**Chương 9: Quản Lý Bộ Nhớ**

Nội dung

[9.1. Bối cảnh 1](#_Toc194925554)

[9.1.1 Phần cứng cơ bản 1](#_Toc194925555)

[a. Sự bảo vệ: 1](#_Toc194925556)

[b.Bảo vệ địa chỉ phần cứng 2](#_Toc194925557)

[9.1.2. Ràng buộc địa chỉ 2](#_Toc194925558)

[a. Ràng buộc cách hướng dẫn và dữ liệu vào bộ nhớ 3](#_Toc194925559)

[b.Xử lý nhiều bước của chương trình người dùng 3](#_Toc194925560)

[9.1.3. Không gian địa chỉ logic và vật lý 4](#_Toc194925561)

[a.Đơn vị quản lý bộ nhớ ( MMU ) 4](#_Toc194925562)

[b.Đơn vị quản lý bộ nhớ (Tiếp theo) 5](#_Toc194925563)

[c.Đơn vị quản lý bộ nhớ (Tiếp theo) 5](#_Toc194925564)

[9.1.4. Tải động 6](#_Toc194925565)

[9.1.5.Liên kết động 6](#_Toc194925566)

[9.2.Cấp phát bộ nhớ liên tục 6](#_Toc194925567)

[9.2.1.Phân bổ liên tục 6](#_Toc194925568)

[9.2.2.Phân bổ liền kề (Tiếp theo) 7](#_Toc194925569)

[9.2.3.Hỗ trợ phần cứng cho các thanh ghi vị lại và giới hạn 7](#_Toc194925570)

[9.2.4.Phân vùng biến 8](#_Toc194925571)

[9.2.5.Sự phân mảnh 9](#_Toc194925572)

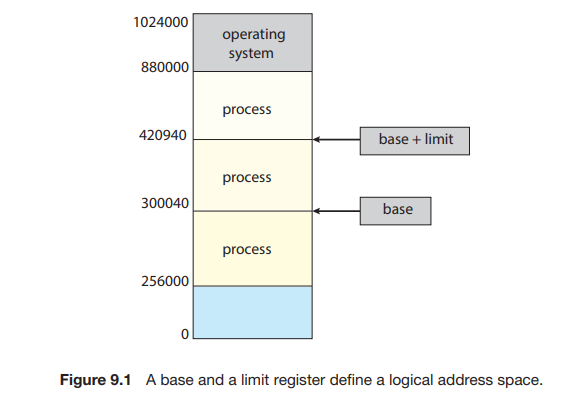
# 9.1. Bối cảnh

## 9.1.1 Phần cứng cơ bản

### a. Sự bảo vệ:

Cần đảm bảo rằng một tiến trình chỉ có thể truy cập những địa chỉ đỏ trong không gian địa chỉ của nó.

Chúng ta có thể cung cấp sự bảo vệ này bằng cách sử dụng một cặp để Và thanh ghi giới hạn xác định không địa chỉ logic của một tiên trình.



\*Giải thích:

Hệ thống điều hành (Operating System):

Được cấp phát ở phần trên cùng của bộ nhớ, từ địa chỉ 880000 đến 1024000.

Đây là không gian dành riêng cho hệ thống điều hành để quản lý và thực thi các tác vụ quan trọng.

Các quy trình (Processes):

Có ba quy trình được phân biệt thông qua giá trị base và limit:

Quy trình 1: Nằm giữa khoảng 420940 và 880000.

Quy trình 2: Nằm giữa khoảng 300040 và 420940.

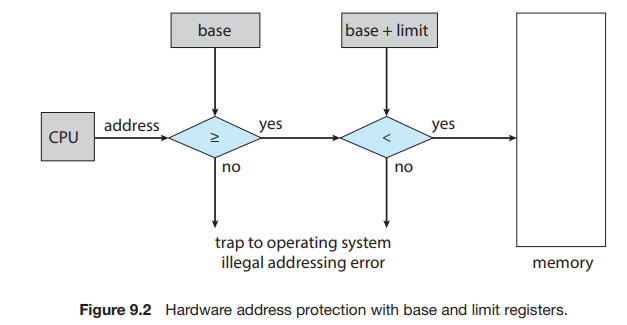
Quy trình 3: Nằm giữa khoảng 256000 và 300040.

Hai mũi tên chỉ ra từ hai trong số các quy trình này đến hộp được ghi là "base + limit" và "base", cho thấy cách các quy trình này được phân biệt trong bộ nhớ.

Địa chỉ bộ nhớ:

Trục dọc bên trái biểu thị các địa chỉ bộ nhớ, từ 0 đến 1024000.

### b.Bảo vệ địa chỉ phần cứng



1.CPU (Central Processing Unit): Đơn vị xử lý trung tâm của máy tính, nơi tạo ra địa chỉ.

2.Base (Cơ sở): Giá trị cơ bản, thường là địa chỉ bắt đầu của một phân đoạn bộ nhớ.

3.Base + Limit (Cơ sở + Giới hạn): Tổng của giá trị cơ sở và giới hạn, xác định địa chỉ kết thúc của phân đoạn bộ nhớ.

4. Memory (Bộ nhớ): Nơi lưu trữ dữ liệu và chương trình.

Các bước hoạt động như sau:

•CPU tạo ra một địa chỉ.

•Địa chỉ được so sánh với Base và Base + Limit.

• Nếu địa chỉ lớn hơn hoặc bằng Base và nhỏ hơn Base + Limit, CPU truy cập bộ nhớ.

• Nếu không, xảy ra lỗi địa chỉ không hợp lệ, và máy tính chuyển đến hệ điều hành để xử lý.

## 9.1.2. Ràng buộc địa chỉ

• Các chương trình trên đĩa, sẵn sàng được đưa vào bộ nhớ để thực thi dưới dạng hàng đợi đâu vào

• Nếu không có hỗ trợ thì phải nạp vào địa chỉ 0000

• Bất tiện khi có địa chỉ vật lý xử lý người dùng đầu tiên luôn ở mức 0000

• Làm thế nào nó có thể không được?

• Địa chỉ được thể hiện theo những cách khác nhau ở các giai đoạn khác nhau trong vòng đời của chương trình

• Địa chỉ mã nguồn thường mang tính biểu tượng

• Địa chỉ mã được biên dịch liên kết đến các địa chỉ có thể di chuyển được.

• 14 byte kể từ đầu mô-đun này "

• Trình liên kết hoặc trình tải sẽ liên kết các địa chỉ có thể định vị lại với các địa chỉ tuyệt đối tức là 74014

• Mỗi liên kết ánh xạ một không gian địa chỉ này sang một không gian địa chỉ khác

### a. Ràng buộc cách hướng dẫn và dữ liệu vào bộ nhớ

• Liên kết địa chỉ của hướng dẫn và dữ liệu với địa chỉ bộ nhớ có thể xảy ra ở ba giai đoạn khác nhau

• Thời gian biên dịch : Nếu vị trí bộ nhớ được biết trước, mã tuyệt đối có thể được tạo; phải biên dịch lại mã nếu vị trí bắt đầu thay đổi

• Thời gian tải : Phải tạo mã có thể định vị lại nếu không biết vị trí bộ nhớ tại thời điểm biên dịch

• Thời gian thực thi : Việc liên kết bị trì hoãn cho đến thời gian chạy nếu tiến trình có thê được di chuyên trong quá trình thực thi từ phân đoạn bộ nhớ này sang phân đoạn bộ nhớ khác

• Cần hỗ trợ phần cứng cho bản đồ địa chỉ (ví dụ: cơ sở và giới hạn số đăng ký)

### b.Xử lý nhiều bước của chương trình người dùng

1.Source Program: Chương trình nguồn được biên dịch bởi trình biên dịch hoặc trình tổng hợp trong quá trình biên dịch.

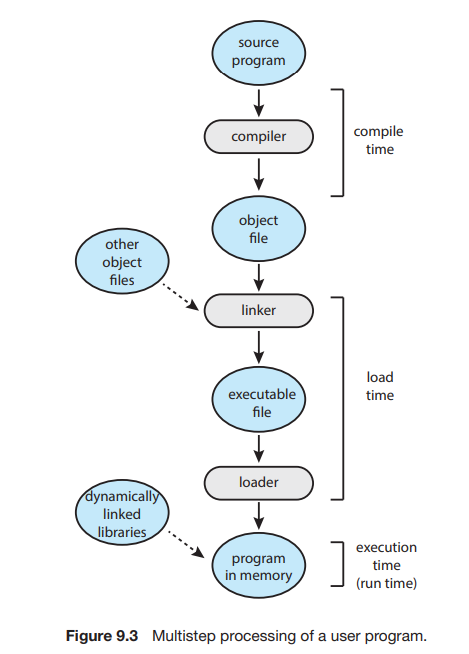
2.Object Module: Sau biên dịch, chương trình nguồn tạo ra một đối tượng module, có thể kết hợp với các đối tượng module khác.

3.Linkage Editor: Trình biên tập liên kết kết hợp các module này để tạo ra một đối tượng tải vào bộ nhớ tại thời điểm tải.

4.System Libraries: Các thư viện hệ thống cũng có thể được bao gồm trong giai đoạn này.

5.Loader: Trình nạp tải module này và thêm các thư viện hệ thống được tải động thông qua liên kết động.

6.In-Memory Binary Image: Cuối cùng, một hình ảnh nhị phân trong bộ nhớ được tạo ra và thực thi trong thời gian chạy.



## 9.1.3. Không gian địa chỉ logic và vật lý

• Khái niệm về không gian địa chỉ logic được gắn với một không gian địa chỉ vật lý là trung tâm của việc quản lý bộ nhớ thích hợp

• Hợp lý Địa chỉ – do CPU tạo ra; còn gọi là ảo Địa chỉ

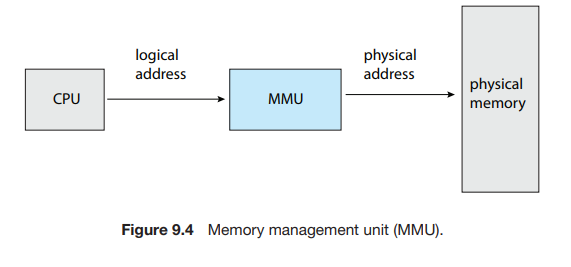
• Thuộc vật chất Địa chỉ – địa chỉ được nhìn thấy bởi đơn vị bộ nhớ

• Địa chỉ logic và địa chỉ vật lý giống nhau trong sơ đồ liên kết địa chỉ thời gian biên dịch và thời gian tải; địa chỉ logic (ảo) và địa chỉ vật lý khác nhau trong sơ đồ liên kết địa chỉ tại thời điểm thực hiện

• Hợp lý Địa chỉ không gian là tập hợp tất cả các địa chỉ logic được tạo bởi một chương trình

• Thuộc vật chất Địa chỉ không gian là tập hợp tất cả các địa chỉ vật lý được tạo bởi một chương trình

### a.Đơn vị quản lý bộ nhớ ( MMU )



• CPU (Central Processing Unit): Đơn vị xử lý trung tâm tạo ra địa chỉ logic. Đây là địa chỉ mà chương trình sử dụng để truy cập dữ liệu.

• MMU (Memory Management Unit): Đơn vị quản lý bộ nhớ chuyển đổi địa chỉ logic thành địa chỉ vật lý. MMU thực hiện việc ánh xạ giữa hai loại địa chỉ này.

• Bộ nhớ vật lý (Physical Memory): Đây là nơi lưu trữ dữ liệu thực tế trong máy tính.

### b.Đơn vị quản lý bộ nhớ (Tiếp theo)

• Hãy xem xét sơ đồ đơn giản. đó là sự tổng quát hóa của sơ đồ thanh ghi cơ sở.

• Thanh ghi cơ sở bây giờ được gọi là thanh ghi tái định vị

• Giá trị trong thanh ghi tái định vị được thêm vào mọi địa chỉ được tạo bởi tiến trình người dùng tại thời điểm nó được gửi tới bộ nhớ

• Chương trình người dùng xử lý các địa chỉ logic; nó không bao giờ nhìn thây địa chỉ vật lý thực sự

• Liên kết thời gian thực thi xảy ra khi tham chiếu được thực hiện tới vị trí trong bộ nhớ

• Địa chỉ logic được liên kết với địa chỉ vật lý

### c.Đơn vị quản lý bộ nhớ (Tiếp theo)

Hoạt động và Luồng điều khiển:

• Một sơ đồ hoạt động cho thấy cách các hoạt động dẫn đến nhau. Mỗi hoạt động có thể được gọi là một hoạt động hệ thống.

Một hoạt động dẫn đến hoạt động tiếp theo trong luồng điều khiển. Luồng này có thể song song, đồng thời, hoặc phân nhánh.

Biểu tượng và Ký hiệu:

Sơ đồ hoạt động sử dụng nhiều biểu tượng, bao gồm:

Biểu tượng bắt đầu: Đại diện cho một quy trình hoặc quy trình công việc bắt đầu.

Biểu đồ quyết định: Hiển thị quyết định và các đường dẫn phân nhánh.

Biểu tượng ghi chú: Cho phép truyền tải thông báo bổ sung không thuộc về sơ đồ.

Biểu tượng kết nối: Hiển thị luồng định hướng hoặc luồng điều khiển của hoạt động.

Mục tiêu và Ứng dụng:

Sơ đồ hoạt động nắm bắt hành vi năng động của hệ thống.

Nó xây dựng hệ thống thực thi bằng cách sử dụng các phương pháp kỹ thuật chuyên tiếp và đảo ngược.

Sơ đồ hoạt động giúp hình dung bản chất động của một hệ thống.

## 9.1.4. Tải động

Toàn bộ chương trình cần phải có trong bộ nhớ để thực thi

Quy trình không được tải cho đến khi nó được gọi

Sử dụng không gian bộ nhớ tốt hơn; thói quen không sử dụng không bao giờ được tải

Tất cả các thủ tục được lưu trên đĩa ở định dạng tải có thể định vị lại

Hữu ích khi cần số lượng lớn mã để xử lý các trường hợp không thường xuyên xảy ra

Không cần hỗ trợ đặc biệt từ hệ điều hành

Thực hiện thông qua thiết kế chương trình

Hệ điều hành có thể trợ giúp bằng cách cung cấp các thư viện để triển khai tải động

## 9.1.5.Liên kết động

• Toàn bộ chương trình cần phải có trong bộ nhớ để thực thi

• Quy trình không được tải cho đến khi nó được gọi

• Sử dụng không gian bộ nhớ tốt hơn; thói quen không sử dụng không bao giờ được tải

• Tất cả các thủ tục được lưu trên đĩa ở định dạng tải có thể định vị lại

• Hữu ích khi cần số lượng lớn mã để xử lý các trường hợp không thường xuyên xảy ra

• Không cần hỗ trợ đặc biệt từ hệ điều hành

• Thực hiện thông qua thiết kế chương trình

• Hệ điều hành có thể trợ giúp bằng cách cung cấp các thư viện để triển khai tải động

# 9.2.Cấp phát bộ nhớ liên tục

## 9.2.1.Phân bổ liên tục

• Bộ nhớ chính phải hỗ trợ cả quy trình của hệ điều hành và người dùng

• Nguồn lực hạn chế, phải phân bổ hiệu quả

• Phân bổ liền kề là một phương pháp sớm

• Bộ nhớ chính thường chia thành hai phân vùng :

• Hệ điều hành thường trú, thường được giữ ở bộ nhớ thấp với vectơ ngắt

• Các tiến trình của người dùng sau đó được giữ trong bộ nhớ cao

• Mỗi quá trình chứa trong một phần bộ nhớ liền kề

## 9.2.2.Phân bổ liền kề (Tiếp theo)

• Các thanh ghi tái định vị được sử dụng để bảo vệ các tiến trình của người dùng với nhau và khỏi việc thay đổi mã và dữ liệu của hệ điều hành

• Thanh ghi cơ sở chứa giá trị địa chỉ vật lý nhỏ nhất

• Thanh ghi giới hạn chứa nhiều địa chỉ logic - mỗi địa chỉ logic phải nhỏ hơn thanh ghi giới hạn

• MMU ánh xạ địa chỉ logic một cách linh hoạt

• Sau đó có thể cho phép các hành động như mã hạt nhân tạm thời và kích thước thay đổi kernel

## 9.2.3.Hỗ trợ phần cứng cho các thanh ghi vị lại và giới hạn

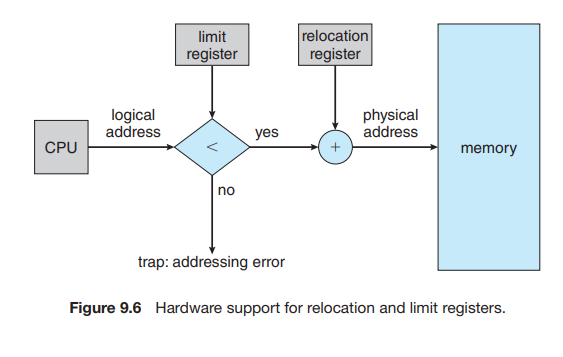
• CPU (Central Processing Unit) tạo ra một địa chỉ logic.

• Địa chỉ logic này được so sánh với giá trị trong thanh ghi giới hạn.

• Nếu địa chỉ logic nhỏ hơn giá trị trong thanh ghi giới hạn, nó được cộng thêm vào thanh ghi di chuyên để tạo ra một địa chỉ vật lý. Địa chỉ vật lý này sau đó được sửdụng để truy cập bộ nhớ.

• Nếu địa chỉ logic không nhỏ hơn giá trị trong thanh ghi giới hạn, một lỗi định vị sẽ được phát hiện và xử lý.

• Thanh ghi di chuyển là một thành phần quan trọng trong việc ánh xạ giữa địa chỉ logic và địa chỉ vật lý. Khi CPU thực hiện các phép tính, nó sử dụng địa chỉ logic để truy cập dữ liệu trong bộ nhớ, và sau đó ánh xạ nó thành địa chỉ vật lý để thực hiện truy cập thực tế.



## 9.2.4.Phân vùng biến

• Phân bổ nhiều phân vùng.

• Mức độ đa chương trình bị giới hạn bởi số lượng phân vùng.

• Phân vùng biến kích thước cho hiệu quả (có kích thước phù hợp với nhu cầu của một quy trình nhất định).

• Hole – khối bộ nhớ khả dụng; các lỗ có kích thước khác nhau nằm rải rác trong bộ nhớ.

• Khi một tiến trình đến, nó được cấp phát bộ nhớ từ một lỗ đủ lớn để chứa nó.

• Quá trình thoát sẽ giải phóng phân vùng của nó, các phân vùng miễn phí liền kề được kết hợp.

Hệ điều hành duy trì thông tin về:

1. Phân vùng được cấp phát:

Có ba cột đại diện cho việc cấp phát bộ nhớ cho các quy trình và hệ điều hành.

Cột đầu tiên có OS được cấp phát ở trên, tiếp theo là quy trình 5, quy trình 8 và quy trình 2 từ trên xuống dưới.

Cột thứ hai cũng có OS ở trên, tiếp theo là chỉ có quy trình 5 và quy trình 2.

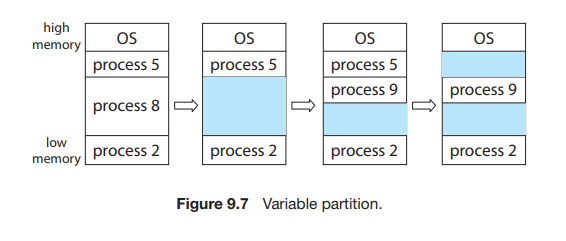
Cột thứ ba hiển thị OS ở trên lại với quy trình 9 ở dưới và sau đó là quy trình 2 ở cuối cùng.

2. Phân vùng trống (lỗ hổng):

Có hai cột đại diện cho các khoảng trống trống trong việc cấp phát bộ nhớ:

Cột đầu tiên có OS ở trên, một khoảng trống được đánh dấu là “lỗ hổng”, tiếp theo là quy trình 9 và sau đó là một lỗ hổng khác ở trên quy trình 2.

Cột thứ hai tương tự nhưng có thêm một lỗ hồng giữa OS và quy trình 9.



## 9.2.5.Sự phân mảnh

• Bên ngoài Sự phân mảnh – tổng dung lượng bộ nhớ tồn tại để đáp ứng yêu cầu nhưng không liền kề nhau.

• Nội bộ Sự phân mảnh – bộ nhớ được phân bổ có thể lớn hơn một chút so với bộ nhớ được yêu cầu; sự khác biệt về kích thước này là bộ nhớ bên trong của một phân vùng nhưng không được sử dụng.

• Phân tích phù hợp đầu tiên cho thấy rằng với N khối được phân bổ, 0,5 N khối bị mất do phân mảnh.

• 1/3 có thể không sử dụng được -> 50% luật lệ.

9.2.6.Sự phân mảnh (tiếp theo)

• Giảm sự phân mảnh bên ngoài bằng cách nén.

• Xáo trộn nội dung bộ nhớ để đặt tất cả bộ nhớ trống vào một. khối lớn.

• Việc nén chỉ có thể thực hiện được nếu việc tái định vị là động. và được thực hiện tại thời điểm thực hiện.

• Vấn đề vào/ra.

• Chốt công việc trong bộ nhớ khi nó liên quan đến I/O.

• Chỉ thực hiện I/O vào bộ đệm của hệ điều hành.

• Bây giờ hãy xem xét rằng cửa hàng hỗ trợ có vấn đề phân mảnh tương tự.