



Lab3 - bt chương 3

Kiến trúc máy tính và Hệ điều hành (Đại học Kinh tế Quốc dân)



Scan to open on Studocu

1. Giải thích sự khác biệt giữa địa chỉ logic và địa chỉ physic?

Tiêu chí	Địa chỉ logic	Địa chỉ vật lý
Định nghĩa	Là địa chỉ ảo do CPU tạo ra	Là 1 vị trí trong 1 đơn vị bộ nhớ
Không gian địa chỉ	Tập hợp tất cả các địa chỉ logic được tạo ra bởi CPU tham chiếu đến một chương trình	Tập hợp tất cả các địa chỉ vật lý được ánh xạ tới các địa chỉ logic tương ứng
Hiện thi	Người dùng có thể xem địa chỉ logic của một chương trình	Người dùng không bao giờ xem được địa chỉ vật lý của chương trình
Truy cập	Được người dùng sử dụng để truy cập địa chỉ vật lý	Người dùng không thể truy cập trực tiếp
Hình thành	Được tạo ra bởi CPU	Được tính bằng MMU

2. Giải thích sự khác biệt giữa hiện tượng phân mảnh nội vi và ngoại vi?

Phân mảnh nội vi	Phân mảnh ngoại vi
Xảy ra khi các khối bộ nhớ có kích thước cố định được phân bổ cho các tiến trình	Xảy ra khi không gian bộ nhớ có kích thước thay đổi được phân bổ cho các tiến trình một cách linh hoạt
Khi bộ nhớ được gán cho tiến trình lớn hơn 1 chút so với bộ nhớ mà tiến trình yêu cầu, việc này sẽ tạo ra không gian bộ nhớ trống trong khối được phân bổ -> phân mảnh bên trong	Khi tiến trình được loại bỏ khỏi bộ nhớ, nó tạo ra không gian trống trong bộ nhớ - > phân mảnh bên ngoài
Bộ nhớ phải được phân vùng thành các khối có kích thước thay đổi và gán khối phù hợp nhất cho tiến trình	Nén, phân trang và phân đoạn

3. Giả sử bộ nhớ chính được phân thành các phân vùng có kích thước là 600K, 500K, 200K, 300K (theo thứ tự), cho biết các tiến trình có kích thước 212K, 417K, 112K và 426K (theo thứ tự) sẽ được cấp phát bộ nhớ như thế nào, nếu sử dụng :

a) Thuật toán First fit

212k được cấp phát vùng nhớ 600K

417K được cấp phát vùng nhớ 500K

112K được cấp phát vùng nhớ 388K(vùng nhớ được tạo thành sau khi cấp phát cho tiến trình 212K, $600K - 212K = 388K$)

426K phải chờ, vì không còn vùng nhớ trống thỏa yêu cầu

b) Thuật toán Best fit

212K được cấp phát vùng nhớ 300K

417K được cấp phát vùng nhớ 500K

112K được cấp phát vùng nhớ 200K

426K được cấp phát vùng nhớ 600K

c) Thuật toán Worst fit

212K được cấp phát vùng nhớ 600K

417K được cấp phát vùng nhớ 500K

112K được cấp phát vùng nhớ 388K (vùng nhớ được tạo thành sau khi cấp phát cho tiến trình 212K, $600K - 212K = 388K$)

426 phải đợi, do không còn vùng nhớ trống thỏa yêu cầu

Thuật toán nào cho phép sử dụng bộ nhớ hiệu quả nhất trong trường hợp trên ? =>

Best fit

4. Xét một hệ thống trong đó một chương trình khi được nạp vào bộ nhớ sẽ phân biệt hoàn toàn phân đoạn code và phân đoạn data. Giả sử CPU sẽ xác định được khi nào cần truy xuất lệnh hay dữ liệu, và phải truy xuất ở đâu. Khi đó mỗi chương trình sẽ được cung cấp 2 bộ thanh ghi base-limit: một cho phân đoạn code, và một cho phân đoạn data. Bộ thanh ghi base-limit của phân đoạn code tự động được đặt thuộc tính readonly. Thảo luận các ưu và khuyết điểm của hệ thống này.

- Ưu điểm:

+ Do bộ thanh ghi base limit của phân đoạn code tự động được đặt thuộc tính read only nên chương trình đó có thể được chia sẻ cho nhiều người dùng khác nhau. Kỹ thuật này có thuận lợi trước hết là vì nó là một cơ chế có hiệu quả trong việc chia sẻ lệnh(code) và dữ liệu(data). Một ví dụ là, chỉ duy nhất có một bản sao của editor (trình soạn thảo) hay một compiler (trình biên dịch) cần thiết được giữ trong bộ nhớ, và đoạn code này được chia sẻ bởi tất cả các tiến trình cần truy xuất đến editor code hay compiler code (code của trình soạn thảo hay trình biên dịch).

+ Bảo vệ phần code của chương trình đối với việc sửa đổi có thể gây ra lỗi cho chương trình

- Nhược điểm: Kỹ thuật này gây ra một khuyết điểm là code và dữ liệu bị phân ra thành 2 phần khác nhau, vì trong thực tế 2 thành phần này thường được kết hợp với nhau trong khi phát sinh trình biên dịch cho chương trình

5. Tại sao kích thước trang luôn là lũy thừa của 2 ?

Kích thước trang được định nghĩa bởi phần cứng. Kích thước của một trang điển hình là lũy thừa của 2, từ 512 bytes đến 16MB trên trang, tùy thuộc vào kiến trúc máy tính. Chọn lũy thừa 2 cho kích thước trang để thực hiện việc dịch địa chỉ luận lý thành số trang và độ dời trang rất dễ dàng.

6. Xét một không gian địa chỉ có 8 trang, mỗi trang có kích thước 1K. ánh xạ vào bộ nhớ vật lý có 32 khung trang.

a) Địa chỉ logic gồm bao nhiêu bit ?

$$8 \times 2^{10} = 2^3 \times 2^{10} = 2^{13} \Rightarrow 13 \text{ bit địa chỉ logic}$$

b) Địa chỉ physic gồm bao nhiêu bit ?

$$32 \times 2^{10} = 2^5 \times 2^{10} = 2^{15} \Rightarrow 15 \text{ bit địa chỉ vật lý}$$

7. Tại sao trong hệ thống sử dụng kỹ thuật phân trang, một tiến trình không thể truy xuất đến vùng nhớ không được cấp cho nó ? Làm cách nào hệ điều hành có thể cho phép sự truy xuất này xảy ra ? Hệ điều hành có nên cho phép điều đó không ? Tại sao?

Phân trang là cơ chế quản lý bộ nhớ cho phép không gian địa chỉ vật lý của quá trình là không kề nhau. Phân trang tránh vấn đề đặt vừa khít nhóm bộ nhớ có kích thước thay đổi vào vùng lưu trữ phụ (backing store) mà hầu hết các cơ chế quản lý bộ nhớ trước đó gặp phải. Khi phân đoạn mã và dữ liệu nằm trong bộ nhớ được hoán vị ra, không gian phải được tìm thấy trên vùng lưu trữ phụ.

Bộ nhớ vật lý được chia thành các khối có kích thước cố định gọi là các khung (frames). Bộ nhớ logic cũng được chia thành các khối có cùng kích thước gọi là các trang (pages). Khi một quá trình được thực thi, các trang của nó được nạp vào các khung bộ nhớ sẵn dùng từ vùng lưu trữ phụ. Vùng lưu trữ phụ được chia thành các khối có kích thước cố định và có cùng kích thước như các khung bộ nhớ.

Phân trang là một dạng của tái định vị động. Mỗi địa chỉ logic được giới hạn bởi phần cứng phân trang tới địa chỉ vật lý. Sử dụng phân trang tương tự sử dụng một bảng các thanh ghi nền (hay tái định vị), một thanh ghi cho mỗi khung bộ nhớ.

Một khía cạnh quan trọng của phân trang là sự phân chia rõ ràng giữa tầm nhìn bộ nhớ của người dùng và bộ nhớ vật lý thực sự. Chương trình người dùng nhìn bộ nhớ như một không gian liên tục, chứa chỉ một chương trình. Sự thật, chương trình người dùng được phân bố khắp bộ nhớ vật lý và nó cũng quản lý các quá trình khác. Sự khác nhau giữa tầm nhìn bộ nhớ của người dùng và bộ nhớ vật lý thực sự được làm cho tương thích bởi phần cứng dịch địa chỉ. Địa chỉ logic được dịch thành địa chỉ vật lý. Ánh xạ này được che giấu từ người dùng và được điều khiển bởi hệ điều hành.

Như vậy, quá trình người dùng không thể truy xuất bộ nhớ mà nó không sở hữu. Không có cách định địa chỉ bộ nhớ bên ngoài bảng trang của nó và bảng chỉ chứa những trang mà quá trình sở hữu.

8. 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 200nanoseconds, thì mất bao nhiêu thời gian cho một thao tác truy xuất bộ nhớ trong hệ thống này?

- Để tìm một dữ liệu khi biết địa chỉ logic của nó cần 2 thao tác truy xuất bộ nhớ
 - + Truy xuất vào bảng phân trang để tìm địa chỉ frame vật lý
 - + Khi có được frame, biết địa chỉ vật lý, truy xuất vào bộ nhớ vật lý lấy dữ liệu
- Đề bài cho một lần truy xuất bộ nhớ tốn 200 ns \Rightarrow 2 thao tác sẽ tốn 400ns

b) Nếu sử dụng TLBs với hit-ratio (tỉ lệ tìm thấy) là 75%, thời gian để tìm trong TLBs xem như bằng 0, tính thời gian truy xuất bộ nhớ trong hệ thống (effective memory reference time)

Thời gian truy xuất TLB = $\epsilon = 0$

Thời gian tìm thấy dữ liệu trong bộ nhớ vật lý khi TLB hit (tức tìm thấy chỉ số frame trong TLB) = ϵ + thời gian truy xuất vào bộ nhớ vật lý = $0 + 200 = 200$ ns

Thời gian tìm thấy dữ liệu trong bộ nhớ vật lý khi TLB miss (tức không tìm thấy chỉ số frame trong TLB) = ϵ + thời gian truy xuất vào bảng phân trang + thời gian truy xuất bộ nhớ vật lý = $0 + 200 + 200 = 400$ ns

Tổng thời gian truy xuất bộ nhớ trong hệ thống (effective memory reference time) = $200 \cdot 0.75 + 400 \cdot 0.25 = 250$ ns

9. Nếu cho phép hai phần tử trong bảng trang cùng lưu trữ một số hiệu khung trang trong bộ nhớ thì sẽ có hiệu quả gì ? Giải thích làm cách nào hiệu quả này có thể được sử dụng để giảm thời gian cần khi sao chép một khối lượng lớn vùng nhớ từ vị trí này sang vị trí khác. Khi đó nếu sửa nội dung một trang thì sẽ tác động đến trang còn lại thế nào?

Nếu cho phép 2 phần tử trong bảng trang cùng lưu trữ một số hiệu khung trang trong bộ nhớ thì người dùng có thể chia sẻ code và dữ liệu. Nếu code là reentrant(cho phép bản sao của chương trình có thể được sử dụng đồng thời bởi nhiều tác vụ) thì sẽ có nhiều không gian bộ nhớ được sao lưu trong suốt quá trình chia sẻ sử dụng như text editor, compiler, hệ thống dữ liệu. Chúng ta cần để cho các phần tử trong bảng trang cùng lưu một số hiệu khung trang trong bộ nhớ mà có liên quan đến dữ liệu cần sao chép .Khi đó kỹ thuật này có thể được sử dụng để giảm thời gian khi cần sao chép một khối lượng lớn vùng nhớ từ vị trí này sang vị trí khác.

Tuy nhiên việc chia sẻ các dữ liệu hay code có thuộc tính nonreentrant thì bất kì người dùng nào cũng có thể thao tác lên code hay dữ liệu và có thể chỉnh sửa chúng, đôi khi việc chỉnh sửa này gây ra những sự cố không mong muốn.

Khi sửa đổi nội dung một trang chia sẻ thì các trang còn lại đều bị tác động.

10. Vì sao đôi lúc người ta kết hợp hai kỹ thuật phân trang và phân đoạn ?

Hai kỹ thuật này được kết hợp để có thể hỗ trợ lẫn nhau. Phân đoạn các trang thì có ích trong trường hợp các bảng trang trở nên rất lớn. Một vùng liên tục của bảng trang mà không được sử dụng có thể chia vào trong một mục trong bảng phân đoạn với địa chỉ trang là 0. Phân trang các đoạn xử lý trường hợp có nhiều đoạn dài mà cần nhiều thời gian để cấp phát. Bằng việc phân trang các đoạn, chúng ta giảm thiểu lãng phí bộ nhớ do phân mảnh ngoại cũng như đơn giản quá trình cấp phát.

11. Mô tả cơ chế cho phép một phân đoạn có thể thuộc về không gian địa chỉ của hai tiến trình.

Do bảng phân đoạn là tập hợp các thanh ghi nền và thanh ghi giới hạn(base-limit register), một phân đoạn nào đó có thể được chia sẻ khi một mục trong bảng phân đoạn của 2 tiến trình khác nhau ánh xạ vào cùng một địa chỉ vật lý duy nhất. Các bảng phân đoạn của 2 tiến trình này cần có con trỏ nền cho mỗi bảng phân đoạn, và số hiệu đoạn cần chia sẻ cần giống nhau ở 2 tiến trình

12. Giải thích vì sao chia sẻ một module trong kỹ thuật phân đoạn lại dễ hơn trong kỹ thuật phân trang?

Chia sẻ một module trong kỹ thuật phân đoạn dễ hơn trong kỹ thuật phân trang vì việc phân chia đoạn dựa trên phân chia logic bộ nhớ chứ không phải phân chia vật lý, một phân đoạn với bất kỳ kích thước nào cũng đều có thể chia sẻ chỉ với một mục trong bảng phân đoạn mục này không cần giống nhau ở các bảng phân đoạn khác nhau. Ngược lại, trong kỹ thuật phân trang, để chia sẻ các trang thì số hiệu trang của trang cần chia sẻ cần phải giống nhau trong nhiều bảng trang khác nhau, tức là cần có 1 mục chung cho các bảng trang với mỗi trang cần chia sẻ.

13. Xét bảng phân đoạn sau đây :

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

Cho biết địa chỉ vật lý tương ứng với các địa chỉ logic sau đây :

a. 0,430

Dữ liệu cần tìm ở địa chỉ 0,430: tức đang ở đoạn 0 và ở vị trí thứ 430 trong đoạn 0

Tra vào bảng phân đoạn, đoạn 0 có Limit = 600 và Base = 219:

Kiểm tra thấy $430 < 600 \Rightarrow$ địa chỉ hợp lý

\Rightarrow Địa chỉ vật lý tương ứng $= 219 + 430 = 649$

b. 1,10

$10 < 14 \Rightarrow$ địa chỉ hợp lý

\Rightarrow Địa chỉ vật lý tương ứng $= 2300 + 10 = 2310$

c. 2,500

$500 > 100 \Rightarrow$ địa chỉ không hợp lý

d. 3,400 \Rightarrow

$400 < 580 \Rightarrow$ địa chỉ hợp lý

$$\Rightarrow \text{Địa chỉ vật lý tương ứng} = 1347 + 400 = 1747$$

e. 4,112

112 > 96 \Rightarrow địa chỉ không hợp lý

14. Một process có 7 page, và được cấp 4 frame trống lúc đầu. Xét các giải thuật thay thế trang: FIFO, LRU, OPT với các chuỗi tham chiếu sau. Đánh giá các thuật toán bằng cách chạy nó trên một chuỗi tham chiếu bộ nhớ cụ thể (chuỗi tham chiếu) và tính số lỗi trang trên chuỗi đó

1, 3, 1, 4, 2, 1, 5, 6, 2, 1, 2, 3, 6, 7, 3, 3, 2, 1, 2, 3, 6

FIFO

1	3	1	4	2	1	5	6	2	1	2	3	6	7	3	3	2	1	2	3	6	
1	1	1	1	1	1	5	5	5	5	5	5	5	7	7	7	7	7			7	
	3	3	3	3	3	3	6	6	6	6	6	6	6	6	6	2	2			2	
			4	4	4	4	4	4	1	1	1	1	1	1	1	1	1			6	
				2	2	2	2	2	2	2	3	3	3	3	3	3	3			3	
P	P		P	P		P	P		P		P		P		P				P	11	

LRU

1	3	1	4	2	1	5	6	2	1	2	3	6	7	3	3	2	1	2	3	6	
1	1	1	1	1	1	1	1	1	1	1	1	1	7	7	7	7	7	7	7	6	
	3	3	3	3	3	5	5	5	5	5	3	3	3	3	3	3	3	3	3	3	
			4	4	4	4	6	6	6	6	6	6	6	6	6	6	1	1	1	1	
				2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
p	p		p	p		p	p				p		p				p			p	10

OPT

1	3	1	4	2	1	5	6	2	1	2	3	6	7	3	3	2	1	2	3	6	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6	
	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
			4	4	4	5	6	6	6	6	6	6	7	7	7	7	7	7	7	7	
				2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
p	p		p	p		p	p						p							p	8