

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 hoạt động hiệu quả nhất.
- Các yêu cầu đối với công việc quản lý bộ nhớ:
 - + Cấp phát bộ nhớ cho các process
 - + Tái định vị (relocation): khi swapping,...
 - + Bảo vệ: phải kiểm tra truy xuất bộ nhớ có hợp lệ không
 - + Chia sẻ: cho phép các process chia sẻ vùng nhớ chung
 - + Kết gán địa chỉ nhớ luận lý của user vào địa chỉ thư

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

- Địa chỉ vật lý (physical address) (địa chỉ thực) là một vị trí thực trong bộ nhớ chính.
- Địa chỉ luận lý (logical address) là một vị trí nhớ được diễn tả trong một chương trình (còn gọi là địa chỉ ảo virtual address).
- Các trình biên dịch (compiler) tạo ra mã lệnh chương trình mà trong đó mọi tham chiếu bộ nhớ đều là địa chỉ luận lý
- Địa chỉ tương đối (relative address) (địa chỉ khả tái định vị, relocatable address) là một 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.
- Execution 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.

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

- Liên kết động (dynamic linking) thực hiện việc kết nối một chương trình với thư viện trong quá trình thực thi (runtime), thay vì tại thời điểm biên dịch (compile time). Điều này cho phép chương trình sử dụng thư viện mà nó cần mà không cần biên dịch lại toàn bộ chương trình.
- Cơ chế: chỉ khi nào cần được gọi đến thì một thủ tục mới được nạp vào bộ nhớ chính => tăng độ hiệu dụng của bộ nhớ bởi vì các thủ tục không được gọi đến sẽ không chiếm chỗ trong bộ nhớ.
- Hỗ trợ từ hệ điều hành
 - + Thông thường, user chịu trách nhiệm thiết kế và hiện thực các chương trình có dynamic loading.
 - + Hệ điều hành chủ yếu cung cấp một số thủ tục thư viện hỗ trợ, tạo điều kiện dễ dàng hơn cho lập trình viên.

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 nội là hiện tượng khi một process chiếm dụng lượng bộ nhớ nhỏ hơn dung lượng mà nó được cấp phát (tức là phần bộ nhớ không được sử dụng nằm bên trong phân vùng đang được cấp phát cho process).

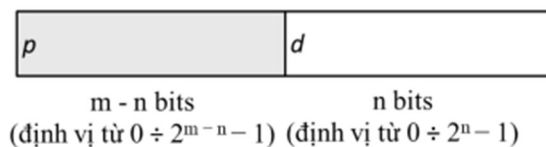
6. Như thế nào là phân vùng động, phân vùng cố định? Các chiến lược placement dùng để làm gì? Có những chiến lược nào?

Trong ngữ cảnh của hệ thống quản lý bộ nhớ (Memory Management), phân vùng động và phân vùng cố định liên quan đến cách quản lý và phân bổ bộ nhớ cho các tiến trình trong hệ điều hành.

- Phân vùng cố định (Fixed Partitioning): Trong phân vùng cố định, bộ nhớ được chia thành các phân vùng có kích thước cố định trước. Mỗi phân vùng này được cấp phát cho một tiến trình. Nếu không có tiến trình nào sử dụng hết kích thước phân vùng, không thể sử dụng phần còn lại cho các tiến trình khác.
- Phân vùng động (Dynamic Partitioning): Trong phân vùng động, bộ nhớ được chia thành các phân vùng có kích thước linh hoạt, phù hợp với kích thước của tiến trình. Khi một tiến trình cần bộ nhớ, hệ thống cấp phát một phân vùng với kích thước phù hợp.
- Chiến lược placement được sử dụng để quyết định vị trí cụ thể trong bộ nhớ nào một tiến trình sẽ được đặt. Các chiến lược placement giúp tối ưu hóa việc sử dụng bộ nhớ và giảm lãng phí, nhưng cũng có nhược điểm và ảnh hưởng đối với hiệu suất của hệ thống. Có một số chiến lược phổ biến:
 - + First Fit (Sắp xếp đầu tiên): Tiến trình được đặt vào phân vùng đầu tiên mà nó vừa vặn được. Đây là chiến lược nhanh chóng và dễ thực hiện, nhưng có thể dẫn đến sự lãng phí không gian.
 - + Best Fit (Sắp xếp tốt nhất): Tiến trình được đặt vào phân vùng nhỏ nhất mà nó vừa vặn được. Điều này giảm thiểu lãng phí không gian, nhưng có thể tạo ra nhiều "đám chặn" nhỏ.
 - + Worst Fit (Sắp xếp tồi nhất): Tiến trình được đặt vào phân vùng lớn nhất mà nó vừa vặn được. Chiến lược này cũng giảm thiểu lãng phí không gian, nhưng có thể tạo ra các đám chặn lớn.
 - + Next Fit (Sắp xếp tiếp theo): Tương tự như First Fit, nhưng bắt đầu tìm kiếm từ vị trí của tiến trình cuối cùng được đặt.
 - + Buddy System: Bộ nhớ được chia thành các khối có kích thước là các số lũy thừa của 2. Khi một yêu cầu bộ nhớ được thực hiện, hệ thống sử dụng chiến lược buddy system để cấp phát một khối có kích thước phù hợp 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:
 - + Là cơ chế cấp phát bộ nhớ không liên tục
 - + Chia bộ nhớ vật lý thành các khối cố định gọi là các khung trang (frames)
 - + Chia bộ nhớ luận lý thành các khối nhớ bằng nhau gọi là các trang (pages). Kích thước của các page bằng kích thích của các frame
 - + Chương trình có bao nhiêu trang thì cần bấy nhiêu khung trống trong bộ nhớ để nạp vào
 - + Thiết lập bảng phân trang (page table) để ánh xạ địa chỉ luận lý thành địa chỉ thực
- Chuyển đổi địa chỉ trong phân trang:
 - + Địa chỉ luận lý gồm có: số hiệu trang (page number) p , địa chỉ tương đối trong trang (page offset) d .
 - + Nếu kích thước của không gian địa chỉ ảo là 2^m và kích thước của trang là 2^n thì:



Bảng trang sẽ có tổng cộng $2^m/2^n = 2^{m-n}$ mục (entry)

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:
 - + Bảng phân trang thường được lưu giữ trong bộ nhớ chính:
 - + Mỗi process được hệ điều hành cấp một bảng phân trang
 - + Thanh ghi page-table base (PTBR) trỏ đến bảng phân trang
 - + Thanh ghi page-table length (PTLR) biểu thị kích thước của bảng phân trang (có thể được dùng trong cơ chế bảo vệ bộ nhớ)
 - + Theo cơ chế cài đặt này thì một thao tác truy cập lệnh hoặc dữ liệu cần đến 2 lần truy cập bộ nhớ chính:
 - + Lần 1 cho bảng trang
 - + Lần 2 cho lệnh hoặc dữ liệu
 - + Thường dùng một bộ phận cache phần cứng có tốc độ truy xuất và tìm kiếm cao, gọi là thanh ghi kết hợp (associative register) hoặc translation look-aside buffers (TLBs)

- Thời gian truy xuất hiệu dụng:
 - + Thời gian tìm kiếm trong TLB (associative lookup): ϵ
 - + Thời gian một chu kỳ truy xuất bộ nhớ: x
 - + Hit ratio: tỉ số giữa số lần chỉ số trang được tìm thấy (hit) trong TLB và số lần truy xuất khởi nguồn từ CPU. Kí hiệu hit ratio: α
 - + Thời gian cần thiết để có được chỉ số frame:
 - + Khi chỉ số trang có trong TLB (hit) $\epsilon + x$
 - + Khi chỉ số trang không có trong TLB (miss) $\epsilon + x + x$
 - + Thời gian truy xuất hiệu dụng $EAT = (\epsilon + x)\alpha + (\epsilon + 2x)(1 - \alpha) = (2 - \alpha)x + \epsilon$

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. Quá trình đó là một cơ chế hoán vị.

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 $200 * 2 = 400ns$.

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 bổ 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?

11	9	12
----	---	----

32 bits

Kích thước 1 trang trong hệ thống là 2^{12} và địa chỉ có 2^{11} trang ảo.

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?

a	b	c	d (offset)
---	---	---	------------

Số lượng trang không phụ thuộc vào kích thước 4 trường. Trường d ảnh hưởng đến số lượng trang vì số trang = $2^{(32-d)}$

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

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

- 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.

Trả lời:

- a. Địa chỉ vật lý 6568 nằm ở khung trang 6 với độ dời 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?

Thời gian truy xuất: $x = 124/2 = 62\text{ns}$

- 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).

$$\alpha = 0.95$$

$$\varepsilon = 34\text{ns}$$

$$x = 62\text{ns}$$

$$\text{EAT} = (2 - \alpha)x + \varepsilon = 99.1\text{ns}$$

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?

$$\alpha = 0.87$$

$$\varepsilon = 24\text{ns}$$

$$x = 175\text{ns}$$

$$\text{EAT} = (2 - \alpha)x + \varepsilon = 175 \Rightarrow x = 133.6$$

Thời gian 1 lần truy xuất bộ nhớ: $2x = 267.3\text{ns}$

18. Biết thời gian truy xuất trong bộ nhớ thường không sử dụng TLBs là 250ns . 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 182ns ?

$$x = 250/2 = 125\text{ns}$$

$$\varepsilon = 26\text{ns}$$

$$\text{EAT} = (2 - \alpha)x + \varepsilon = 182\text{ns} \Rightarrow \alpha = 0.752$$

Vậy xác suất tìm thấy trong TLBs là 75.2%