Cơ sở

- Bộ nhớ ảo là 1 kỹ thuật quản lý bộ nhớ được sử dụng bởi các hệ điều hành để đảm bảo hoạt động của máy tính khi thiếu bộ nhớ vật lý bằng cách tạm thời chuyển các trang dữ liệu từ bộ nhớ truy cập ngẫu nhiên (RAM) sang lưu trữ đĩa. Bộ nhớ ảo tạo ra sự giả dối về một lượng bộ nhớ lớn hơn so với sự hiện diện thực tế trong hệ thống.

- Trong bộ nhớ ảo, hệ điều hành chia bộ nhớ của 1 chương trình thành các khối có kích thước cố định gọi là trang. Các trang này chỉ được tải vào bộ nhớ vật lý khi cần đến và được đưa ra khỏi bộ nhớ khi không còn cần thiết. Khi một tiến trình yêu cầu truy cập đến một trang không có sẵn trong bộ nhớ vật lý, hệ điều hành lấy trang đó từ đĩa và chuyển nó vào bộ nhớ vật lý.

Không gian địa chỉ ảo

- (Virtual Address Space) là 1 không gian địa chỉ được tạo bởi HĐH nhằm cung cấp cho mỗi tiến trình (process) 1 địa chỉ ảo độc lập nhau, dù chúng có thể có các bộ nhớ thực tế (physical memory) trùng lặp. Các địa chỉ ảo được sử dụng bởi tiến trình để truy cập dữ liệu của nó, còn các địa chỉ thực tế sẽ tương ứng với các vùng nhớ thực tế của hệ thống.

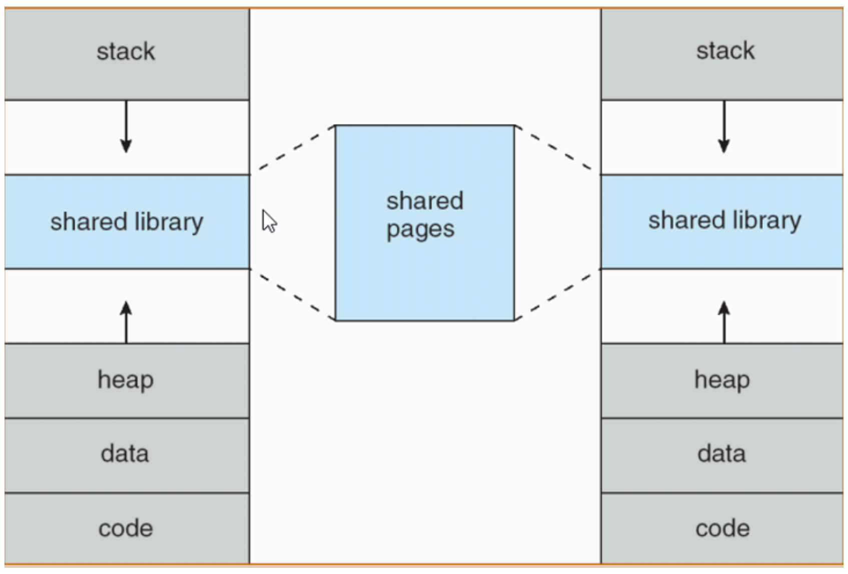
- Không gian địa chỉ ảo cho phép các tiến trình truy cập đến 1 không gian địa chỉ lớn hơn so với tổng dung lượng bộ nhớ thực tế trên hệ thống. HĐH sẽ quản lý việc ánh xạ địa chỉ ảo thành các địa chỉ thực tế tương ứng và sẽ thực hiện việc chuyển đổi địa chỉ ảo sang địa chỉ thực tế khi quá trình yêu cầu truy cập tới 1 vùng nhớ được ánh xạ vào không gian địa chỉ ảo nhưng chưa được tải lên bộ nhớ thực tế.

- Điều này cho phép các quá trình sử dụng bộ nhớ lớn hơn so với dung lượng bộ nhớ thực tế trên hệ thống, tăng hiệu quả sử dụng bộ nhớ và cho phép các quá trình được thực thi mà không cần có sẵn đầy đủ dung lượng bộ nhớ cần thiết.

Thư viện chia sẻ dùng bộ nhớ ảo

- Có nhiều thư viện chia sẻ sử dụng bộ nhớ ảo trong các ngôn ngữ lập trình khác nhau. Ví dụ:
+ C++: Boost.Interprocess, Qt Shared Memory
+ Python: mmap module
+ Java: java.nio.MappedByteBuffer, Chronicle Map
+ .NET: MemoryMappedFile class, System.IO.MemoryMappedFiles namespace

- Những thư viện này cho phép các tiến trình khác nhau truy cập vào cùng một vùng bộ nhớ ảo và chia sẻ dữ liệu với nhau. Điều này có thể giúp tăng hiệu suất và giảm thiểu sự sử dụng bộ nhớ thực tế trên hệ thống.



Phân trang theo yêu cầu (on-demand paging)

- Là 1 kỹ thuật trong quản lý bộ nhớ ảo mà chỉ các trang cần thiết của 1 tiến trình mới được tải vào bộ nhớ vật lý. Thay vì tải toàn bộ tiến trình vào bộ nhớ vật lý khi bắt đầu, HĐH chỉ tải các trang cần thiết để thực thi các phần của tiến trình đó.

- Khi 1 tiến trình yêu cầu truy cập đến 1 trang bị trục trặc, nó sẽ kích hoạt 1 trang không có sẵn. Trang mới này sẽ được sao chép từ bộ nhớ đĩa và tải vào bộ nhớ vật lý trước khi tiến trình được phép truy cập vào nó.

- Phân trang theo yêu cầu giúp tối ưu hóa việc sử dụng bộ nhớ vật lý bằng cách giảm thiểu lượng bộ nhớ được dành riêng cho các quá trình và cho phép nhiều tiến trình chia sẻ bộ nhớ. Tuy nhiên, nó cũng có thể dẫn đến hiện tượng gián đoạn bộ nhớ và giảm hiệu suất hệ thống nếu các trang cần thiết không được tải đúng thời điểm.

- Ưu điểm:
+ Tiết kiệm bộ nhớ hơn so với phân trang truyền thống, bởi vì chỉ tải các trang cần thiết.
+ Giảm thời gian phân trang ban đầu, không cần phải tải toàn bộ chương trình vào bộ nhớ thứ cấp một lần.
+ Tăng hiệu suất đọc và ghi dữ liệu trên đĩa, bởi vì chỉ tải các trang cần thiết từ đĩa.

- Nhược điểm:
+ Tăng thời gian truy cập dữ liệu ban đầu, bởi vì cần phải tải các trang cần thiết từ đĩa vào bộ nhớ chính trước khi sử dụng.
+ Gây ra độ trễ trong việc chuyển đổi giữa trang trên bộ nhớ chính và trên bộ nhớ thứ cấp, nếu các trang không nằm gần nhau về mặt vật lý.
+ Đôi khi có thể gây ra tình trạng tràn bộ đệm (buffer overflow) nếu chương trình yêu cầu quá nhiều trang cùng lúc.

Lỗi trang (page fault)

- Là 1 sự kiện xảy ra khi 1 chương trình yêu cầu truy cập đến 1 trang (page) trong bộ nhớ ảo nhưng trang đó chưa được nạp vào bộ nhớ vật lý. Khi đó, hệ điều hành sẽ phải thực hiện việc đọc dữ liệu từ bộ nhớ thứ cấp (disk) vào bộ nhớ vật lý trước khi cung cấp cho chương trình yêu cầu truy cập trang đó.

- Lỗi trang là 1 phần của cơ chế quản lý bộ nhớ ảo (virtual memory) trong hệ thống máy tính. Nó cho phép các chương trình sử dụng bộ nhớ nhiều hơn so với bộ nhớ vật lý có sẵn bằng cách sử dụng bộ nhớ ảo. Tuy nhiên, việc sử dụng bộ nhớ ảo có thể dẫn đến lỗi trang khi một trang cần thiết không có sẵn trong bộ nhớ vật lý và phải được đọc từ đĩa cứng.

- Ưu điểm: cho phép sử dụng bộ nhớ ảo để tăng hiệu suất của hệ thống bằng cách giảm lượng tài nguyên bộ nhớ cần thiết để chạy chương trình.
- Nhược điểm: có thể gây ra một số nhược điểm như tăng thời gian truy cập bộ nhớ khi một trang mới phải được tải vào bộ nhớ và làm giảm hiệu suất của hệ thống

Các bước xử lý lỗi trang

- Có thể liệt kê các bước xử lý lỗi trang như sau:
+ HĐH xác định trang bị lỗi bằng cách kiểm tra bản đồ trang hoặc thông tin bảo vệ của trang đó.
+ HĐH sau đó cố gắng khắc phục lỗi trang bằng cách sử dụng các kỹ thuật khôi phục lỗi như đọc lại dữ liệu từ ổ đĩa hoặc sửa chữa các tệp tin bị lỗi.
+ Nếu HĐH không thể khắc phục lỗi trang, nó sẽ ghi nhận lỗi đó và thông báo cho người dùng hoặc ứng dụng về lỗi trang.
+ Hệ điều hành tiếp tục hoạt động với các trang còn lại của ứng dụng hoặc quá trình, trong khi đó thông báo lỗi có thể được gửi đến quản trị hệ thống để xử lý lỗi toàn hệ thống.
+ Trong một số trường hợp, quản trị viên hệ thống có thể phải can thiệp bằng cách xóa các trang bị lỗi hoặc tắt các ứng dụng hoặc tiến trình đang sử dụng các trang bị lỗi để ngăn ngừa lỗi trang phát sinh lại.

Trường hợp không còn frame lỗi

- Trong trường hợp không còn frame trống để lưu trữ trang được yêu cầu từ bộ nhớ ảo, hệ thống sẽ phải thực hiện một số biện pháp để giải quyết vấn đề này. Có một số phương pháp để xử lý tình huống này như sau:
+ Swapping: Hệ thống sẽ tìm kiếm trong bộ nhớ thứ cấp (ví dụ: đĩa cứng) để tìm một khối dữ liệu không sử dụng và sao chép nội dung của một số frame khác trong bộ nhớ ảo vào đó. Trang yêu cầu sẽ được sao chép vào frame mới này.

+ Sử dụng một kỹ thuật khác để xử lý lỗi trang: Ví dụ như sử dụng bộ nhớ đệm để lưu trữ trang hoặc thực hiện giải phóng bộ nhớ không sử dụng hoặc giảm thiểu số lượng trang cần sử dụng.

- Tuy nhiên, việc sử dụng các phương pháp này để giải quyết tình huống không còn frame trống để lưu trữ trang có thể ảnh hưởng đến hiệu suất của hệ thống, do đó, nên cân nhắc và tối ưu hóa để đạt được hiệu suất tốt nhất.

Hiệu năng của phân trang theo yêu cầu

Thay thế trang

Các thuật toán thay thế trang

Cấp phát frames

Thrashing

- Thrashing là tình trạng trong hệ thống máy tính khi việc truy xuất trang (page) trong bộ nhớ ảo (virtual memory) xảy ra quá nhiều, dẫn đến hiệu suất của hệ thống giảm đáng kể. Khi máy tính bị thrashing, hệ thống sẽ dành nhiều thời gian để swap các trang từ RAM sang đĩa cứng và ngược lại thay vì thực hiện các tác vụ thực tế của người dùng, dẫn đến sự chậm trễ và không thể sử dụng được ứng dụng hoặc hệ điều hành.

- Thrashing thường xảy ra khi bộ nhớ thực (physical memory) đã được sử dụng hết và hệ thống cố gắng sử dụng bộ nhớ ảo quá nhiều, tuy nhiên do số lượng trang được sử dụng quá lớn nên việc swap giữa RAM và đĩa cứng trở nên quá tải và khiến hệ thống hoạt động chậm lại.