

MÔN HỌC: HỆ ĐIỀU HÀNH

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 8

1. Tại sao cần phải có bộ nhớ ảo?

Hiện nay, các chương trình ngày một phình to về kích thước. Vì vậy, việc load toàn bộ chương trình vào bộ nhớ chính để thực thi gần như là chuyện bất khả thi.

Bộ nhớ ảo là một kỹ thuật cho phép chạy một process, mà process đó không cần thiết phải nằm hoàn toàn trong bộ nhớ.

Với bộ nhớ ảo, chúng ta sẽ có thể làm được những điều sau :

- Thực thi một chương trình có kích thước lớn hơn nhiều so với bộ nhớ chính.
- Lượng process trong bộ nhớ cũng nhiều hơn.
- Trừu tượng hoá bộ nhớ chính thành một mảng array lớn. Người dùng chỉ thấy được bộ nhớ thông qua bộ nhớ luận lý (logical memory). Từ đó giúp người lập trình viên đỡ cơ cực hơn.
- Bộ nhớ ảo cũng giúp cho các tiến trình chia sẻ dữ liệu với nhau dễ dàng, ngoài ra còn hỗ trợ trong việc cài đặt shared memory (bộ nhớ dùng chung).

2. Có bao nhiêu kỹ thuật cài đặt bộ nhớ ảo? Mô tả sơ lược các kỹ thuật đó?

Có hai kỹ thuật :

- Phân trang theo yêu cầu (Demand Paging) : Các trang dữ liệu sẽ không được sao chép vào bộ nhớ chính cho đến khi nó được cần tới.
- Phân đoạn theo yêu cầu (Segmentation Paging) : Các segment của chương trình sẽ không được chép vào bộ nhớ chính cho đến khi nó được cần tới.
- Phần cứng memory management phải hỗ trợ paging, segmentation.
- OS phải quản lý swap giữa bộ nhớ chính và bộ nhớ thứ cấp.

3. Các bước thực hiện kỹ thuật phân trang theo yêu cầu?

- Phân trang theo yêu cầu :
 - Các trang của quá trình 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 (valid bit) thì phần cứng sẽ gây ra một ngắt (gọi là pagefault – các bạn hay gặp từ này khi làm bài tập về các giải thuật phân trang), kích khởi pagefault service routine (PFSR) của hệ điều hành.
 - PFSR :
 - Chuyển process về trạng thái blocked.
 - Phát ra một yêu cầu đọc đĩa để nạp trang được tham chiếu vào một frame trống, trong khi đợi I/O, một 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 trạng thái ready.
- Giả sử phải thay trang vì không tìm được frame trống, PFSR được bổ sung như sau :
- Xác định vị trí trên đĩa của trang đang cần.
- Tìm một frame trống.
 - Nếu có frame trống thì dùng thôi chờ gì nữa.
 - Nếu không có frame trống thì dùng một giải thuật thay trang để chọn một trang hi sinh (victim page).
 - Ghi victim page lên đĩa, cập nhật page table và frame table tương ứng.
- Đọc trang đang cần vào frame trống (đã có được từ bước 2), cập nhật page table và frame table tương ứng.

4. Mô tả các giải thuật thay thế trang FIFO, OPT, LRU?

FIFO (First-In-First-Out): Như tên gọi của nó, trang nhớ nào được vào trước sẽ được “ưu tiên” swap ra ngoài khi có page-fault.

OPT (Optimal Page-Replacement Algorithm): Trang nhớ nào được tham chiếu trễ nhất trong tương lai sẽ được swap ra ngoài.

LRU (Least Recently Used Page-Replacement Algorithm): Mỗi trang được ghi nhận thời điểm được tham chiếu => Trang LRU là trang nhớ có thời điểm tham chiếu nhỏ nhất (OS tốn chi phí tìm kiếm trang LRU này mỗi khi có pagefault). Do vậy, LRU cần sự hỗ trợ của phần cứng cho việc tìm kiếm. (LRU giống OPT, nhưng tìm về quá khứ thay vì tương lai. Tức là, trang nhớ nào được tham chiếu ở thời điểm cách xa thời điểm hiện tại nhất sẽ được chọn để thay thế).

5. Giải pháp tập làm việc hoạt động như thế nào?

- Giải pháp tập làm việc dựa trên nguyên lý locality, xác định xem process thực sự sử dụng bao nhiêu frame.
- Giải pháp này định nghĩa một $WS(t)$ các tham chiếu trang nhớ của process gần đây nhất được quan sát, một Δ xác định khoảng thời gian tham chiếu.
- Đặt D bằng tổng các working-set size của mọi process trong hệ thống. Nếu $D > m$ (số frame của hệ thống) sẽ xảy ra thrashing. Khi khởi tạo một thủ tục sẽ cung cấp cho quá trình số lượng frame thỏa mãn working-set size của nó. Khi $D > m$ sẽ tạm dừng một trong các process.

6. (Bài tập mẫu) 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 4 khung trang.

- a. LRU
- b. FIFO
- c. Chiến lược tối ưu (OPT)

Trả lời:

a. LRU: Có 10 lỗi trang

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
1	1	1	1	1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	5	5	5	5	5	3	3	3	3	3	3	3	3	3
			4	4	4	4	6	6	6	6	6	7	7	7	7	1	1	1	1
*	*	*	*			*	*				*	*	*			*			

b. FIFO: Có 14 lỗi trang

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
1	1	1	1	1	1	5	5	5	5	5	3	3	3	3	3	1	1	1	1
	2	2	2	2	2	2	6	6	6	6	6	7	7	7	7	7	7	3	3
		3	3	3	3	3	3	2	2	2	2	2	6	6	6	6	6	6	6
			4	4	4	4	4	4	1	1	1	1	1	1	2	2	2	2	2
*	*	*	*			*	*	*	*		*	*	*		*	*		*	

c. OPT: Có 8 lỗi trang

1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
1	1	1	1	1	1	1	1	1	1	1	1	7	7	7	7	1	1	1	1
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
			4	4	4	5	6	6	6	6	6	6	6	6	6	6	6	6	6
*	*	*	*			*	*					*				*			