



>>> COURSE MATERIAL <<<

INTRODUCTION TO OPERATING SYSTEM / Course ID 502047

Thông tin dưới đây được dịch từ tài liệu StudyGuide tại trang OS-Book của sách giáo trình chính và phần Summary của sách. Mọi lỗi sai sót hay góp ý, xin gửi về cho tôi qua email trantrungtin@tdtu.edu.vn

Study Guide for Lecture notes ch10

- Bộ nhớ ảo [Virtual memory]: phân tách biệt giữa bộ nhớ luận lý của người dùng và bộ nhớ vật lý.
 - Chỉ một phần của chương trình cần thiết nằm trong bộ nhớ cho việc thực thi. Không gian bộ nhớ luận lý luôn lớn hơn không gian địa chỉ vật lý.
 - Cho phép không gian địa chỉ được chia sẻ giữa nhiều tiến trình, từ đó sẽ hạn chế hoán đổi [swapping].
 - Cho phép các trang được chia sẻ trong quá trình fork(), tăng tốc độ cho quá trình tạo mới tiến trình.
- Lỗi trang [Page fault] là kết quả của lần đầu tiên một trang cụ thể được truy cập, gây ra một bẫy [trap] trong bộ nhớ.
 - Phải cân nhắc hủy bỏ nếu bit tham khảo không hợp lệ, hoặc nếu như trang mong đợi chưa có trong bộ nhớ.
 - Nếu trang chưa có trong bộ nhớ: lấy khung trống, hoán đổi (nạp) trang vào khung, đặt lại bảng phân trang để chỉ ra trang đã nằm trong bộ nhớ, đặt bit xác thực, khởi động lại lệnh đã gây ra lỗi trang.
 - Nếu một chỉ thị truy cập vào nhiều trang gần nhau thì thiệt hại ít hơn do tính chất địa phương của tham khảo [locality of reference]
- Phân trang theo yêu cầu [Demand Paging] chỉ mang một trang vào bộ nhớ khi trang đó cần thiết, điều này sẽ giảm I/O và bộ nhớ cần cấp.
 - Bộ hoán đổi lười biếng [Lazy swapper] không bao giờ hoán đổi một trang và trong bộ nhớ trừ khi trang đó được yêu cầu.
 - Có thể dẫn đến hậu quả: tăng số lỗi trang.
 - Hiệu quả: $EAT = [(1-p) * \text{memory access} + p * (\text{page fault overhead} + \text{swap page out} + \text{swap page in} + \text{restart overhead})]$; với tỉ lệ lỗi trang p và $0 \leq p \leq 1$
 - memory access: thời gian truy cập bộ nhớ.
 - page fault overhead: thời gian phần cứng phát hiện lỗi trang.
 - swap page out / in: thời gian hoán chuyển trang ra / vào.
 - restart overhead: thời gian phần cứng khởi động lại câu lệnh đã bị lỗi trang.

- Nếu $p = 0$, không có lỗi trang; còn nếu $p = 1$, mọi tham khảo đều là lỗi trang.
- Có thể tối ưu Phân trang theo yêu cầu bằng cách tải toàn bộ ảnh của tiến trình vào vùng không gian hoán đổi tại thời điểm nạp.
- Phân trang theo yêu cầu thuần túy [Pure Demand Paging]: tiến trình bắt đầu khi không có trang nào trong bộ nhớ chính. Tất cả các trang khi yêu cầu lần đầu đều bị lỗi trang.
- Sao-chép-khi-ghi [Copy-on-Write (COW)] cho phép tiến trình cha và con chia sẻ trang trong bộ nhớ ngay từ khi khởi động.
 - Nếu một trong hai tiến trình sửa đổi trang chia sẻ, sau đó trang sẽ được sao chép.
- Bit sửa đổi [Modify (dirty) bit] có thể dùng để giảm quá tải di chuyển trang bằng cách chỉ ghi vào đĩa những trang có sửa đổi (trang có bit modify = TRUE).
- + Khi một trang bị thay thế, nó sẽ được ghi vào đĩa nếu được đánh dấu “dirty” và mang trang mong muốn vào.
- Các trang có thể được thay thế bằng các thuật toán khác nhau: FIFO, LRU.
 - Ngăn xếp có thể được dùng để ghi nhận lại những tham khảo trang trong quá khứ gần nhất (LRU là một giải thuật “ngăn xếp”).
 - Giải thuật Cơ hội thứ hai [Second chance] sử dụng một bit tham khảo.
 - Nếu giá trị 1, giảm thành 0 và để lại trong bộ nhớ.
 - Nếu giá trị 0, thay thế frame.
- Cấp phát trang cố định [Fixed page allocation]: Cấp phát tỉ lệ thuận – Cấp phát dựa vào kích cỡ tiến trình.
 - s_i = kích thước tiến trình P_i , $S = \sum s_i$, m = tổng số khung, a_i – số khung cấp phát cho P_i
 - $a_i = (s_i/S) * m$
- Thay thế toàn cục [Global replacement]: tiến trình sẽ chọn nạn nhân từ tập hợp tất cả khung trang.
 - Một tiến trình có thể lấy khung trang từ những tiến trình khác.
 - Thời gian thực thi tiến trình có thể có sự cách biệt lớn trong các lần thực thi.
 - Thông năng lớn hơn.
- Thay thế địa phương [Local replacement]: mỗi tiến trình sẽ chọn nạn nhân chỉ từ tập các khung trang của nó.
 - Hiệu suất phù hợp (và có lý hơn).
 - Có thể sử dụng hiệu quả thấp [under-utilization] bộ nhớ vật lý.
- Tần số lỗi trang sẽ rất cao nếu một tiến trình không có đủ các trang được nạp vào bộ nhớ chính.
 - Thrashing: a process is busy swapping pages in and out → minimal work is actually being performed
- Tập tin nhập xuất ánh xạ bộ nhớ [Memory-mapped] cho phép các tập tin nhập xuất được đối xử theo cách truy cập bộ nhớ thông thường bằng cách ánh xạ một khối đĩa cứng vào một trang.
- Đồng bộ nhập xuất [I/O Interlock]: Các trang đôi khi phải bị khoá vào trong bộ nhớ chính.

Summary Chapter 08 of book “OS concepts 10th edition”

- Bộ nhớ ảo trừu tượng hóa bộ nhớ vật lý thành một mảng lưu trữ thống nhất cực lớn.
- Các lợi ích của bộ nhớ ảo bao gồm: (1) chương trình có thể lớn hơn bộ nhớ vật lý, (2) chương trình không cần hoàn toàn trong bộ nhớ, (3) quy trình có thể chia sẻ bộ nhớ và (4) quy trình có thể được tạo ra một cách hiệu quả hơn.
- Phân trang theo yêu cầu là một kỹ thuật trong đó các trang chỉ được tải khi chúng được yêu cầu trong khi thực hiện chương trình. Do đó, các trang không bao giờ được yêu cầu không bao giờ được tải vào bộ nhớ.
- Xảy ra lỗi trang khi một trang hiện không có trong bộ nhớ được truy cập. Trang phải được đưa từ cửa hàng sao lưu vào khung trang có sẵn trong bộ nhớ.
- Sao-chép-khi-ghi [Copy-on-write] cho phép một tiến trình con chia sẻ cùng một không gian địa chỉ với cha của nó. Nếu một trong hai quá trình con hoặc cha ghi (hay sửa đổi) một trang, một bản sao của trang được tạo ra.

- Khi bộ nhớ khả dụng sắp hết, thuật toán thay thế trang sẽ chọn một trang hiện có trong bộ nhớ để thay thế bằng một trang mới. Các thuật toán thay thế trang bao gồm FIFO, tối ưu và LRU. Các thuật toán LRU thuần túy là không thực tế nên khó hiện thực và thay vào đó, các hệ thống sử dụng thuật toán xấp xỉ LRU [LRU-approximation].
- Các thuật toán thay thế trang toàn cục chọn một trang từ bất kỳ tiến trình nào trong hệ thống để thay thế, trong khi các thuật toán thay thế trang cục bộ chọn một trang từ tiến trình đã gây lỗi trang.
- Tồn thất nặng [Thrashing] xảy ra khi một hệ thống dành nhiều thời gian phân trang hơn là thực thi.
- Tính địa phương thể hiện rằng một tập hợp các trang được tích cực sử dụng cùng nhau. Khi một tiến trình thực thi, nó di chuyển từ địa phương này đến địa phương khác. Một bộ làm việc [working set] dựa trên tính địa phương và được định nghĩa là tập hợp các trang hiện đang được sử dụng bởi một tiến trình.
- Nén bộ nhớ [Memory compression] là một kỹ thuật quản lý bộ nhớ để nén một số trang thành một trang. Bộ nhớ nén là một giải pháp thay thế cho kỹ thuật phân trang và được sử dụng trên các hệ thống di động không hỗ trợ phân trang.
- Bộ nhớ dành cho nhân [Kernel memory] được phân bổ khác với các tiến trình cấp độ người dùng; nó được phân bổ trong các khối liền kề với các kích cỡ khác nhau. Hai kỹ thuật phổ biến để phân bổ bộ nhớ kernel là (1) hệ thống bạn bè [buddy system] và (2) phân bổ bản mỏng [slab allocation].
- Phạm vi tiếp cận TLB [TLB reach] đề cập đến lượng bộ nhớ có thể truy cập từ TLB và bằng với số lượng mục nhập trong TLB nhân với kích thước trang. Một kỹ thuật để tăng phạm vi tiếp cận TLB là tăng kích thước của các trang.
- Linux, Windows và Solaris quản lý bộ nhớ ảo tương tự nhau, sử dụng phân trang theo yêu cầu và sao-chép-khi-ghi, bên cạnh các tính năng khác. Mỗi hệ thống cũng sử dụng một biến thể của xấp xỉ LRU được gọi là thuật toán đồng hồ [clock algorithm].