LAB 3.2: BỘ NHỚ ẢO

(2312678 – Nguyễn Lê Bảo Long – CTK47A)

# Mục tiêu

Hiểu rõ các thuật toán thay thế trang:

* + LRU replacement
  + FIFO replacement
  + Optimal (OPT) replacement

# Nội dung

1. Áp dụng các thuật toán thay thế trang (1) FIFO, (2) LRU, and (3) optimal (OPT) với các chuỗi tham chiếu sau:
   1. 2, 6, 9, 2, 4, 2, 1, 7, 3, 0, 5, 2, 1, 2, 9, 5, 7, 3, 8, 5
   2. 0, 6, 3, 0, 2, 6, 3, 5, 2, 4, 1, 3, 0, 6, 1, 4, 2, 3, 5, 7
   3. 3, 1, 4, 2, 5, 4, 1, 3, 5, 2, 0, 1, 1, 0, 2, 3, 4, 5, 0, 1
   4. 4, 2, 1, 7, 9, 8, 3, 5, 2, 6, 8, 1, 0, 7, 2, 4, 1, 3, 5, 8
   5. 0, 1, 2, 3, 4, 4, 3, 2, 1, 0, 0, 1, 2, 3, 4, 4, 3, 2, 1, 0

Cho biết số lỗi trang xảy ra của mỗi thuật toán với 3 khung trang (frame) được cấp (Vẽ bảng minh họa thuật toán).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Chuỗi tham chiếu | FIFO | ‬LRU | ‬OPT | Số khung trang |
| 2, 6, 9, 2, 4, 2, 1, 7, 3, 0, 5, 2, 1, 2, 9, 5, 7, 3, 8, 5 | 10 | 5 | 3 | 3 |
| 0, 6, 3, 0, 2, 6, 3, 5, 2, 4, 1, 3, 0, 6, 1, 4, 2, 3, 5, 7 | 9 | 7 | 5 | 3 |
| 3, 1, 4, 2, 5, 4, 1, 3, 5, 2, 0, 1, 1, 0, 2, 3, 4, 5, 0, 1 | 10 | 6 | 4 | 3 |
| 4, 2, 1, 7, 9, 8, 3, 5, 2, 6, 8, 1, 0, 7, 2, 4, 1, 3, 5, 8 | 12 | 8 | 5 | 3 |
| 0, 1, 2, 3, 4, 4, 3, 2, 1, 0, 0, 1, 2, 3, 4, 4, 3, 2, 1, 0 | 11 | 6 | 3 | 3 |

1. Xét mảng hai chiều A:

int A[][] = new int[100][100];

Với A[0][0] được lưu trữ tại vị trí 200 trong bộ nhớ tổ chức theo kỹ thuật phân trang với kích thước trang là 200 (Bộ nhớ định địa chỉ theo byte, mỗi phần tử kiểu int chiếm 1 byte). Một tiến trình tính toán ma trận trên đang trong trang (bộ nhớ luận lý có 200 trang, đánh số từ 0 đến 199); như vậy mỗi lệnh thực thi của tiến trình này sẽ được nạp từ trang 0.

Cho 3 khung trang (frame), có bao nhiêu lỗi trang sẽ phát sinh khi khởi tạo mảng như đoạn code bên dưới? Biết hệ điều hành dùng thuật toán thay thế trang LRU; một khung trang trong 3 khung cấp cho process đã dành chứa page 0 (chứa code process); hai khung trang còn lại được khởi động trống.

* 1. for (int j = 0; j < 100; j++)

for (int i = 0; i < 100; i++) A[i][j] = 0;

Mảng và phân trang:

* Mảng A[100][100] chiếm 10.000 phần tử (mỗi int = 1 byte).
* Mỗi trang nhớ chứa 200 byte → cần 50 trang để lưu mảng (10.000 / 200 = 50).

Cách truy cập mảng (theo cột):

* Vòng lặp gán A[i][j] = 0 chạy theo cột:  
  A[0][0], A[1][0], ..., A[99][0], A[0][1], ..., A[99][99].
* Thứ tự truy cập trang:  
  1,1, 2,2, 3,3, ..., 50,50 (lặp lại 100 lần, mỗi lần cho 1 cột).

Thay thế trang (LRU với 3 khung trang):

* 1 khung luôn chứa trang 0 (mã chương trình).
* 2 khung còn lại dành cho dữ liệu mảng.
* Do 50 trang > 2 khung, mỗi lần truy cập trang mới sẽ gây lỗi trang (trừ lần truy cập liên tiếp cùng trang).
* Mỗi cột: 50 lỗi trang (mỗi trang mới bị đẩy ra do LRU).
* Tổng lỗi trang: 100 cột×50 lỗi=5000

Kết luận: Tổng số lỗi trang xảy ra: 5000.

* 1. for (int i = 0; i < 100; i++)

for (int j = 0; j < 100; j++) A[i][j] = 0;

Mảng và phân trang:

* Mảng A[100][100] chiếm 10.000 phần tử (mỗi int = 1 byte).
* Mỗi trang nhớ chứa 200 byte → cần 50 trang để lưu mảng (10.000 / 200 = 50).

Cách truy cập mảng (theo hàng):

* Vòng lặp gán A[i][j] = 0 chạy theo hàng:

A[0][0], A[0][1], ..., A[0][99],

A[1][0], ..., A[1][99],

...,

A[99][0], ..., A[99][99]

* Thứ tự truy cập trang:

Trang 1 (hàng 0-1): 100 lần truy cập liên tục (A[0][0] đến A[1][99])

Trang 2 (hàng 2-3): 100 lần truy cập liên tục

...

Trang 50 (hàng 98-99): 100 lần truy cập liên tục  
→ Trình tự: Trang 1 (100 lần), Trang 2 (100 lần), ..., Trang 50 (100 lần).

Thay thế trang (LRU với 3 khung trang):

* 1 khung luôn chứa trang 0 (mã chương trình).
* 2 khung còn lại dành cho dữ liệu mảng.
* Phân tích lỗi trang:

+ Khi bắt đầu:

Truy cập Trang 1: Lỗi trang → nạp vào khung 1.

Truy cập 99 phần tử còn lại của Trang 1: Không lỗi (đã có trong khung 1).

+ Tiếp theo:

Truy cập Trang 2: Lỗi trang → nạp vào khung 2.

Truy cập 99 phần tử còn lại của Trang 2: Không lỗi.

+ Tiếp theo:

Truy cập Trang 3: Lỗi trang → thay thế Trang 1 (LRU, vì Trang 2 vừa dùng gần nhất).

Truy cập 99 phần tử còn lại của Trang 3: Không lỗi.

+ Lặp lại quy trình cho đến hết 50 trang.

Tính toán lỗi trang:

* Mỗi trang mới gây 1 lỗi trang khi truy cập lần đầu.
* Tổng số trang dữ liệu: 50.
* Tổng lỗi trang: 50 (mỗi trang gây 1 lỗi khi được nạp lần đầu).

Kết luận: Tổng số lỗi trang xảy ra: 50.