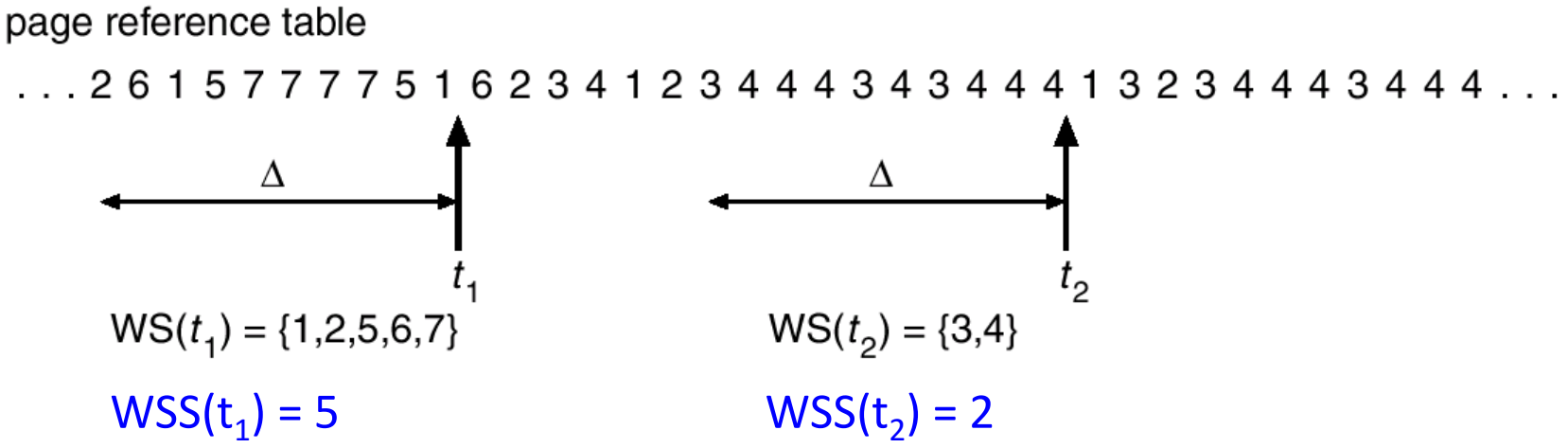
- Working set model: dựa trên nguyên lý cục bộ.
- Δ: working set window: số lượng các tham chiếu trang nhớ gần nhất cần được tham khảo
- Working set WS_i (tập làm việc của tiến trình P_i): là tập các số trang trong working set window (tại 1 thời điểm t nào đó)



- Nhận xét:
  - Δ nhỏ, không bao phủ lân cận, ngược lại bao trùm lên nhiều lân cận

3.11.3. Thrashing và nguyên lý cục bộ

- VD: Δ = 10



- WSS_i là kích thước của working set của P_i: WSS_i = số lượng các trang trong WS_i
- Đặt D = $\sum WSS_i$ = tổng các working-set size của mọi process trong hệ thống.
- Rõ ràng nếu D>m (số frame bộ nhớ) thì sẽ xảy ra thrashing

3.11.3. Thrashing và nguyên lý cục bộ

- **Giải pháp working set:**
  - Khi khởi tạo 1 tiến trình, cấp cho nó số frame thỏa mãn WSS của nó.
  - Nếu D>m thì đưa 1 tiến trình về trạng thái suspend
- **Giải pháp PFF** (Page-Faults Frequency): điều khiển số page-fault rate (số lỗi trang trong 1 giây):
  - page-fault rate thấp -> giảm số frame cho tiến trình
  - page-fault rate cao -> cấp bổ sung frame