**Kĩ thuật phân trang**

Phân trang là một kĩ thuật quản lý bộ nhớ giúp loại bỏ nhu cầu cấp phát liên tục của bộ nhớ vật lý (cấp phát không liên tục). Lược đồ này cho phép không gian địa chỉ vật lý của một tiến trình không liền nhau.

* Địa chỉ logic hoặc địa chỉ ảo (bit): Một địa chỉ do CPU tạo ra
* Không gian địa chỉ logic hoặc không gian địa chỉ ảo (word hoặc byte): Tập hợp tất cả các địa chỉ logic được tạo bởi một chương trình
* Địa chỉ vật lý (bit): Một địa chỉ thực sự có sẵn trên đơn vị bộ nhớ
* Không gian địa chỉ vật lý (word hoặc byte): Tập hợp tất cả các địa chỉ vật lý tương ứng với các địa chỉ logic

Việc ánh xạ từ địa chỉ ảo sang địa chỉ vật lý được thực hiện bởi đơn vị quản lý bộ nhớ (Memory Management Unit - MMU) là một thiết bị phần cứng và sự liên kết này được gọi là kỹ thuật phân trang.

* Không gian địa chỉ vật lý được chia thành các khối có kích thước cố định, được gọi là khung (frame).
* Không gian địa chỉ logic cũng được chia thành các khối có kích thước cố định, được gọi là các trang (page).
* Kích thước page và frame bằng nhau

Địa chỉ do CPU tạo ra được chia thành:

* Số trang – page number (p): Số bit đại diện cho các trang trong không gian địa chỉ logic
* Offset trang – page offset (d): Số bit đại diện cho vị trí offset của page trong không gian địa chỉ logic

Địa chỉ thực được chia thành

* Số khung – frame number (f): Số bit đại diện cho các khung của Không gian địa chỉ vật lý
* Offset khung – frame offset (d): Số bit biểu thị vị trí offset của frame trong không gian địa chỉ vật lý

Một cấu trúc dữ liệu được gọi là bảng trang (page table) được sử dụng để theo dõi mối quan hệ giữa một trang của quy trình với một khung trong bộ nhớ vật lý.

Việc triển khai bảng trang (page table) có thể được thực hiện bằng cách sử dụng các thanh ghi chuyên dụng. Nhưng việc sử dụng đăng ký cho bảng trang chỉ thỏa mãn nếu bảng trang nhỏ. Nếu bảng trang chứa một số lượng lớn các entry thì chúng ta có thể sử dụng TLB (translation Look-aside buffer) một bộ đệm phần cứng đặc biệt, nhỏ gọn, tra cứu nhanh.

Mỗi mục nhập trong TLB bao gồm hai phần: thẻ và giá trị.

Khi bộ nhớ này được sử dụng, một mục sẽ được so sánh với tất cả các thẻ đồng thời. Khi mục tương tự được tìm thấy thì giá trị tương ứng sẽ được trả về.

Khi một quy trình được thực thi, các page tương ứng của nó sẽ được tải vào bất kỳ frame bộ nhớ khả dụng nào. Khi máy tính hết RAM, hệ điều hành sẽ di chuyển các page bộ nhớ thích hợp hoặc không mong muốn sang bộ nhớ phụ để giải phóng RAM cho các tiến trình khác và đưa chúng trở lại khi chương trình cần.

Ưu và nhược điểm:

* Phân trang làm giảm phân mảnh ngoại vi, nhưng vẫn bị phân mảnh nội vi.
* Phân trang đơn giản, dễ thực hiện và được coi là một kỹ thuật quản lý bộ nhớ hiệu quả.
* Do kích thước của các trang và khung bằng nhau, việc hoán đổi trở nên rất dễ dàng.
* Bảng trang yêu cầu thêm dung lượng bộ nhớ, do đó có thể không tốt cho hệ thống có RAM nhỏ.

**Bộ nhớ ảo**

Bộ nhớ ảo là kĩ thuật mở rộng vùng nhớ, giúp cho người dùng được giải phóng hoàn toàn khỏi mối bận tâm về giới hạn bộ nhớ. Trích xuất một phần của bộ nhớ phụ, vùng nhớ đó sẽ được xem như là bộ nhớ ảo hỗ trợ cho các thao tác của RAM, giả lập RAM.

Nó là một kỹ thuật được thực hiện bằng cả phần cứng và phần mềm. Cho phép xử lý một tiến trình không được nạp toàn bộ vào bộ nhớ vật lý. Bộ nhớ ảo mô hình hoá bộ nhớ như một bảng lưu trữ rất lớn và đồng nhất, tách biệt hẳn khái niệm không gian địa chỉ và không gian vật lý. Người sử dụng chỉ nhìn thấy và làm việc trong không gian địa chỉ ảo, việc chuyển đổi sang không gian vật lý do hệ điều hành thực hiện với sự trợ giúp của các cơ chế phần cứng cụ thể.

Bộ nhớ ảo được triển khai bằng cách sử dụng phân trang theo nhu cầu (demand paging) hoặc Phân đoạn theo nhu cầu (demand segmentation).

Một hệ thống phân trang theo yêu cầu khá giống với hệ thống sử dụng kỹ thuật phân trang kết hợp với kỹ thuật swapping. Một tiến trình được xem như một tập các trang, thường trú trên

bộ nhớ phụ (bộ nhớ ảo – thường là đĩa cứng). Khi cần xử lý, tiến trình sẽ được nạp vào bộ nhớ chính nhưng chỉ đối với những trang cần thiết ở thời điểm run-time. Như vậy một trang chỉ được nạp vào bộ nhớ chính khi có yêu cầu.

Trong khi thực thi một chương trình, nếu chương trình tham chiếu đến một trang không có sẵn trong bộ nhớ chính vì nó đã bị hoán đổi gần đây, bộ xử lý sẽ coi tham chiếu bộ nhớ không hợp lệ này là lỗi trang (page fault) và chuyển quyền điều khiển từ chương trình sang hệ điều hành yêu cầu trang trở lại bộ nhớ.

Ưu và nhược điểm:

* Bộ nhớ ảo lớn.
* Sử dụng bộ nhớ hiệu quả hơn.
* Không có giới hạn về mức độ đa chương trình.
* Số lượng bảng và chi phí xử lý các ngắt trang lớn hơn so với trường hợp của các kỹ thuật quản lý phân trang đơn giản.

Thuật toán thay thế trang

Thuật toán thay thế trang được áp dụng khi gặp lỗi trang, xem xét thông tin hạn chế đồng thời cố gắng chọn trang nào nên được thay thế để giảm thiểu tổng số trang bị bỏ lỡ, đồng thời cân bằng nó với chi phí lưu trữ chính và thời gian xử lý của chính thuật toán. Có nhiều thuật toán thay thế trang khác nhau. Chỉ tiêu đánh giá thuật toán thông qua khả năng chạy trên một chuỗi các địa chỉ cần truy xuất và giảm thiểu số trang lỗi (page fault) phát sinh.

Thuật toán First In First Out (FIFO)

* Trang cũ nhất trong bộ nhớ chính là trang sẽ được chọn để thay thế.
* Dễ dàng thực hiện, thay thế các trang từ phần đuôi và thêm các trang mới ở phần đầu.

Thuật toán Optimal page

* Thuật toán thay thế trang tối ưu có tỷ lệ lỗi trang thấp nhất trong tất cả các thuật toán. Các thuật toán thay thế trang tối ưu OPT, MIN.
* Thay thế trang có khả năng không được sử dụng trong thời gian dài nhất.

Thuật toán Least Recently Used (LRU)

* Trang không được sử dụng trong thời gian dài nhất trong bộ nhớ chính là trang sẽ được chọn để thay thế.
* Dễ dàng thực hiện, thay thế các trang bằng cách nhìn lại mốc thời gian.

Thuật toán Page Buffering

* Để quá trình bắt đầu nhanh chóng, hãy giữ một nhóm các khung trống.
* Trên trang bị lỗi, hãy chọn một trang để thay thế.
* Viết trang mới trong khung của nhóm khung trống, đánh dấu bảng trang và khởi động lại quá trình.
* Bây giờ ghi trang tương ứng ra khỏi đĩa và đặt khung chứa trang đã thay thế vào vùng trống.

Thuật toán Least frequently Used (LFU)

* Trang có số lượng nhỏ nhất là trang sẽ được chọn để thay thế.
* Thuật toán này gặp phải tình huống trong đó một trang được sử dụng nhiều trong giai đoạn đầu của quy trình, nhưng sau đó không bao giờ được sử dụng lại.

Thuật toán Most frequently Used (MFU): Thuật toán này dựa trên lập luận rằng trang có số lượng nhỏ nhất mới có thể được đưa vào tuy nhiên vẫn chưa được sử dụng.

**Khảo sát:**

Graphical user interface, application

Description automatically generated **Graphical user interface, application

Description automatically generated**

Graphical user interface, application

Description automatically generated

Ta có:

Kích thước trang = kích thước khung trang =

Không gian vật lý (Main memory – RAM) =

Không gian logic (Kích thước bộ nhớ ảo) =

Ta có không gian vật lý số bit quản lý địa chỉ ô nhớ

Ta có kích thước trang số bit tối thiểu để quản lý các offset trong trang

Số khung trang vật lý

Số trang logic tối đa trên không gian tiến trình

**Tham khảo:**

Giáo trình Hệ điều hành

[Operating System - Memory Management (tutorialspoint.com)](https://www.tutorialspoint.com/operating_system/os_memory_management.htm)

[Operating System - Virtual Memory (tutorialspoint.com)](https://www.tutorialspoint.com/operating_system/os_virtual_memory.htm)

[Paging in Operating System - GeeksforGeeks](https://www.geeksforgeeks.org/paging-in-operating-system/)

[Virtual Memory in Operating System - GeeksforGeeks](https://www.geeksforgeeks.org/virtual-memory-in-operating-system/)

[Memory Hierarchy Design - Part 6. The Intel Core i7, fallacies, and pitfalls - EDN](https://www.edn.com/memory-hierarchy-design-part-6-the-intel-core-i7-fallacies-and-pitfalls/)