**CHƯƠNG 3-BỘ NHỚ**

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

|  |  |
| --- | --- |
| Địa chỉ logic | Địa chỉ Physic |
| Là địa chỉ sinh bởi CPU, tham hiếu đến một  vị trí độc lập với cấu trúc, tổ chức vật lý của  bộ nhớ. Mọi tham chiếu bộ nhớ trong mã lệnh  chương trình được compiler tạo ra đều là địa  chỉ logic.  VD: các tiến trình biên dịch tạo ra mã lệnh  chương trình mà trong đó tham chiếu đến bộ  nhớ đều là địa chỉ logic | Là địa chỉ sinh bởi đơn vị quản lý bộ nhớ  MMU, là một địa chỉ thực trong bộ nhớ  chính, địa chỉ Physic được chuyển từ địa chỉ  logic khi trình biên dịch thực hiện quá trình  linking/loading |

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

|  |  |
| --- | --- |
| Phân mảnh nội vi | Phân mảnh ngoại vi |
| -Là hiện tượng sẽ có vùng nhớ dư thừa khi  ta cấp phát một cùng nhớ hơi lớn hơn kích  thước yêu cầu. VD: Cấp một khoảng trống  24,484 bytes cho một process yêu cầu  24,482 bytes.  - Hiện tượng phân mảnh nội vi thường xảy  ra khi bộ nhớ thực được chia thành các  khối kích thước xảy ra khi bộ nhớ thực  chia thành các khối kích thước cố định và  các process được cấp phát theo đơn vị khối | - Là hiện tượng khi kích thước không gian  nhớ còn trống đủ thỏa moan yêu cầu cấp  phát nhưng không gian nhớ này lại không  lien tục.  - Hiện tượng phân mảnh ngoại vi xảy ra khi  bạn thường xuyên cấp phát cùng nhớ mới,  sau đo xóa đi những phần vùng nhớ đã cấp  phát một cách không thứ tự. |

**3. 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

readonly,nên chương trình đó có thể được chia sẻ cho nhiều người dùng khác nhau.

+ Kĩ 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 thí 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).

+ Một ưu điểm nữa của kĩ thuật này là bảo vệ phần code của chương trình đối với

việc sửa đổi có thế gây ra lỗi cho chương trình.

**- Khuyết điểm:**

Kĩ 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.

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

Kích thước trang luôn là lũy thừa của 2 vì: để thực hiện việc dịch địa chỉ luận lý

thành số trang và độ dời trang rất dễ dàng.

**5. 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?**

- Trong hệ thống sử dụng kỹ thuật phân trang, một tiến trình không thể truy xuất đến

vùng nhớ không được cấp cho nó vì: sự phân chia rõ ràng giữa tầm nhìn bộ nhớ của

người dùng và bộ nhớ vật lý thực sự.

- Cách để hệ điều hành có thể cho phép sự truy xuất này xảy ra: 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 luận lý được dịch thành địa chỉ vật

lý.

- Hệ điều hành có nên cho phép điều đó. Vì bằng cách này, việc có thêm vùng nhớ giúp

người dùng có được quyền hành chi tiết hơn đối với hệ thống bộ nhớ và các phương

**thức truy cập.**

**6. Nếu cho phép hai phần tử trong bảng trang cùng lưu trữ một số hiệu khung trang**

**trong bộ nhớ thì sẽ có hiệu qủa gì ? Giải thích làm cách nào hiệu qủa 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 hai phần tử trong bảng trang cùng lưu trữ một số hiệu khung trang trong

bộ nhớ thì sẽ có hiệu quả: giảm thời gian cần khi sao chép một khối lượng lớn vùng

nhớ từ vị trí này sang vị trí khác.

- Cách làm hiệu qủa 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 bằng việc thay đổi số hiệu khung

trang.

- Khi đó nếu sửa nội dung một trang thì sẽ tác động đến trang còn lại: sẽ được ánh xạ

toàn bộ trên trang còn lại (thực tế không phải là trang còn lại mà là chính nó).

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

- Đôi lúc người ta kết hợp hai kỹ thuật phân trang và phân đoạn vì: 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 bàng 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 lí 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.

**8. Mô tả cơ chế cho phép một phân đoạn có thể thuộc về không gian điạ chỉ của hai**

**tiến trình.**

- Cơ chế cho phép một phân đoạn có thể thuộc về không gian điạ 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 lí 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

**9. 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 lại dễ hơn trong kỹ thuật phân trang vì:

phân đoạn thì dựa trên sự phân chia logic bộ nhớ chứ không phải phân chia vật lý, do

đó những đoạn có kích thước bất kì có thể được chia sẻ với chỉ một mục trong bảng

phân đoạn của tiến trình. Trong phân trang thì bắt buộc phải có một mục chung trong

bảng phân trang cho mỗi trang nhớ được chia sẻ.

**10. Xét bảng phân đoạn sau đây :**

|  |  |  |
| --- | --- | --- |
| Segmen | 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 logique sau đây :

a. 0,430

b. 1,10

c. 2,500

d. 3,400

e. 4,112

a) Địa chỉ vật lý tương ứng = 219 + 430 = 649 (hợp lệ)

b) Địa chỉ vật lý tương ứng = 2300 + 10 = 2310 (hợp lệ)

c) Địa chỉ vật lý tương ứng = 90 +500 =590 (không hợp lệ vì địa chỉ

logic=500>Length=100)

d) Địa chỉ vật lý tương ứng = 1327 + 400 =1727 (hợp lệ)

e) Địa chỉ vật lý tương ứng = 1952 + 112 (không hợp lệ vì địa chỉ

logic=112>Length=96)

14.

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 |  | 5 | 5 |  | 5 |  | 5 |  | 7 |  |  | 7 |  |  |  | 7 |
|  | 3 |  | 3 | 3 |  | 3 | 6 |  | 6 |  | 6 |  | 6 |  |  | 2 |  |  |  | 2 |
|  |  |  | 4 | 4 |  | 4 | 4 |  | 1 |  | 1 |  | 1 |  |  | 1 |  |  |  | 6 |
|  |  |  |  | 2 |  | 2 | 2 |  | 2 |  | 3 |  | 3 |  |  | 3 |  |  |  | 3 |

Page fault: 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 |  | 7 |  |  |  | 7 |  |  | 6 |
|  | 3 |  | 3 | 3 |  | 5 | 5 |  |  |  | 3 |  | 3 |  |  |  | 3 |  |  | 3 |
|  |  |  | 4 | 4 |  | 4 | 6 |  |  |  | 6 |  | 6 |  |  |  | 1 |  |  | 1 |
|  |  |  |  | 2 |  | 2 | 2 |  |  |  | 2 |  | 2 |  |  |  | 2 |  |  | 2 |

Page fault: 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 |  |  |  |  |  |  | 6 |
|  | 3 |  | 3 | 3 |  | 3 | 3 |  |  |  |  |  | 3 |  |  |  |  |  |  | 3 |
|  |  |  | 4 | 4 |  | 5 | 6 |  |  |  |  |  | 7 |  |  |  |  |  |  | 7 |
|  |  |  |  | 2 |  | 2 | 2 |  |  |  |  |  | 2 |  |  |  |  |  |  | 2 |

Page fault: 8