

**MÔN HỌC: HỆ ĐIỀU HÀNH**  
**CÂU HỎI VÀ BÀI TẬP CHƯƠNG 7**

**1. Quản lý bộ nhớ là gì? Các yêu cầu đối với quản lý bộ nhớ là gì?**

Quản lý bộ nhớ là công việc của hệ điều hành với sự hỗ trợ của phần cứng nhằm phân phối, sắp xếp các process trong bộ nhớ sao cho hiệu quả.

Các yêu cầu đối với việc quản lý bộ nhớ bao gồm:

- Cấp phát bộ nhớ cho các process: Việc thực hiện một yêu cầu cấp phát bao gồm việc định vị một ô nhớ chưa được sử dụng với kích thước đủ lớn.
- Tái định vị (relocation): Khi swapping.
- Bảo vệ: Phải kiểm tra sự truy xuất bộ nhớ từ các process có hợp lệ không.
- Chia sẻ: Cho phép các process chia sẻ vùng nhớ chung.
- Kết gắn địa chỉ nhớ luận lý của user vào địa chỉ thực.

**2. Trình bày đặc điểm các loại địa chỉ bộ nhớ?**

- Địa chỉ vật lý (physical address-địa chỉ thực): một vị trí thực trong bộ nhớ chính.
- Địa chỉ luận lý (logical address): một vị trí nhớ được diễn tả trong một chương trình
  - Trình biên dịch (compiler) tạo ra mã lệnh chương trình trong đó mỗi tham chiếu bộ nhớ đều là địa chỉ luận lý
  - Địa chỉ tương đối (relative address): kiểu địa chỉ luận lý trong đó các địa chỉ được biểu diễn tương đối so với một vị trí xác định nào đó trong chương trình.
  - Địa chỉ tuyệt đối (absolute address): địa chỉ tương đương với địa chỉ thực.

**3. Địa chỉ lệnh và dữ liệu được chuyển đổi thành địa chỉ thực có thể xảy ra tại những thời điểm nào?**

Địa chỉ lệnh và dữ liệu được chuyển đổi thành địa chỉ thực có thể xảy ra tại ba thời điểm khác nhau.

- **Compile time:** nếu biết trước địa chỉ bộ nhớ của chương trình thì có thể kết gán địa chỉ tuyệt đối lúc biên dịch
  - Ví dụ: chương trình .COM của MS-DOS
  - Khuyết điểm: phải biên dịch lại nếu thay đổi địa chỉ nạp chương trình
- **Load time:** vào thời điểm loading, loader phải chuyển đổi địa chỉ khả tái định vị thành địa chỉ thực dựa trên một địa chỉ nền
  - Địa chỉ thực được tính toán vào thời điểm nạp chương trình  
⇒ phải tiến hành reload nếu địa chỉ nền thay đổi
- **Excution time:** khi trong quá trình thực thi, process có thể được di chuyển từ segment này sang segment khác trong bộ nhớ thì quá trình chuyển đổi địa chỉ được trì hoãn đến thời điểm thực thi
  - Cần sự hỗ trợ của phần cứng cho việc ánh xạ địa chỉ  
Ví dụ: Trường hợp địa chỉ luận lý là relocatable thì có thể dùng thanh ghi base và limit,..
  - Sử dụng trong đa số các OS đa dụng trong đó có các cơ chế swapping, paging, segmentation

#### 4. Liên kết động (dynamic linking) là gì?

Quá trình link đến một module ngoài (external module) được thực hiện sau khi đã tạo xong load module (i.e. file có thể thực thi, executable)

- Ví dụ trong Windows: module ngoài là các file .DLL còn trong Unix, các module ngoài là các file .so (shared library)

#### 5. Phân mảnh ngoại và phân mảnh nội là gì?

Phân mảnh ngoại tồn tại khi tổng lượng bộ nhớ còn trống đủ để phục vụ một yêu cầu mới từ process, tuy nhiên do các phần trống đó không liên tục nhau mà nằm ở nhiều vị trí cách xa nhau, vì vậy có bộ nhớ trống mà không thể cấp phát được.

Phân mảnh ngoại xảy ra do việc nạp và rút các process trong bộ nhớ. Như hình trên các bạn thấy, những ô xám từng là nơi những process

nhỏ được nạp vào hệ thống trước đó. Những process nhỏ đó đã hoàn thành và rút ra khỏi hệ thống, để lại phần trống của bộ nhớ rời rạc như trên. Bây giờ, process 7 muốn vào nhưng không được, do không có vị trí nào đủ bộ nhớ để cung cấp cho nó.

## **6. Như thế nào là phân vùng động, phân vùng cố định? Các chiến lược placement dùng để làm gì? Có những chiến lược nào?**

Phân vùng động (Dynamic Partitioning) là một kỹ thuật trong hệ điều hành, nơi kích thước của phân vùng không được khai báo ban đầu mà được khai báo tại thời điểm tải quá trình<sup>1</sup>. Phân vùng đầu tiên được dành riêng cho hệ điều hành và không gian còn lại được chia thành nhiều phần. Kích thước của mỗi phân vùng sẽ bằng kích thước của tiến trình. Kích thước phân vùng thay đổi tùy theo nhu cầu của quá trình để có thể tránh được sự phân mảnh bên trong.

Phân vùng cố định (Fixed Partitioning) là một kỹ thuật trong hệ điều hành, nơi danh sách các phân vùng được tạo một lần và sẽ không bao giờ thay đổi. Trong phân vùng cố định, việc phân bổ và phân bổ rất phức tạp vì kích thước phân vùng sẽ thay đổi mỗi khi nó được gán cho một quá trình mới<sup>1</sup>. Hệ điều hành phải theo dõi tất cả các phân vùng.

Chiến lược placement (Placement Strategy) là một chiến lược quan trọng trong quản lý bộ nhớ của hệ điều hành. Các chiến lược placement thường được sử dụng để quyết định vị trí nào trong bộ nhớ sẽ được chọn để cấp phát cho một tiến trình<sup>2</sup>. Có ba chiến lược placement cơ bản:

- Best-fit: Chọn khối nhớ trống nhỏ nhất phù hợp với yêu cầu.
- First-fit: Chọn khối nhớ trống phù hợp đầu tiên kể từ đầu bộ nhớ.
- Next-fit: Chọn khối nhớ trống phù hợp đầu tiên kể từ vị trí cấp phát cuối cùng.
- Worst-Fit : Chọn khối nhớ trống lớn nhất.

## **7. Cơ chế phân trang là gì? Cơ chế chuyển đổi địa chỉ trong phân trang được thực hiện như thế nào?**

Cơ chế phân trang (Paging) là một kỹ thuật quản lý bộ nhớ trong hệ điều hành. Trong cơ chế này, không gian bộ nhớ được chia thành các đơn vị cố định có kích thước bằng nhau gọi là trang. Mỗi tiến trình sẽ được cấp một bảng phân trang, mỗi mục trong bảng phân trang chứa chỉ số frame chứa trang tương ứng trong bộ nhớ thực.

Cơ chế chuyển đổi địa chỉ trong phân trang được thực hiện như sau:

- Cơ chế phần cứng hỗ trợ thực hiện chuyển đổi địa chỉ trong cơ chế phân trang là bảng trang (pages table).
- Mỗi phần tử trong bảng trang cho biết các địa chỉ bắt đầu của vị trí lưu trữ trang tương ứng trong bộ nhớ vật lý (số hiệu khung trang trong bộ nhớ vật lý đang chứa trang).
- Page offset,  $d$ , được kết hợp với địa chỉ nền (base address) của frame để định vị địa chỉ thực.

## **8. Bảng trang được cài đặt và tổ chức như thế nào? Thời gian truy xuất hiệu dụng (effective access time) được xác định như thế nào?**

Bảng trang (Page Table) là một cấu trúc dữ liệu được sử dụng trong hệ điều hành để quản lý bộ nhớ ảo. Bảng trang chứa thông tin về cách ánh xạ từ địa chỉ ảo sang địa chỉ thực. Mỗi tiến trình sẽ có một bảng trang riêng.

Cách tổ chức bảng trang có thể khác nhau tùy thuộc vào hệ điều hành và phần cứng. Một số hệ điều hành có thể sử dụng bảng trang đa cấp, bảng trang có độ rộng biến đổi, hoặc bảng trang chia sẻ.

Thời gian truy xuất hiệu dụng (Effective Access Time) liên quan đến thời gian cần thiết để truy cập bộ nhớ. Trong hệ thống sử dụng bộ nhớ ảo và phân trang, thời gian truy xuất hiệu dụng sẽ bao gồm thời gian cần thiết để tìm kiếm bảng trang và thời gian cần thiết để truy

cập bộ nhớ thực. Nếu xảy ra trường hợp trang bị trùng phạt (page fault), thời gian truy xuất hiệu dụng sẽ tăng lên do cần phải tải trang từ bộ nhớ phụ (như ổ đĩa cứng) vào bộ nhớ chính.

## 9. Cơ chế hoán vị là gì?

Một process có thể tạm thời bị swap ra khỏi bộ nhớ chính và lưu trên một hệ thống lưu trữ phụ. Sau đó, process có thể được nạp lại vào bộ nhớ để tiếp tục quá trình thực thi.

10. (Bài tập mẫu) Giả sử bộ nhớ chính được cấp phát thành các phân vùng có kích thước là 600K, 500K, 200K, 300K (theo thứ tự), sau khi thực thi xong, các tiến trình có kích thước 212K, 417K, 112K, 426K (theo thứ tự) sẽ được cấp phát bộ nhớ như thế nào, nếu sử dụng: Thuật toán First fit, Best fit, Next fit, Worst fit? Thuật toán nào cho phép sử dụng bộ nhớ hiệu quả nhất trong trường hợp trên?

Trả lời:

a. Xét trường hợp bộ nhớ được phân vùng cố định:

\*First fit:

Tiến trình 212K sẽ được cấp phát vùng nhớ 600K.

Tiến trình 417K sẽ được cấp phát vùng nhớ 500K.

Tiến trình 112K sẽ được cấp phát vùng nhớ 200K.

Tiến trình 426K sẽ phải chờ vì không còn vùng nhớ trống thỏa yêu cầu.

\* Best fit:

Tiến trình 212K sẽ được cấp phát vùng nhớ 300K.

Tiến trình 417K sẽ được cấp phát vùng nhớ 500K.

Tiến trình 112K sẽ được cấp phát vùng nhớ 200K.

Tiến trình 426K sẽ được cấp phát vùng nhớ 600K.

\* Next fit:

Tiến trình 212K sẽ được cấp phát vùng nhớ 600K.

Tiến trình 417K sẽ được cấp phát vùng nhớ 500K.

Tiến trình 112K sẽ được cấp phát vùng nhớ 200K.

Tiến trình 426K sẽ phải chờ vì không còn vùng nhớ trống thỏa yêu cầu.

\* Worst fit:

Tiến trình 212K sẽ được cấp phát vùng nhớ 600K.

Tiến trình 417K sẽ được cấp phát vùng nhớ 500K.

Tiến trình 112K sẽ được cấp phát vùng nhớ 300K.

Tiến trình 426K sẽ phải chờ vì không còn vùng nhớ trống thỏa yêu cầu.

b. Xét trường hợp bộ nhớ được phân vùng động:

\*First fit:

Tiến trình 212K sẽ được cấp phát vùng nhớ 600K => Vùng nhớ trống 388K.

Tiến trình 417K sẽ được cấp phát vùng nhớ 500K.

Tiến trình 112K sẽ được cấp phát vùng nhớ 388K.

Tiến trình 426K sẽ phải chờ vì không còn vùng nhớ trống thỏa yêu cầu.

\* Best fit:

Tiến trình 212K sẽ được cấp phát vùng nhớ 300K.

Tiến trình 417K sẽ được cấp phát vùng nhớ 500K.

Tiến trình 112K sẽ được cấp phát vùng nhớ 200K.

Tiến trình 426K sẽ được cấp phát vùng nhớ 600K.

\* Next fit:

Tiến trình 212K sẽ được cấp phát vùng nhớ 600K.

Tiến trình 417K sẽ được cấp phát vùng nhớ 500K.

Tiến trình 112K sẽ được cấp phát vùng nhớ 200K.

Tiến trình 426K sẽ phải chờ vì không còn vùng nhớ trống thỏa yêu cầu.

\* Worst fit:

Tiến trình 212K sẽ được cấp phát vùng nhớ 600K. => Vùng nhớ trống 388K.

Tiến trình 417K sẽ được cấp phát vùng nhớ 500K.

Tiến trình 112K sẽ được cấp phát vùng nhớ 388K.

Tiến trình 426K sẽ phải chờ vì không còn vùng nhớ trống thỏa yêu cầu.

**11. (Bài tập mẫu) Xét một không gian địa chỉ có 12 trang, mỗi trang có kích thước 2K, ánh xạ vào bộ nhớ vật lý có 32 khung trang.**

**a. Địa chỉ logic gồm bao nhiêu bit?**

**b. Địa chỉ physic gồm bao nhiêu bit?**

Trả lời:

a. Địa chỉ logic gồm 2 phần: chỉ số trang và độ dời (offset) trong trang. Cần 4 bit để biểu diễn chỉ số trang và 11 bit ( $2^{11} = 2K = 2048$ ) để biểu diễn độ dời trong trang. Vậy địa chỉ logic gồm 15 bit.

b. Thực hiện tương tự. Địa chỉ physic gồm 16 bit.

**12. (Bài tập mẫu) Xét một hệ thống sử dụng kỹ thuật phân trang, với bảng trang được lưu trữ trong bộ nhớ chính.**

**a. Nếu thời gian cho một lần truy xuất bộ nhớ bình thường (memory reference) là 200ns thì mất bao nhiêu thời gian cho một thao tác truy xuất bộ nhớ (paged memory reference) trong hệ thống này?**

**b. Nếu sử dụng TLBs với hit-ratio là 75%, thời gian để tìm trong TLBs xem như bằng 0, tính thời gian truy xuất bộ nhớ (effective access time) trong hệ thống?**

Trả lời:

a. Mỗi thao tác truy xuất bộ nhớ trong hệ thống này sẽ cần thực hiện 2 lần truy xuất bộ nhớ thông thường: truy xuất bảng trang (để xác định vị trí khung trang) và truy xuất vị trí bộ nhớ (xác định dựa trên sự kết hợp giá trị khung trang tìm được ở lần trước với độ dời trong trang). Do đó thời gian của một thao tác truy xuất bộ nhớ sẽ là  $200 \times 2 = 400\text{ns}$ .

b. Thời gian truy xuất bộ nhớ (effective access time) trong hệ thống:  
$$EAT = (2 - \alpha)x + \varepsilon = 250 \text{ ns}.$$

13. Một máy tính 32-bit địa chỉ, sử dụng một bảng trang 2 cấp. Địa chỉ ảo được phân bố như sau: 9 bit dành cho bảng trang cấp 1, 11 bit cho bảng trang cấp 2 và còn lại cho offset. Cho biết kích thước một trang trong hệ thống và địa chỉ ảo có bao nhiêu trang?

Kích thước một trang trong hệ thống: Số bit dành cho offset xác định kích thước của một trang. Trong trường hợp này, địa chỉ ảo 32-bit có 9 bit dành cho bảng trang cấp 1, 11 bit dành cho bảng trang cấp 2, vậy số bit còn lại cho offset là  $32 - 9 - 11 = 12$  bit. Do đó, kích thước một trang sẽ là  $2^{12}$  byte = 4KB.

Số lượng trang trong không gian địa chỉ ảo được xác định bởi số bit dành cho bảng trang cấp 1 và bảng trang cấp 2. Vì vậy, tổng số trang sẽ là  $2^{(9+11)} = 2^{20} = 1048576$  trang.

14. Giả sử địa chỉ ảo 32-bit được phân tách thành 4 trường a, b, c, d. 3 trường đầu tiên được dùng cho bảng trang 3 cấp, trường thứ 4 dành cho offset. Số lượng trang có phụ thuộc vào kích thước của cả 4 trường này không? Nếu không, những trường nào ảnh hưởng đến số lượng trang, những trường nào không ảnh hưởng?

Số lượng trang không phụ thuộc vào cả 4 trường mà chỉ phụ thuộc vào 3 trường đầu tiên a,b,c và trường d (offset) sẽ không ảnh hưởng đến số lượng trang.

15. (Bài tập mẫu) Cho bảng trang như hình bên.

a. Địa chỉ vật lý 6568 sẽ được chuyển thành địa chỉ ảo bao nhiêu? Biết rằng kích thước mỗi frame là 1K bytes.

b. Địa chỉ ảo 3254 sẽ được chuyển thành địa chỉ vật lý bao nhiêu? Biết rằng kích thước mỗi frame là 2K bytes.

0	6
1	4
2	5
3	7
4	1
5	9

Trả lời:



- a. Địa chỉ vật lý 6568 nằm ở khung trang 6 với độ dài 424. Trang 0 được nạp vào khung trang 6 => Địa chỉ ảo là 424.
- b. Địa chỉ 3254 nằm ở trang 1 với độ dài 1206. Trang 1 được nạp vào khung trang 4 => Địa chỉ vật lý là 9398.

16. Xét một hệ thống sử dụng kỹ thuật phân trang, với bảng trang được lưu trữ trong bộ nhớ chính.

- a. Nếu thời gian cho một lần truy xuất bộ nhớ bình thường là 124 nanoseconds, thì mất bao nhiêu thời gian cho một thao tác truy xuất bộ nhớ trong hệ thống này?
- b. Nếu sử dụng TLBs với hit-ratio (tỉ lệ tìm thấy) là 95%, thời gian để tìm trong TLBs bằng 34ns, tính thời gian cho một thao tác truy xuất bộ nhớ trong hệ thống (effective memory reference time).

a. Trong một hệ thống sử dụng kỹ thuật phân trang, mỗi thao tác truy xuất bộ nhớ thực tế đều cần hai lần truy xuất bộ nhớ: một lần để lấy thông tin từ bảng trang và một lần để lấy dữ liệu thực sự. Do đó, thời gian cho một thao tác truy xuất bộ nhớ trong hệ thống này sẽ là 2 lần thời gian truy xuất bộ nhớ bình thường, tức là  $2 * 124 \text{ ns} = 248 \text{ ns}$ .

b. Khi sử dụng TLBs, thời gian truy xuất bộ nhớ hiệu dụng được tính như sau:

- Nếu TLB hit  $\rightarrow 34 + 124 = 158 \text{ ns}$ .

- Nếu TLB miss  $\rightarrow 34 + 2 * 124 = 282 \text{ ns}$ .

Vậy, thời gian truy xuất bộ nhớ hiệu dụng là:  $0,95 * 158 + 0,05 * 282 = 166,2 \text{ ns}$ .

17. Xét một hệ thống sử dụng kỹ thuật phân trang, với bảng trang được lưu trữ trong bộ nhớ chính. Nếu sử dụng TLBs với hit-ratio

(tỉ lệ tìm thấy) là 87%, thời gian để tìm trong TLBs là 24 ns. Thời gian truy xuất bộ nhớ trong hệ thống (effective memory reference time) là 175ns. Tính thời gian cho một lần truy xuất bộ nhớ bình thường?

Ta có:  $175 = 0,87 \cdot (24 + t) + 0,13 \cdot (24 + 2t)$

->  $t \approx 133,63 \text{ ns}$

18. Biết thời gian truy xuất trong bộ nhớ thường không sử dụng TLBs là 250 ns. Thời gian tìm kiếm trong bảng TLBs là 26ns. Hỏi xác suất tìm thấy trong TLBs bằng bao nhiêu nếu thời gian truy xuất trong bộ nhớ chính là 182 ns?

Ta có:  $182 = p \cdot (26 + 250) + (1-p) \cdot (250)$

->  $p = 1,376$  -> Không bao giờ xảy ra.