Bài thực hành 6

Họ tên: Võ Anh Kiệt

MSSV: 20520605

Lớp: IT007.M21.ANTN

Bài làm

Câu hỏi chuẩn bị

Câu 1: Lỗi trang là gì? Khi nào xảy ra trường hợp lỗi trang? Vẽ lưu đồ mô tả cách hệ điều hành xử lý lỗi trang? Trình bày cách cài đặt Demand paging?

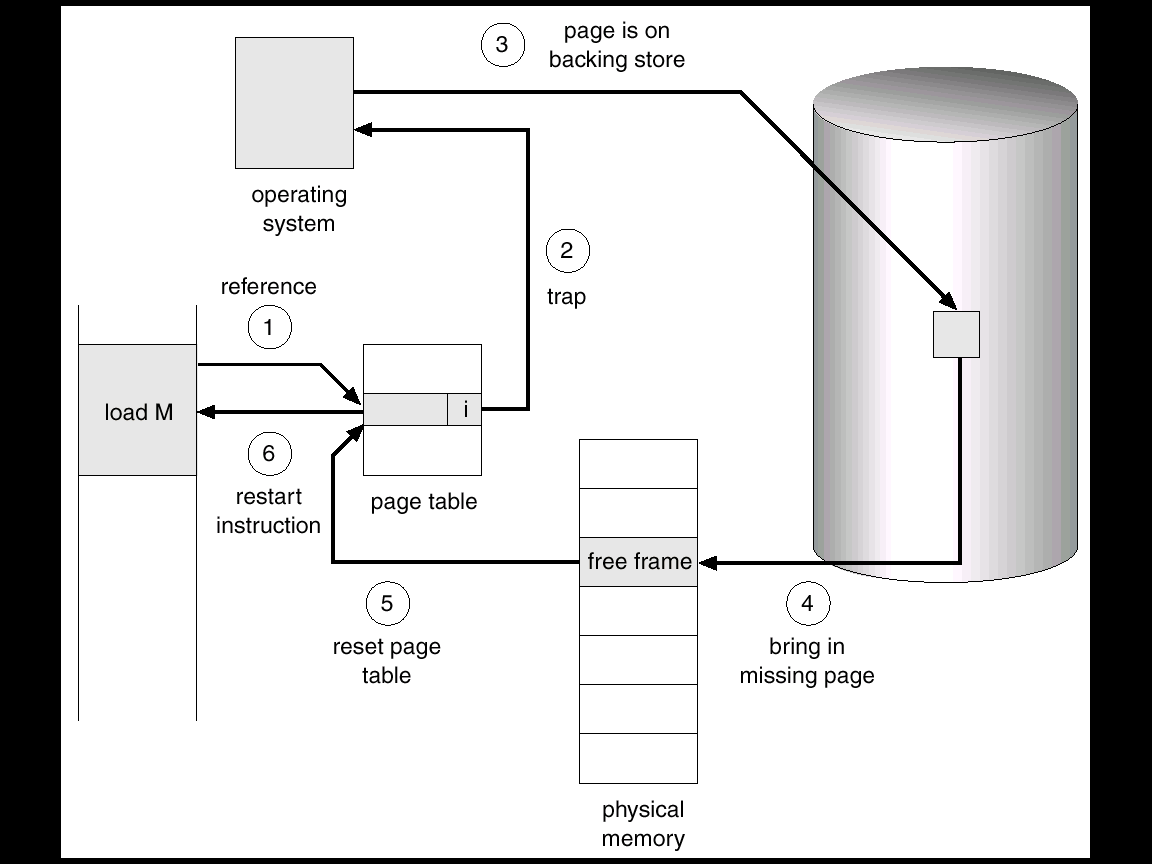
Lỗi trang:

Được định nghĩ là khi dò tìm các bảng trang để lấy thông tin cần thiết trong việc chuyển đổi địa chỉ, nếu thấy trang được yêu cầu truy vấn hoặc trang được đánh dấu không hợp lệ thì đó được xác định là lỗi trang.

Trường hợp:

Lỗi trang xảy ra khi người dùng truy vấn tới một trang không hợp lệ (trang không ở trong không gian địa chỉ của quá trình – không đang ở trong bộ nhớ chính) hoặc truy cập tới một trang hợp lệ nhưng đang ở bộ nhớ phụ.

Lưu đồ:



Demand paging:

Cơ chế demand paging là các trang của quá trình sẽ chỉ được nạp vào bộ nhớ chính khi được yêu cầu.

Khi có một tham chiếu đến một trang mà không có trong bộ nhớ chính thì phần cứng sẽ gây ra một ngắt, kích khởi page-fault service routine (PFSR) của hệ điều hành.

Quá trình 1 của PFSR

* Chuyển process về blocked
* Request đọc đĩa để nạp trang được tham chiếu vào một frame trống; trong khi đợi I/O, 1 process khác được cấp CPU để thực thi
* Sau khi I/O hoàn tất, đĩa gây ra một ngắt đến hệ điều hành; PFSR cập nhật page table và chuyển process về ready

Quá trình 2 cảu PFSR

* Indicate vị trí trên đĩa của trang đang cần
* Tìm một frame trống:
* Nếu có frame trống thì dùng
* Nếu không có frame trống thì dùng một giải thuật thay trang để chọn một trang nạn nhân.
* Ghi trang nạn nhân lên đĩa; cập nhật bảng trang và bảng frame tương ứng
* Đọc trang đang cần vào frame trống, cập nhật bảng trang và bảng frame tương ứng.

Câu 2: Tại sao phải thực hiện chiến lược thay thế trang?

Khi có lỗi trang xuất hiện hệ thống sẽ phải chọn một trang (trang nạn nhân) để loại khỏi bộ nhớ và nhường không gian cho trang đang được yêu cầu. Nếu trang được chọn đã có sửa đổi về mặt nội dung thì hệ thống phải cập nhật lại nội dung của trang này trên bộ nhớ phụ. Trang mới lấy từ bộ nhớ phụ chỉ việc ghi đè lên trang nạn nhân. Trang nạn nhân phải thường là trang phải là trang ít được khi sử dụng đến nhất. Vì nếu hệ thống chọn một trang được sử dụng rất nhiều làm trang nạn nhân thì tổng chi phí cho việc thay trang của hệ thống tăng lên rất nhiều. Vì vậy cần thực hiện chiến lược thay thế trang.

Câu 3: Vẽ lưu đồ thuật toán mô tả cách thức xử lý của 3 giải thuật thay thế trang: FIFO, OPT, LRU? Vẽ sơ đồ trình bày thay thế trang bằng 3 giải thuật trên với chuỗi tham chiếu:

0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Giải sử có 3 khung trang và các khung trang ban đầu là rỗng. Xác định số Page Fault từ đó đưa ra đánh giá các giải thuật

FIFO

Sơ đồ trình bày thế trang

A screenshot of a computer

Description automatically generated with medium confidence

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 1 | 6 | 4 | 0 | 1 | 0 | 3 | 1 | 2 | 1 |
| 0 | 0 | 0 | 6 | 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 |
|  | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 |
|  |  | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| \* | \* | \* | \* | \* | \* | \* |  | \* |  | \* |  |

Số lỗi trang là 9

Giải thuật OPT

Sơ đồ trình bày thế trang

A screenshot of a computer

Description automatically generated with medium confidence

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 1 | 6 | 4 | 0 | 1 | 0 | 3 | 1 | 2 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 2 |
|  | 2 | 2 | 6 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
|  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \* | \* | \* | \* | \* |  |  |  | \* |  | \* |  |

Số lỗi trang là 7

Giải thuật LRU

Sơ đồ trình bày thế trang

A screenshot of a computer

Description automatically generated with medium confidence

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 1 | 6 | 4 | 0 | 1 | 0 | 3 | 1 | 2 | 1 |
| 0 | 0 | 0 | 6 | 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 |
|  | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 |
|  |  | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| \* | \* | \* | \* | \* | \* | \* |  | \* |  | \* |  |

Số lỗi trang là 9

Hướng dẫn thực hành

Coding

FIFO



OTP

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

LRU

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

Check existed

Text

Description automatically generated

Get input

Text

Description automatically generated

Main function

Text

Description automatically generated

Text

Description automatically generated

Runcode

FIFO

A picture containing text

Description automatically generated

OPT

A picture containing text

Description automatically generated

LRU

A picture containing text

Description automatically generated

Bài tập ôn tập

Câu 1 Nghịch lý Belady là gì? Sử dụng chương trình đã viết trên để chứng minh nghịch lý này.

Khi số lượng khung trang được tăng lên thì số lượng lỗi trang cũng sẽ tăng lên. Đây được gọi là nghịch lý Belady

Ví dụ thực tế trên chương trình đã build

2 khung trang

Schematic

Description automatically generated with low confidence

3 khung trang

Schematic

Description automatically generated with medium confidence

4 khung trang

A picture containing schematic

Description automatically generated

Câu 2: Nhận xét về mức độ hiệu quả và tính khả thi của các giải thuật FIFO, OPT, LRU.

Giải thuật nào là bất khả thi nhất? Vì sao?

Giải thuật nào là phức tạp nhất? Vì sao?

Vấn đề chính khi thay thế trang là chọn lựa một trang nạn nhân để chuyển ra bộ nhớ phụ, cho dù ở thuật toán nào thì bài toán luôn quay về việc chọn trang nạn nhân để thay thế sao cho ít lỗi nhất có thể. Đánh giá hiệu quả bằng cách thực hiện bài toán xử lý trên một chuỗi các địa chỉ cần được truy vấn và tính số lỗi trang phát sinh.

FIFO: chọn các trang đã được nạp vào bộ nhớ có thời gian dài nhất để thay thế

Ưu điểm: thuật toán này dễ hiểu, dễ cài đặt.

Mặt hạn chế: trang được chọn để thay thế có thể là trang chứa nhiều dữ liệu cần thiết, được nạp sớm do thường xuyên được sử dụng, từ đó việc bị chuyển ra bộ nhớ phụ sẽ dễ gây ra lỗi trang cho những lần truy xuất sau.

Khi số lượng khung trang được tăng lên thì số lượng lỗi trang cũng sẽ tăng lên. Đây được gọi là nghịch lý Belady

OPT: chọn trang sẽ có thời gian được sử dụng chậm nhất trong tương lai để thay thế,

Ưu điểm: ý tưởng tốt, lỗi trang phát sinh là thấp nhất, nó cũng không gánh chịu nghịch lý Belady

Hạn chế: không khả thi trong thực tế, vì không thể biết trước chuỗi truy xuất của tiến trình trong tương lai

- LRU: thay thế trang có thời gian không được sử dụng lâu nhất trong bộ nhớ, dùng quá khứ gần để đoán tương lai, để biết thời điểm nạp vào

Ưu điểm: tối ưu thời điểm sẽ truy cập

Hạn chế: đòi hỏi phần cứng phải hỗ trợ khá nhiều: biến bộ đếm, stack

Giải thuật LRU bất khả thi nhất: gây nhiều lỗi và yêu cầu nhiều phần cứng

Giải thuật OPT phức tạp nhất: không thấy được số trang phát sinh trong tương lai