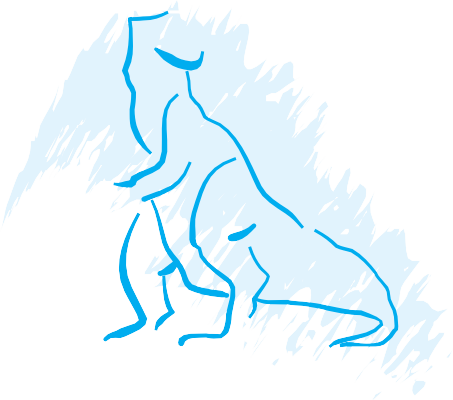
## Đồng bộ hóa quy trình



C H*6*APTER

### Bài tập thực hành

* 1. Trong Phần 6.4, chúng tôi đã đề cập rằng việc tắt ngắt thường xuyên có thể ảnh hưởng đến đồng hồ của hệ thống. Giải thích tại sao nó có thể và làm thế nào những tác động đó có thể giảm thiểu.

**Trả lời:** Đồng hồ hệ thống được cập nhật tại mỗi lần ngắt đồng hồ. Nếu các ngắt bên trong bị vô hiệu hóa — đặc biệt là trong một khoảng thời gian dài — đồng hồ hệ thống có thể dễ dàng mất thời gian chính xác. Đồng hồ hệ thống cũng được sử dụng cho mục đích lập lịch trình. Ví dụ, lượng tử thời gian cho một quá trình được biểu thị bằng một số tích tắc đồng hồ. Tại mỗi lần ngắt đồng hồ, bộ lập lịch sẽ xác định xem lượng tử thời gian cho hiện tại đang chạy quy trình có hết hạn. Nếu cái đồng hồ làm gián đoạn là bị vô hiệu hóa Người lập kế hoạch có thể không phải chính xác chỉ định thời gian lượng tử. Cái này hiệu ứng có thể được giảm thiểu bằng cách tắt các ngắt đồng hồ chỉ trong thời gian rất ngắn Chu kỳ.

* 1. *Các Người hút thuốc lá Vấn đề* . Xem xét Một hệ thống với ba *người hút thuốc* quy trình và một *đại lý* quy trình. Mỗi người hút thuốc liên tục Bánh cuốn Một thuốc lá và sau đó hút nó. Nhưng để cuộn và hút một điếu thuốc, người hút thuốc cần ba thành phần: thuốc lá, giấy và diêm. Một trong số những người hút thuốc có giấy, một người khác có thuốc lá, và chiếc thứ ba có que diêm. Đại lý có nguồn cung cấp vô hạn của cả ba nguyên liệu. Người đại diện đặt hai trong số các thành phần trên bàn. Người hút thuốc có thành phần còn lại sau đó chế biến và hút một điếu thuốc, báo hiệu tác nhân đã hoàn thành. Sau đó, đại lý đưa ra hai trong số ba thành phần khác, và chu kỳ lặp lại. Viết chương trình để đồng bộ hóa tác nhân và người hút thuốc bằng cách sử dụng đồng bộ hóa Java.

**Trả lời:** Xin vui lòng tham khảo đến các hỗ trợ Web Địa điểm vì nguồn mã số giải pháp.

* 1. Đưa ra lý do tại sao Solaris, Windows XP và Linux thực hiện mul- tiple khóa các cơ chế. Diễn tả các trường hợp Dưới cái mà họ

**17**

**18 Chương 6 Quy trình Đồng bộ hóa**

sử dụng spinlocks, mutexes, semaphores, mutex thích ứng và các biến điều kiện. Trong mỗi trường hợp, hãy giải thích lý do tại sao cơ chế đó là cần thiết.

**Trả lời:** Các hệ điều hành này cung cấp các cơ chế khóa khác nhau tùy thuộc vào nhu cầu của nhà phát triển ứng dụng. Spinlocks rất hữu ích vì đa xử lý hệ thống ở đâu Một chủ đề có thể chạy trong Một vòng lặp bận rộn (trong một khoảng thời gian ngắn) thay vì phải chịu chi phí đưa vào hàng đợi ngủ. Mutexes rất hữu ích để khóa tài nguyên. Solaris 2 sử dụng mutex thích ứng, có nghĩa là mutex được thực hiện với một khóa quay trên các máy đa xử lý. Semaphores và các biến thể điều kiện là những công cụ thích hợp hơn để đồng bộ hóa khi tài nguyên phải là được tổ chức vì Một Dài giai đoạn = Stage của thời gian, từ quay tròn Là không hiệu quả vì một thời gian dài.

* 1. Giải thích sự khác biệt, về mặt chi phí, giữa ba loại lưu trữ dễ bay hơi, không dễ bay hơi và ổn định.

**Trả lời:** Bộ nhớ linh hoạt đề cập đến bộ nhớ chính và bộ nhớ đệm và rất nhanh. Tuy nhiên, lưu trữ dễ bay hơi không thể tồn tại khi hệ thống gặp sự cố hoặc sập nguồn. Lưu trữ không bay hơi vẫn tồn tại sự cố hệ thống và hệ thống bị sập nguồn. Đĩa và băng là những ví dụ về lưu trữ không bay hơi. Gần đây, các thiết bị USB sử dụng bộ nhớ chỉ đọc chương trình có thể xóa được ( EPROM ) đã xuất hiện cung cấp khả năng lưu trữ không linh hoạt. Stor- age ổn định đề cập đến đến kho điều đó về mặt kỹ thuật có thể *không bao giờ* là mất bằng ở đó Chúng tôi redun- dant các bản sao lưu dữ liệu (thường là trên đĩa).

* 1. Giải thích mục đích của cơ chế điểm kiểm tra. Các điểm kiểm tra nên được thực hiện bao lâu một lần? Mô tả tần suất của các điểm kiểm tra ảnh hưởng như thế nào:
     + Hiệu suất hệ thống khi không có lỗi xảy ra
     + Thời gian cần để khôi phục từ một hệ thống tai nạn
     + Thời gian cần để khôi phục từ đĩa tai nạn

**Trả lời:** Bản ghi nhật ký điểm kiểm tra chỉ ra rằng một bản ghi nhật ký và nó đã được sửa đổi dữ liệu có đã bằng văn bản đến ổn định kho và điều đó các nhu cầu giao dịch không phải đến là làm lại trong trường hợp của Một hệ thống tai nạn. Chắc chắn, các hơn thường xuyên thực hiện các điểm kiểm tra, càng ít có khả năng các bản cập nhật thừa sẽ phải được thực hiện trong quá trình khôi phục quy trình.

* + - Hiệu suất hệ thống khi không xảy ra lỗi — Nếu không có lỗi nào xảy ra, hệ thống phải chịu chi phí thực hiện các trạm kiểm soát về cơ bản là không cần thiết. Trong tình huống này, việc thực hiện các trạm kiểm soát ít thường xuyên hơn sẽ dẫn đến hệ thống tốt hơn màn biểu diễn.
    - Thời gian cần thiết để khôi phục sau sự cố hệ thống — Sự tồn tại của một bản ghi điểm kiểm tra có nghĩa là một hoạt động sẽ không phải là

làm lại trong quá trình khôi phục hệ thống. Trong tình huống này, các trạm kiểm soát được thực hiện càng thường xuyên thì thời gian khôi phục sau sự cố hệ thống càng nhanh.

* + - Thời gian cần để khôi phục sau sự cố đĩa — Sự tồn tại của bản ghi điểm kiểm tra có nghĩa là một thao tác sẽ không phải thực hiện lại trong quá trình khôi phục hệ thống. Trong tình huống này, càng thường

**Thực hành Bài tập 19**

các trạm kiểm soát được thực hiện, thời gian khôi phục càng nhanh sau sự cố đĩa.

* 1. Giải thích khái niệm giao dịch tính nguyên tử.

**Trả lời:** Một giao dịch là một chuỗi các thao tác đọc và ghi dựa trên một số dữ liệu theo sau qua Một làm hoạt động. Nếu các hàng loạt của hoạt động trong một Giao dịch không thể là hoàn thành, các Giao dịch cần phải là bỏ dở và các hoạt động đã diễn ra phải được quay trở lại. Điều quan trọng là chuỗi hoạt động trong một giao dịch xuất hiện dưới dạng một hoạt động không thể phân chia để đảm bảo tính toàn vẹn của dữ liệu được cập nhật. Nếu không, dữ liệu có thể bị xâm phạm nếu hoạt động từ hai (hoặc nhiều) giao dịch khác nhau trộn lẫn.

* 1. Cho thấy rằng một số lịch biểu có thể thực hiện được theo giao thức khóa hai pha nhưng không thể thực hiện được theo giao thức dấu thời gian và ngược lại ngược lại. **Trả lời:** Lịch biểu được phép trong giao thức khóa hai pha nhưng không được phép trong giao thức dấu thời gian Là:

|  |  |  |  |
| --- | --- | --- | --- |
| bươc chân | *T* 0 | *T* 1 | Quyền ưu tiên |
| 1 | **lock-S** ( *A* ) |  |  |
| 2 | **đọc** ( *A* ) |  |  |
| 3 |  | **lock-X** ( *B* ) |  |
| 4 |  | **ghi** ( *B* ) |  |
| 5 |  | **mở khóa** ( *B* ) |  |
| 6 | **lock-S** ( *B* ) |  |  |
| 7  số 8 | **đọc** ( *B* )  **mở khóa** ( *A* ) |  | *T* 1 → *T* 0 |

9 **mở khóa** ( *B* )

Lịch trình này không được phép trong giao thức dấu thời gian vì ở bước 7, dấu thời gian W của *B* là 1.

Lịch biểu được phép trong giao thức dấu thời gian nhưng không được phép trong giao thức khóa hai pha là:

|  |  |  |  |
| --- | --- | --- | --- |
| bươc chân | *T* 0 | *T* 1 | *T* 2 |
| 1 | **ghi** ( *A* ) |  |  |
| 2 |  | **ghi** ( *A* ) |  |
| 3 |  |  | **ghi** ( *A* ) |
| 4 | **ghi** ( *B* ) |  |  |
| 5 |  | **ghi** ( *B* ) |  |

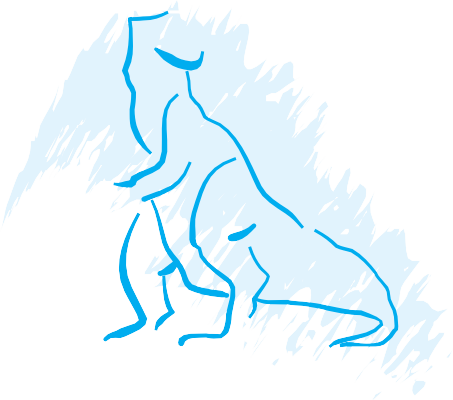
Cái này lịch trình không thể có Khóa hướng dẫn thêm đến làm nó hợp pháp dưới hai giai đoạn khóa giao thức bởi vì *T* 1 cần phải mở khóa ( *A* ) giữa các bước 2 và 3, và phải khóa ( *B* ) giữa các bước 4 và 5.

* 1. Các Chờ đã() tuyên bố trong tất cả Java chương trình ví dụ là phần của Một vòng lặp while . Giải thích lý do tại sao bạn luôn cần sử dụng câu lệnh while khi sử dụng wait () và tại sao bạn không bao giờ sử dụng câu lệnh if . **Trả lời:** Đây là một vấn đề quan trọng cần nhấn mạnh! Java chỉ cung cấp thông báo ẩn danh —bạn không thể thông báo cho một chuỗi nhất định rằng một chứng nhận

**20 Chương 6 Quy trình Đồng bộ hóa**

điều kiện tain là đúng. Khi một luồng được thông báo, nó có trách nhiệm kiểm tra lại điều kiện mà nó đang chờ. Nếu một chuỗi không kiểm tra lại điều kiện, nó có thể đã nhận được thông báo mà không đáp ứng điều kiện.

# Bế tắc



C H7APTER

### Thực hành Bài tập

* 1. Liệt kê ba ví dụ về các deadlock không liên quan đến môi trường hệ thống máy tính.

#### Trả lời:

* + - Hai xe ô tô băng qua cầu một làn ngược chiều hướng.
    - Một người đi xuống một cái thang trong khi một người khác đang leo lên thang.
    - Hai đoàn tàu đi về phía nhau trên cùng một theo dõi.
    - Hai người thợ mộc phải giã móng tay. Chỉ có một cái búa và một cái gầu đóng đinh. Bế tắc xảy ra nếu một người thợ mộc có chiếc búa và người thợ mộc khác có móng tay.
  1. Giả sử rằng một hệ thống đang ở trạng thái không an toàn. Cho thấy rằng các tiến trình có thể hoàn thành việc thực thi của chúng mà không đi vào trạng thái bế tắc.

**Trả lời:** Trạng thái không an toàn có thể không nhất thiết dẫn đến deadlock, nó chỉ có nghĩa là chúng ta không thể đảm bảo rằng deadlock sẽ không xảy ra. Do đó, có thể một hệ thống ở trạng thái không an toàn vẫn có thể cho phép tất cả các tiến trình hoàn thành mà không xảy ra bế tắc. Xem xét tình huống mà một hệ thống có 12 tài nguyên phân bổ giữa quy trình *P* 0 , *P* 1 , và *P* 2 . Các nguồn lực được phân bổ theo những điều sau chính sách:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Max | Hiện hành | Nhu cầu |
| *P* 0 | 10 | 5 | 5 |
| *P* 1 | 4 | 2 | 2 |
| *P* 2 | 9 | 3 | 6 |

**21**

**22 chương 7 Bế tắc**

for (int i = 0; i <n; i ++) *{*

// đầu tiên tìm một luồng có thể kết thúc for (int j = 0; j <n; j ++) *{*

if (! finish [j]) *{*

boolean temp = true;

for (int k = 0; k <m; k ++) *{*

if (need [j] [k]> work [k]) temp = false;

*}*

if (temp) *{* // nếu luồng này có thể kết thúc xong [j] = true;

for (int x = 0; x <m; x ++) work [x] + = work [j] [x];

*}*

*}*

*}*

*}* **Hình 7.1** An toàn thuật toán của ngân hàng thuật toán.

Hiện tại có hai tài nguyên có sẵn. Hệ thống này ở trạng thái không an toàn như quy trình *P* 1 có thể hoàn thành, do đó giải phóng tổng cộng bốn tài nguyên. Nhưng chúng tôi không thể đảm bảo rằng các quy trình *P* 0 và *P* 2 có thể kết hợp. Tuy nhiên, nó Là có thể được điều đó Một quy trình có thể phóng thích tài nguyên trước khi yêu cầu bất cứ thêm nữa. Vì ví dụ, quy trình *P* 2 có thể phóng thích Một tài nguyên, do đó tăng các toàn bộ con số của tài nguyên đến năm. Cái này cho phép tỷ lệ *P* 0 để hoàn thành, điều này sẽ giải phóng tổng cộng chín tài nguyên, do đó cho phép quá trình *P* 2 hoàn thành như ổn.

* 1. Chứng minh rằng thuật toán an toàn được trình bày trong Phần 7.5.3 yêu cầu một thứ tự của *m n* 2 các hoạt động.

×

#### Trả lời:

Hình 7.1 cung cấp mã Java thực thi thuật toán an toàn của thuật toán của ngân hàng (việc triển khai hoàn chỉnh thuật toán của ngân hàng có sẵn với tải xuống mã nguồn).

Như có thể thấy, các vòng ngoài lồng nhau — trong đó vòng lặp qua *n* lần — cung cấp hiệu suất *n* 2 . Trong các vòng ngoài này là hai vòng tuần tự bên trong vòng lặp cái mà vòng *m* lần. Các ồ của cái này do đó thuật toán là *O* ( *m* × *\ n \ t2* ) .

* 1. Hãy xem xét một hệ thống máy tính chạy 5.000 công việc mỗi tháng mà không có kế hoạch ngăn chặn bế tắc hoặc tránh bế tắc. Bế tắc xảy ra về hai lần mỗi tháng, và các nhà điều hành cần phải chấm dứt và chạy lại khoảng 10 công việc cho mỗi lần bế tắc. Mỗi công việc trị giá khoảng 2 đô la (tính theo thời gian CPU ) và các công việc bị chấm dứt có xu hướng được hoàn thành một nửa khi chúng bị hủy bỏ.

Một nhà lập trình hệ thống đã ước tính rằng một thuật toán tránh bế tắc (giống như thuật toán của ngân hàng) có thể được cài đặt trong hệ thống với một tăng lên trong các Trung bình cộng chấp hành thời gian mỗi Công việc của xung quanh 10 phần trăm. Từ các máy móc Hiện nay có 30 phần trăm nhàn rỗi thời gian, tất cả 5.000 việc làm vẫn có thể chạy mỗi tháng, mặc dù thời gian quay vòng sẽ tăng khoảng 20 phần trăm vào Trung bình cộng.

**Thực hành Bài tập 23**

1. Các đối số để cài đặt thuật toán tránh bế tắc là gì?
2. Gì Chúng tôi các tranh luận chống lại cài đặt các thuật toán tránh bế tắc?

**Trả lời:** Một đối số để cài đặt tính năng tránh bế tắc trong hệ thống là điều đó chúng tôi có thể chắc chắn bế tắc sẽ không bao giờ xảy ra. Ở trong phép cộng, mặc dù tăng lên trong quay lại thời gian, tất cả 5.000 việc làm có thể còn chạy.

Một lập luận chống lại việc cài đặt phần mềm tránh deadlock là các deadlock xảy ra không thường xuyên và chúng tốn rất ít chi phí khi chúng xảy ra.

* 1. Có thể Một hệ thống phát hiện điều đó một số của nó là quy trình Chúng tôi chết đói? Nếu bạn trả lời " có " , giải thích làm thế nào nó có thể. Nếu bạn trả lời " không " , hãy giải thích cách hệ thống có thể đối phó với nạn đói vấn đề.

**Trả lời:** Chết đói Là Một khó khăn đề tài đến định nghĩa bằng nó có thể bần tiện những thứ khác nhau cho các hệ thống khác nhau. Đối với mục đích của câu hỏi này, chúng tôi sẽ định nghĩa chết đói là tình huống theo đó một quy trình phải đợi sau một khoảng thời gian hợp lý — có thể là vô thời hạn — trước khi nhận được tài nguyên được yêu cầu. Một cách để phát hiện nạn đói trước tiên là xác định Một giai đoạn = Stage của thời gian- *T* -điều đó Là xem xét không hợp lý. Khi nào một tiến trình yêu cầu Một nguồn, Một hẹn giờ Là đã bắt đầu. Nếu các trôi qua thời gian vượt quá *T* , thì quá trình này được coi là chết đói.

Một chiến lược để đối phó với nạn đói là áp dụng một chính sách trong đó các nguồn lực chỉ được giao cho quá trình được chờ đợi lâu nhất. Ví dụ, nếu quy trình *P a* đã đợi lâu hơn cho nguồn *X* hơn quy trình *P b* , các yêu cầu từ quy trình *P b* sẽ là hoãn lại cho đến khi quá trình *P a* yêu cầu của đã được thỏa mãn.

Một chiến lược khác sẽ ít nghiêm ngặt hơn những gì vừa được đề cập. Trong này kịch bản, Một nguồn có thể là được cấp đến Một quy trình điều đó có đã đợi ít hơn một quy trình khác, với điều kiện là quy trình khác không bị chết đói. Tuy nhiên, nếu một quá trình khác được coi là đang chết đói, yêu cầu của nó sẽ được đáp ứng đầu tiên.

* 1. Xem xét các tiếp theo phân bổ tài nguyên chính sách. Yêu cầu và cho phép phát hành tài nguyên bất cứ lúc nào. Nếu không thể đáp ứng yêu cầu về tài nguyên vì tài nguyên không có sẵn, thì chúng tôi sẽ kiểm tra mọi quy trình điều đó Chúng tôi bị chặn, chờ vì tài nguyên. Nếu họ có các tài nguyên mong muốn, sau đó này tài nguyên Chúng tôi Lấy xa từ họ và Chúng tôi đưa ra cho quá trình yêu cầu. Vectơ tài nguyên mà quá trình đang chờ được tăng lên để bao gồm các tài nguyên đã được sử dụng xa.

Ví dụ: hãy xem xét một hệ thống có ba loại tài nguyên và vectơ *Có sẵn* khởi tạo đến (4,2,2). Nếu quy trình *P* 0 hỏi vì (2,2,1), nó lấy chúng. Nếu *P* 1 yêu cầu (1,0,1), nó nhận được chúng. Sau đó, nếu *P* 0 yêu cầu (0,0,1), nó bị chặn (tài nguyên không có sẵn). Nếu *P* 2 bây giờ yêu cầu (2,0,0), nó có sẵn một (1,0,0) và một điều đó là phân bổ đến *P* 0 (từ *P* 0 Là bị chặn). *P* 0 's *Phân bổ* vectơ đi xuống đến (1,2,1) và nó là *Nhu cầu* vectơ đi hướng lên thành (1,0,1).

1. Có thể xảy ra bế tắc không? Nếu bạn trả lời " có " , hãy đưa ra một ví dụ. Nếu bạn trả lời " không " , hãy nêu rõ điều kiện cần thiết nào không thể xảy ra.
2. Có thể xảy ra chặn vô thời hạn không? Giải thích của bạn trả lời.

**24 chương 7 Bế tắc**

#### Trả lời:

1. Bế tắc không thể xảy ra vì quyền ưu tiên tồn tại.
2. Vâng. MỘT quy trình có thể không bao giờ có được, thu được tất cả các tài nguyên nó nhu cầu nếu họ liên tục bị tấn công bởi một loạt các yêu cầu, chẳng hạn như yêu cầu của quá trình *C.* \_
   1. Giả sử rằng bạn đã mã hóa thuật toán an toàn tránh bế tắc và bây giờ được yêu cầu triển khai thuật toán phát hiện bế tắc. Bạn có thể làm như vậy chỉ bằng cách sử dụng mã thuật toán an toàn và xác định lại *Ma x tôi* = *Waiti ng tôi* + *Alloca tion tôi* , ở đâu *Waiti ng tôi* Là Một vectơ chỉ định quy trình tài nguyên mà *tôi* đang chờ đợi và *Alloca tion tôi* như được định nghĩa trong Phần 7.5? Giải thích của bạn trả lời.

#### Trả lời:

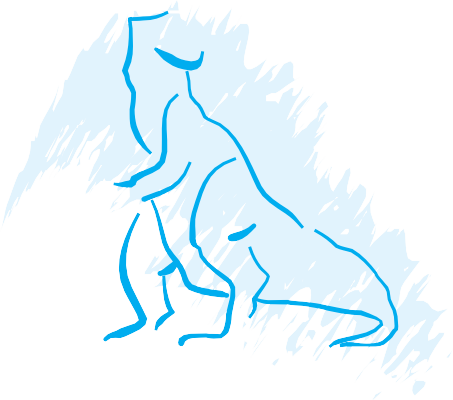
Vâng. Vectơ *Max* đại diện cho yêu cầu tối đa mà một quy trình có thể thực hiện. Khi tính toán thuật toán an toàn, chúng tôi sử dụng ma trận *Cần ,* đại diện *Max - Phân bổ* . Khác đường đến nghĩ của cái này Là *Max*

*= Nhu cầu + Phân bổ* . Theo đến các câu hỏi, các *Chờ* ma trận hoàn thành một vai diễn giống đến các *Nhu cầu* ma trận, vì thế *Max = Chờ + Phân bổ* .

* 1. Có thể có một bế tắc chỉ liên quan đến một quy trình duy nhất không? Giải thích của bạn trả lời.

**Trả lời:** Không. Điều này xảy ra trực tiếp từ điều kiện giữ và chờ.

*Quản lý bộ nhớ*



C H*8*APTER

### Bài tập thực hành

* 1. Đặt tên cho hai điểm khác biệt giữa địa chỉ logic và địa chỉ vật lý. **Trả lời:** Một địa chỉ logic không tham chiếu đến một địa chỉ thực hiện có; hơn, nó đề cập đến đến một trừu tượng Địa chỉ trong một trừu tượng Địa chỉ khoảng trống. Sự tương phản cái này với Một thuộc thân thể Địa chỉ điều đó đề cập đến đến một thật sự thuộc thân thể địa chỉ trong bộ nhớ. Một địa chỉ logic được tạo ra bởi CPU và được đơn vị quản lý bộ nhớ ( MMU ) dịch thành địa chỉ vật lý . Do đó, các địa chỉ vật lý được tạo bởi MMU .
  2. Hãy xem xét một hệ thống trong đó một chương trình có thể được tách thành hai phần: mã và dữ liệu. CPU biết liệu nó muốn một lệnh (tìm nạp trong cấu trúc) hay dữ liệu (tìm nạp hoặc lưu trữ dữ liệu) . Do đó, hai cặp thanh ghi giới hạn cơ sở được cung cấp: một cho lệnh và một cho dữ liệu. Cặp thanh ghi giới hạn-cơ sở lệnh tự động ở chế độ chỉ đọc, do đó có thể là đã chia sẻ giữa khác nhau người dùng. Bàn luận các thuận lợi và nhược điểm của điều này kế hoạch.

**Trả lời:** Ưu điểm chính của chương trình này là nó là một cơ chế hiệu quả vì mã số và dữ liệu sự chia sẻ. Vì ví dụ, chỉ một một sao chép của trình soạn thảo hoặc trình biên dịch cần được lưu trong bộ nhớ và mã này có thể được chia sẻ bởi tất cả các quy trình cần quyền truy cập vào trình biên tập hoặc mã trình biên dịch. Khác thuận lợi Là sự bảo vệ của mã số chống lại sai sự sửa đổi. Điểm bất lợi duy nhất là mã và dữ liệu phải được tách biệt, thường được gắn vào trong trình biên dịch tạo mã số.

* 1. Tại sao kích thước trang luôn có giá trị 2?

**Trả lời:** Nhớ lại rằng phân trang được thực hiện bằng cách chia nhỏ một địa chỉ thành một trang và số hiệu. Hiệu quả nhất là chia địa chỉ thành các bit trang X và bit bù Y, thay vì thực hiện số học trên Địa chỉ đến tính toán các trang con số và bù lại. Tại vì mỗi chút

**25**

**26 Chương 8 Trí nhớ Sự quản lý**

Chức vụ đại diện Một sức mạnh của 2, tách ra một Địa chỉ giữa chút ít dẫn đến kích thước trang là sức mạnh của 2.

* 1. Hãy xem xét một không gian địa chỉ logic gồm tám trang, mỗi trang 1024 từ, được ánh xạ vào một bộ nhớ vật lý gồm 32 khung.
     1. Có bao nhiêu bit trong lôgic Địa chỉ?
     2. Có bao nhiêu bit trong vật lý Địa chỉ?

#### Trả lời:

1. Địa chỉ logic: 13 chút ít
2. Địa chỉ thực: 15 chút ít
   1. Tác dụng của việc cho phép hai mục trong bảng trang trỏ đến cùng một khung trang trong bộ nhớ là gì? Giải thích cách sử dụng hiệu ứng này để giảm lượng thời gian cần thiết để sao chép một lượng lớn bộ nhớ từ một địa điểm đến khác. Gì hiệu ứng sẽ đang cập nhật một số byte trên trang này có trang kia trang?

**Trả lời:** Bằng cách cho phép hai mục nhập trong một bảng trang trỏ đến cùng một trang khung trong kỉ niệm, người dùng có thể đăng lại mã số và dữ liệu. Nếu các mã số Là reen- trant, nhiều kỉ niệm khoảng trống có thể là lưu lại bởi vì các đã chia sẻ sử dụng của các chương trình lớn như trình soạn thảo văn bản, trình biên dịch và hệ thống cơ sở dữ liệu. “ Sao chép ” một lượng lớn bộ nhớ có thể được thực hiện bằng cách để các bảng trang khác nhau trỏ đến cùng một bộ nhớ vị trí.

Tuy nhiên, việc chia sẻ mã hoặc dữ liệu không liên quan có nghĩa là bất kỳ người dùng nào có quyền truy cập vào mã đều có thể sửa đổi nó và những sửa đổi này sẽ được phản ánh trong “ bản sao của người dùng khác. ”

* 1. Diễn tả Một cơ chế qua cái mà một bộ phận có thể thuộc về đến các không gian địa chỉ của hai khác nhau các quy trình.

**Trả lời:** Vì bảng phân đoạn là tập hợp các thanh ghi giới hạn cơ sở, các phân đoạn có thể là đã chia sẻ khi nào mục trong các bộ phận chiếc bàn của hai các công việc khác nhau chỉ đến cùng một vị trí thực tế. Hai bảng phân đoạn phải có con trỏ cơ sở giống hệt nhau và số phân đoạn được chia sẻ phải giống nhau trong hai các quy trình.

* 1. Chia sẻ phân đoạn giữa quy trình không có yêu cầu các tương tự số phân đoạn có thể có trong phân đoạn được liên kết động hệ thống.
     1. Định nghĩa Một hệ thống điều đó cho phép tĩnh liên kết và chia sẻ của phân đoạn mà không yêu cầu số phân đoạn là tương tự.
     2. Mô tả một lược đồ phân trang cho phép các trang được chia sẻ mà không yêu cầu số trang là tương tự.

**Trả lời:** Cả hai vấn đề này đều giảm thành chương trình có thể tham chiếu cả hai nó là làm chủ mã số và nó là dữ liệu không có biết các bộ phận hoặc số trang được liên kết với địa chỉ. MULTICS đã giải quyết vấn đề này bằng cách liên kết bốn thanh ghi với mỗi quá trình. Một sổ đăng ký có địa chỉ của các hiện hành chương trình bộ phận, khác có Một cơ sở Địa chỉ cho cây rơm, khác có Một cơ sở Địa chỉ vì các toàn cầu dữ liệu, và Vì thế trên. Ý tưởng Là điều đó tất cả người giới thiệu có đến là gián tiếp bởi vì Một Đăng ký điều đó bản đồ tới các hiện hành bộ phận hoặc trang con số. Qua thay đổi này đăng ký, các

**Thực hành Bài tập 27**

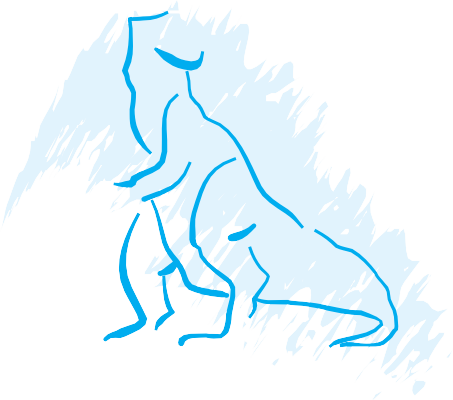
cùng một mã có thể thực thi cho các quy trình khác nhau mà không có cùng số trang hoặc phân đoạn.

* 1. Trong IBM / 370 , bảo vệ bộ nhớ được cung cấp thông qua việc sử dụng các *phím.* MỘT Chìa khóa Là Một 4-bit số lượng. Mỗi 2K khối của kỉ niệm có Một Chìa khóa (các khóa lưu trữ) được liên kết với nó. CPU cũng có một chìa khóa (chìa khóa bảo vệ) đi kèm với nó. Hoạt động cửa hàng chỉ được phép nếu cả hai khóa bằng nhau hoặc nếu một trong hai khóa bằng không. Sơ đồ quản lý bộ nhớ nào sau đây có thể được sử dụng thành công với điều này phần cứng?
     1. Máy trần
     2. Một người dùng hệ thống
     3. Đa chương trình với một số lượng cố định quy trình
     4. Đa chương trình với một số biến quy trình
     5. Phân trang
     6. Phân đoạn

#### Trả lời:

1. Bảo vệ không cần thiết, hãy đặt khóa hệ thống thành 0.
2. Đặt khóa hệ thống thành 0 khi ở trong trình giám sát chế độ.
3. Vùng đất kích thước cần phải là đã sửa trong gia số của 2k byte, chỉ định chìa khóa với bộ nhớ các khối.
4. Giống như ở trên.
5. Kích thước khung hình phải theo gia số 2k byte, phân bổ khóa với các trang.
6. Bộ phận kích thước cần phải là trong gia số của 2k byte, chỉ định Chìa khóa với các phân đoạn.

## Bộ nhớ ảo



C H*9*APTER

### Bài tập thực hành

* 1. Lỗi trang xảy ra trong những trường hợp nào? Mô tả các hành động được thực hiện bởi hệ điều hành khi một trang bị lỗi xảy ra.

**Trả lời:** MỘT trang lỗi xảy ra khi nào một truy cập đến Một trang điều đó có không phải được đưa vào bộ nhớ chính diễn ra. Hệ điều hành xác minh quyền truy cập bộ nhớ, hủy bỏ chương trình nếu nó không hợp lệ. Nếu nó hợp lệ, một khung miễn phí được đặt và I / O được yêu cầu để đọc trang cần thiết vào miễn phí khung. Trên hoàn thành của I / O , các quy trình chiếc bàn và trang bảng được cập nhật và hướng dẫn là đã khởi động lại.

* 1. Giả sử rằng bạn có một chuỗi tham chiếu trang cho một quy trình với *m* khung (ban đầu tất cả đều trống). Chuỗi tham chiếu trang có độ dài *p* ; *n* khác biệt trang con số xảy ra trong nó. Trả lời này câu hỏi vì bất cứ trang- thay thế thuật toán:
     1. Giới hạn dưới trên số trang là gì lỗi?
     2. Giới hạn trên trên số trang là gì lỗi?

#### Trả lời:

1. *n*
2. *P*
   1. Kỹ thuật và cấu trúc lập trình nào sau đây là “ tốt ” cho môi trường phân trang theo yêu cầu? Cái nào là " không tốt " ? Ex- rõ ràng của bạn các câu trả lời.
      1. Cây rơm
      2. Ký hiệu băm chiếc bàn

**29**

**30 Chương 9 Ảo Kỉ niệm**

* + 1. Tuần tự Tìm kiếm
    2. Tìm kiếm nhị phân
    3. Mã thuần túy
    4. toán vectơ
    5. Chuyển hướng

#### Trả lời:

1. Cây rơm -tốt.
2. Bảng ký hiệu băm — not tốt.
3. Tuần tự tìm kiếm — tốt.
4. Tìm kiếm nhị phân — không phải tốt.
5. Nguyên chất mã — tốt.
6. toán vectơ -tốt.
7. Hướng dẫn — không phải tốt.
   1. Hãy xem xét các thuật toán thay thế trang sau đây. Xếp hạng các nhịp điệu này trên thang điểm năm từ “ xấu ” đến “ hoàn hảo ” theo Lỗi trang tỷ lệ. Tách rời những, cái đó thuật toán điều đó chịu đựng từ Sự bất thường của Belady so với những điều đó không phải.
      1. LRU sự thay thế
      2. FIFO sự thay thế
      3. Thay thế tối ưu
      4. Cơ hội thứ hai sự thay thế

#### Trả lời:

|  |  |  |
| --- | --- | --- |
| Cấp | Thuật toán | Chịu đựng sự bất thường của Belady |
| 1 | Tối ưu | không |
| 2 | LRU | không |
| 3 | Cơ hội thứ hai | Vâng |
| 4 | FIFO | Vâng |

* 1. Khi bộ nhớ ảo được triển khai trong một hệ thống máy tính, chắc chắn chi phí có liên quan với các kỹ thuật và chắc chắn những lợi ích. Danh sách chi phí và các những lợi ích. Là nó có thể được vì các chi phí đến quá các những lợi ích? Nếu nó Là, Cái gì đo có thể là Lấy đến chắc chắn điều đó cái này làm không phải xảy ra? **Trả lời:** Các chi phí Chúng tôi thêm vào phần cứng và Chậm hơn truy cập thời gian. Những lợi ích Chúng tôi tốt sử dụng của kỉ niệm và lớn hơn hợp lý Địa chỉ không gian hơn địa chỉ thực khoảng trống.
  2. bộ nhớ ảo được phân trang , sử dụng bộ xử lý trung tâm với thời gian chu kỳ là 1 micro giây. Nó tốn thêm 1 micro giây đến truy cập Một trang khác hơn các hiện hành một. Các trang có 1000

**Thực hành Bài tập 31**

từ ngữ, và các phân trang thiết bị Là Một cái trống điều đó xoay vòng tại 3000 số vòng quay mỗi phút và chuyển 1 triệu từ mỗi giây. Các phép đo thống kê sau đây thu được từ hệ thống:

* 1 phần trăm của tất cả các hướng dẫn được thực thi đã truy cập vào một trang không phải hiện tại trang.
* Của các hướng dẫn điều đó đã truy cập khác trang, 80 phần trăm đã truy cập một trang đã có trong kỉ niệm.
* Khi một trang mới được yêu cầu, trang được thay thế đã được sửa đổi 50 phần trăm thời gian.

Tính toán các có hiệu lực hướng dẫn thời gian trên cái này hệ thống, giả định điều đó hệ thống Là đang chạy một quy trình chỉ một và điều đó các bộ xử lý Là nhàn rỗi trong quá trình chuyển trống.

#### Trả lời:

có hiệu lực truy cập thời gian = 0,99 × (1 µ giây + 0,008 × (2 µ giây)

+ 0,002 × (10.000 µ giây + 1.000 µ giây)

+ 0,001 × (10.000 µ giây + 1.000 µ giây)

= (0,99 + 0,016 + 22.0 + 11.0) µ giây

= 34,0 µ giây

* 1. Xem xét mảng hai chiều A :

int A [] [] = new int [100] [100];

ở đâu A [0] [0] Là tại vị trí 200, trong Một phân trang kỉ niệm hệ thống với các trang có kích thước 200. Một quy trình nhỏ ở trang 0 (vị trí từ 0 đến 199) để lập ma trận; do đó, mọi lần tìm nạp lệnh sẽ từ trang 0.

Vì ba trang khung, Làm sao nhiều trang lỗi lầm Chúng tôi được tạo ra qua các tiếp theo khởi tạo mảng vòng lặp, sử dụng LRU thay thế, và trang giả định khung 1 có các quy trình trong nó, và các khác hai Chúng tôi ban đầu trống rỗng?

Một. for (int j = 0; j <100; j ++) for (int i = 0; i <100; i ++) A [i] [j] = Số 0;

b. for (int i = 0; i <100; i ++) for (int j = 0; j <100; j ++) A [i] [j] = Số 0;

#### Trả lời:

* + 1. 50

b. 5.000

* 1. Xem xét tài liệu tham khảo trang sau sợi dây:

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

**32 Chương 9 Ảo Kỉ niệm**

Có bao nhiêu lỗi trang xảy ra đối với các nhịp thay thế sau đây, giả sử là một, hai, ba, bốn, năm, sáu hoặc bảy khung? Hãy nhớ rằng ban đầu tất cả các khung đều trống, vì vậy các trang duy nhất đầu tiên của bạn sẽ có giá một lỗi cho mỗi trang.

* LRU sự thay thế
* FIFO sự thay thế
* Tối ưu sự thay thế

#### Trả lời:

|  |  |  |  |
| --- | --- | --- | --- |
| Số khung s | LRU | FIFO | Tối ưu |
| 1 | 20 | 20 | 20 |
| 2 | 18 | 18 | 15 |
| 3 | 15 | 16 | 11 |
| 4 | 10 | 14 | số 8 |
| 5 | số 8 | 10 | 7 |
| 6 | 7 | 10 | 7 |
| 7 | 7 | 7 | 7 |

* 1. Giả sử rằng bạn muốn sử dụng thuật toán phân trang yêu cầu giới thiệu chút (như là bằng cơ hội thứ hai sự thay thế hoặc bộ làm việc người mẫu), nhưng phần cứng không cung cấp một. Phác thảo cách bạn có thể mô phỏng một bit tham chiếu ngay cả khi một bit không được phần cứng cung cấp hoặc giải thích lý do tại sao không thể làm như vậy. Nếu có thể, hãy tính toán chi phí sẽ là bao nhiêu là.

**Trả lời:** Bạn có thể sử dụng bit hợp lệ / không hợp lệ được hỗ trợ trong phần cứng để mô phỏng các thẩm quyền giải quyết chút. Ban đầu đặt các chút đến không hợp lệ. Trên đầu tiên tham chiếu một cái bẫy đến hệ điều hành được tạo ra. Hệ điều hành sẽ đặt một bit phần mềm thành 1 và đặt lại bit hợp lệ / không hợp lệ thành có hiệu lực.

* 1. Bạn có nghĩ ra Một Mới thay thế trang thuật toán điều đó bạn nghĩ có lẽ tối ưu. Ở trong một số vặn vẹo kiểm tra các trường hợp, Belady's nghĩa bóng xảy ra. Là thuật toán mới tối ưu? Giải thích của bạn trả lời.

**Trả lời: Không. Một thuật toán tối ưu sẽ không mắc phải sự bất thường của** Belady vì — theo định nghĩa — một thuật toán tối ưu sẽ thay thế trang sẽ không được sử dụng trong thời gian dài nhất. Sự bất thường của Belady xảy ra khi thuật toán thay thế trang loại bỏ một trang sẽ cần thiết ngay lập tức. Một thuật toán tối ưu sẽ không chọn một trang.

* 1. Phân đoạn tương tự như phân trang nhưng sử dụng các trang có kích thước thay đổi . ” Loại bỏ hai thuật toán thay thế phân đoạn dựa trên các lược đồ thay thế trang FIFO và LRU . Hãy nhớ rằng vì các phân đoạn không có cùng kích thước, nên phân đoạn được chọn để thay thế có thể không đủ lớn để rời bỏ đầy đủ liên tiếp địa điểm vì các cần thiết bộ phận. Xem xét các chiến lược cho các hệ thống không thể di dời các phân đoạn và các chiến lược cho các hệ thống mà chúng có thể.

#### Trả lời:

**Thực hành Bài tập 33**

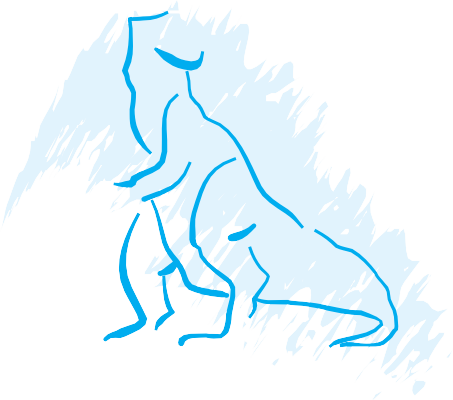
* + 1. **FIFO** . Tìm thấy các đầu tiên bộ phận lớn đầy đủ đến cung cấp các mới đến bộ phận. Nếu tái định cư Là không phải có thể được và không một phân khúc đủ lớn, hãy chọn một tổ hợp các phân khúc có phần ghi nhớ liền nhau, " gần nhất với phần đầu tiên trong danh sách " và có thể phù hợp với phân khúc mới. Nếu có thể di dời, sắp xếp lại các kỉ niệm Vì thế điều đó các đầu tiên *n* phân đoạn đủ lớn vì các mới đến bộ phận Chúng tôi tiếp giáp trong kỉ niệm. Thêm bất kỳ dung lượng còn lại nào vào danh sách dung lượng trống trong cả hai các trường hợp.
    2. **LRU** . Chọn phân đoạn đã không được sử dụng trong thời gian dài nhất của thời gian và điều đó Là lớn đầy đủ, thêm vào bất cứ thức ăn thừa không gian cho danh sách dung lượng trống. Nếu không có một phân đoạn nào đủ lớn, hãy chọn kết hợp các phân đoạn " cũ nhất " nằm liền kề nhau trong bộ nhớ (nếu tái định cư Là không phải có sẵn) và điều đó Chúng tôi lớn đầy đủ. Nếu khả dụng di dời, hãy sắp xếp lại *N* phân đoạn cũ nhất để liền kề trong bộ nhớ và thay thế chúng bằng phân đoạn mới bộ phận.
  1. Hãy xem xét một hệ thống máy tính phân trang theo yêu cầu mà mức độ đa chương trình hiện được cố định ở mức bốn. Hệ thống gần đây đã được kiểm tra để xác định việc sử dụng CPU và đĩa phân trang. Kết quả là một trong những lựa chọn thay thế sau đây. Đối với mỗi trường hợp, điều gì đang xảy ra? Có thể tăng mức độ đa chương trình để tăng hiệu suất sử dụng CPU không? Là phân trang giúp đỡ?
     1. CPU 13 phần trăm; sử dụng đĩa 97 phần trăm
     2. CPU 87 phần trăm; sử dụng đĩa 3 phần trăm
     3. CPU 13 phần trăm; sử dụng đĩa 3 phần trăm

#### Trả lời:

1. Đập là đang xảy ra.
2. CPU đủ cao để mọi thứ yên tĩnh và tăng mức độ đa chương trình.
3. Tăng mức độ của đa chương trình.
   1. Chúng tôi có hệ điều hành cho máy sử dụng thanh ghi cơ sở và giới hạn, nhưng chúng tôi đã sửa đổi máy để cung cấp bảng trang. Các bảng trang có thể được thiết lập để mô phỏng các thanh ghi cơ sở và giới hạn không? Làm thế nào họ có thể được, hoặc tại sao họ có thể không là?

**Trả lời:** Các trang chiếc bàn có thể là đặt hướng lên đến mô phỏng cơ sở và giới hạn thanh ghi với điều kiện là bộ nhớ được cấp phát trong các phân đoạn có kích thước cố định. Bằng cách này , cơ sở của một phân đoạn có thể được nhập vào bảng trang và giá trị hợp lệ / không hợp lệ chút đã sử dụng đến biểu thị điều đó phần của các bộ phận bằng cu tru tai các kỉ niệm. Ở đó sẽ là một số vấn đề với Nội bộ sự phân mảnh.

## Giao diện hệ thống tệp



C*1*H A*0*PTER

### Bài tập thực hành

* 1. Một số hệ thống tự động xóa bỏ tất cả người sử dụng các tập tin khi nào Một người sử dụng nhật ký tắt hoặc một công việc chấm dứt, trừ khi người dùng yêu cầu rõ ràng rằng chúng được giữ lại; các hệ thống khác giữ tất cả các tệp trừ khi người dùng xóa chúng một cách rõ ràng. Thảo luận về giá trị tương đối của mỗi cách tiếp cận.

**Trả lời:** Xóa tất cả các tệp không được người dùng lưu cụ thể có lợi thế là giảm thiểu dung lượng tệp cần thiết cho mỗi người dùng bằng cách không lưu không mong muốn hoặc không cần thiết các tập tin. Tiết kiệm tất cả các tập tin trừ phi xóa cụ thể là an toàn hơn cho người dùng vì không thể vô tình làm mất tệp do quên lưu họ.

* 1. Tại sao làm một số hệ thống giữ theo dõi của các thể loại của Một tập tin, trong khi khác để nó cho người dùng hoặc đơn giản là không triển khai nhiều loại tệp? Hệ thống nào “ tốt hơn? ”

**Trả lời:** Một số hệ thống cho phép các hoạt động tệp khác nhau dựa trên loại của các tập tin (vì ví dụ, một ascii tập tin có thể là đọc bằng Một dòng trong khi một tệp cơ sở dữ liệu có thể được đọc thông qua một chỉ mục cho một khối). Các hệ thống khác để lại như vậy diễn dịch của Một các tập tin dữ liệu đến các quy trình và cung cấp không Cứu giúp trong việc truy cập dữ liệu. Phương pháp " tốt hơn " phụ thuộc vào nhu cầu của các quy trình trên các hệ thống, và các yêu cầu các người dùng địa điểm trên điều hành hệ thống. Nếu Một hệ thống chạy hầu hết cơ sở dữ liệu các ứng dụng, nó có thể hiệu quả hơn để hệ điều hành triển khai một tệp kiểu cơ sở dữ liệu và cung cấp các hoạt động, thay vì làm cho mỗi chương trình triển khai cùng một thứ (có thể theo những cách khác nhau). Đối với các hệ thống có mục đích chung, tốt hơn là chỉ nên triển khai các loại tệp cơ bản để giữ cho kích thước hệ điều hành nhỏ hơn và cho phép tự do tối đa đối với các quy trình trên hệ thống.

**35**

**36 Chương 10 Hệ thống tệp Giao diện**

* 1. Tương tự, một số hệ thống hỗ trợ nhiều kiểu cấu trúc cho dữ liệu của tệp, trong khi những hệ thống khác chỉ hỗ trợ một luồng byte. Những lợi thế là gì và nhược điểm?

**Trả lời:** Một lợi thế của việc hệ thống hỗ trợ các cấu trúc tệp khác nhau là sự hỗ trợ đến từ hệ thống; các ứng dụng cá nhân không bắt buộc phải cung cấp hỗ trợ. Ngoài ra, nếu hệ thống cung cấp hỗ trợ cho các cấu trúc tệp khác nhau, nó có thể thực hiện hỗ trợ có lẽ hiệu quả hơn một ứng dụng.

Các bất lợi của đang có các hệ thống cung cấp hỗ trợ vì xác định loại tập tin Là điều đó nó tăng các kích cỡ của các hệ thống. Ở trong phép cộng, ứng dụng có thể yêu cầu khác nhau tập tin các loại khác hơn Cái gì Là cung cấp qua hệ thống có thể không chạy được trên các hệ thống.

Một thay thế chiến lược Là vì các điều hành hệ thống đến định nghĩa không hỗ trợ cho các cấu trúc tệp và thay vào đó coi tất cả các tệp là một chuỗi các byte. Đây là cách tiếp cận được thực hiện bởi các hệ thống UNIX . Ưu điểm của phương pháp này là điều đó nó đơn giản hóa các điều hành hệ thống hỗ trợ vì tập tin hệ thống, bằng hệ thống không còn phải cung cấp cấu trúc cho các loại tệp khác nhau. Hơn nữa, nó cho phép các ứng dụng đến định nghĩa tập tin cấu trúc, bằng cách ấy giảm nhẹ các tình hình ở đâu Một hệ thống có thể không phải cung cấp Một tập tin định nghĩa cần thiết cho một ứng dụng cụ thể.

* 1. Bạn có thể mô phỏng cấu trúc thư mục đa cấp với một thư mục cấp đơn không kết cấu trong cái mà tùy tiện Dài những cái tên có thể là đã sử dụng? Nếu câu trả lời của bạn Là Vâng, giải thích Làm sao bạn có thể làm Vì thế, và sự tương phản cái này kế hoạch với lược đồ thư mục đa cấp. Nếu câu trả lời của bạn là không, hãy giải thích điều gì ngăn cản sự thành công của mô phỏng. Câu trả lời của bạn sẽ thay đổi như thế nào nếu tên tệp được giới hạn ở bảy nhân vật?

**Trả lời:** Nếu các tên dài tùy ý có thể được sử dụng thì có thể mô phỏng Một đa cấp danh mục kết cấu. Cái này có thể là xong, vì ví dụ, bởi sử dụng các nhân vật “ . ” đến biểu thị các chấm dứt của Một thư mục con. Vì vậy, Ví dụ, các Tên *jim.java.F1* chỉ định điều đó *F1* Là Một tập tin trong thư mục con *java* lần lượt nằm trong thư mục gốc *jim* .

Nếu tên tệp được giới hạn trong bảy ký tự, thì lược đồ trên có thể không phải là sử dụng và do đó, trong chung, các trả lời Là *không* . Các tiếp theo cách tiếp cận tốt nhất trong trường hợp này là sử dụng một tệp cụ thể làm bảng ký hiệu (thư mục) để ánh xạ các tên dài tùy ý (chẳng hạn như *jim.java.F1* ) thành các tên tùy ý ngắn hơn (chẳng hạn như *XX00743* ), sau đó được sử dụng cho tệp thực tế truy cập.

* 1. Giải thích mục đích của open () và close () các hoạt động.

#### Trả lời:

* + - open () thông báo cho hệ thống rằng tệp được đặt tên sắp trở thành tích cực.
    - close () thông báo cho hệ thống rằng tệp được đặt tên không còn được sử dụng tích cực bởi người dùng đã ban hành lệnh đóng hoạt động.
  1. Đưa ra một ví dụ về một ứng dụng trong đó dữ liệu trong một tệp sẽ được truy cập như sau đơn hàng:

1. Tuần tự

**Thực hành Bài tập 37**

1. Ngẫu nhiên

#### Trả lời:

1. In nội dung của tập tin.
2. In nội dung của bản ghi *i* . Bản ghi này có thể được tìm thấy bằng cách sử dụng băm hoặc chỉ mục kỹ thuật.
   1. Trong một số hệ thống, một thư mục con có thể được đọc và ghi bởi người dùng có thẩm quyền, giống như các tệp thông thường có thể là.
3. Mô tả các vấn đề bảo vệ có thể nảy sinh.
4. Đề xuất một kế hoạch để đối phó với từng biện pháp bảo vệ mà bạn đã nêu tên trong một phần Một.

#### Trả lời:

1. Một mảnh của thông tin đã giữ trong Một danh mục lối vào Là tập tin vị trí. Nếu người dùng có thể sửa đổi vị trí này, thì anh ta có thể truy cập các tệp khác đánh bại tính năng bảo vệ truy cập kế hoạch.
2. Không cho phép người dùng ghi trực tiếp vào thư mục con. Thay vào đó, cung cấp các hoạt động hệ thống để thực hiện Vì thế.
   1. Xem xét Một hệ thống điều đó ủng hộ 5000 người dùng. Giả sử điều đó bạn muốn để cho phép 4990 trong số những người dùng này có thể truy cập một tập tin.
3. Bạn sẽ chỉ định sơ đồ bảo vệ này như thế nào trong UNIX ?
4. Bạn có thể đề xuất một sơ đồ bảo vệ khác có thể được sử dụng hiệu quả hơn cho mục đích này so với sơ đồ do UNIX cung cấp không?

#### Trả lời:

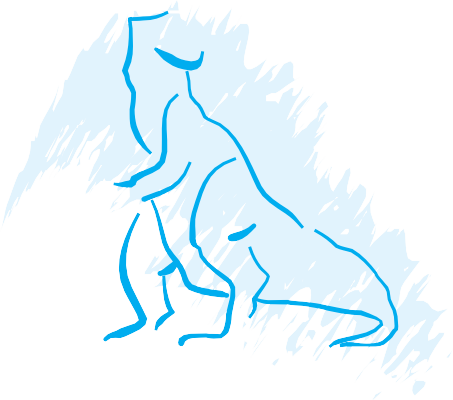
1. Có hai phương pháp để đạt được cái này:
   1. Tạo danh sách kiểm soát truy cập với tên của tất cả 4990 người dùng.
   2. Đặt này 4990 người dùng trong một tập đoàn và đặt các tập đoàn truy cập phù hợp. Không phải lúc nào cũng có thể triển khai lược đồ này vì các nhóm người dùng bị hệ thống hạn chế.
2. Các phổ cập truy cập đến các tập tin áp dụng đến tất cả người dùng trừ phi của chúng tên xuất hiện trong danh sách kiểm soát truy cập với các vị trí truy cập khác nhau. Với lược đồ này, bạn chỉ cần đặt tên của mười người dùng đang cấp lại trong danh sách kiểm soát truy cập nhưng không có đặc quyền truy cập cho phép.
   1. Các nhà nghiên cứu có gợi ý điều đó, thay thế của đang có một truy cập danh sách có liên quan với mỗi tập tin (xác định cái mà người dùng có thể truy cập các tập tin, và làm sao chúng ta Nên có Một *người sử dụng điều khiển danh sách* có liên quan với mỗi người sử dụng (chỉ định tệp người dùng có thể truy cập và cách thức). Thảo luận về giá trị tương đối của hai điều này các kế hoạch.

#### Trả lời:

**38 Chương 10 Hệ thống tệp Giao diện**

* *Tập tin điều khiển danh sách* . Từ các truy cập điều khiển thông tin Là tập trung ở một Độc thân địa điểm, nó Là dễ dàng hơn đến thay đổi truy cập điều khiển thông tin và điều này yêu cầu ít hơn khoảng trống.
* *Danh sách kiểm soát người dùng* . Điều này đòi hỏi ít chi phí hơn khi mở một tập tin.

## Triển khai hệ thống tệp



C H*1*A*1*PTER

### Bài tập thực hành

* 1. Hãy xem xét một tệp hiện bao gồm 100 khối. Giả sử rằng khối điều khiển tệp (và khối chỉ mục, trong trường hợp phân bổ được lập chỉ mục) đã có trong bộ nhớ. Tính toán số lượng hoạt động I / O của đĩa được yêu cầu cho các chiến lược allocaon (cấp đơn) liền kề, được liên kết và được lập chỉ mục, nếu đối với một khối, các điều kiện sau đây được giữ nguyên. Trong phân bổ tiếp giáp trường hợp, giả định điều đó ở đó Là không phòng đến lớn lên trong sự bắt đầu, Nhưng ở đó Là phòng đến lớn lên trong các chấm dứt. Giả định điều đó các thông tin khối được thêm vào được lưu trữ trong kỉ niệm.
     1. Khối được thêm vào sự khởi đầu.
     2. Khối được thêm vào tên đệm.
     3. Khối được thêm vào chấm dứt.
     4. Khối được xóa khỏi sự khởi đầu.
     5. Khối được xóa khỏi tên đệm.
     6. Khối được xóa khỏi chấm dứt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Trả lời:** | Liền kề | Đã liên kết | Đã lập chỉ mục |  |
|  | Một. 201 | 1 | 1 |  |
|  | b. 101 | 52 | 1 |  |
|  | C. 1 | 3 | 1 |  |
|  | d. 198 | 1 | 0 |  |
|  | e. 98 | 52 | 0 |  |
|  | f. 0 | 100 | 0 |  |
|  |  |  |  | **39** |

**40 Chương 11 Hệ thống tệp Thực hiện**

* 1. Sự cố nào có thể xảy ra nếu hệ thống cho phép hệ thống tệp được gắn đồng thời tại nhiều hơn một vị trí?

**Trả lời:** Sẽ có nhiều đường dẫn đến cùng một tệp, điều này có thể gây nhầm lẫn cho người dùng hoặc khuyến khích nhầm lẫn (xóa tệp bằng một đường dẫn sẽ xóa tệp trong tất cả các đường dẫn khác).

* 1. Tại sao cần phải các chút bản đồ vì tập tin sự phân bổ là đã giữ trên khối kho, chứ không phải trong chính kỉ niệm?

**Trả lời:** Trong trường hợp hệ thống gặp sự cố (lỗi bộ nhớ), danh sách dung lượng trống sẽ không phải là mất bằng nó sẽ là nếu các chút bản đồ có đã cất giữ trong bộ nhớ chính .

* 1. Xem xét một hệ thống hỗ trợ các chiến lược tiếp giáp, liên kết và lập chỉ mục sự phân bổ. Gì tiêu chuẩn Nên là đã sử dụng trong quyết định chiến lược nào được sử dụng tốt nhất cho một tập tin?

#### Trả lời:

* **Liền** kề — nếu tệp thường được truy cập tuần tự, nếu tệp tương đối nhỏ.
* **liên kết** —nếu tệp lớn và thường được truy cập một cách tuần tự.
* **Đã lập chỉ mục** —nếu tệp lớn và thường được truy cập ngẫu nhiên.
  1. Một vấn đề với phân bổ liền kề là người dùng phải chuẩn bị trước đủ dung lượng cho mỗi tệp. Nếu tệp phát triển lớn hơn không gian được cấp cho nó, thì phải thực hiện các hành động đặc biệt. Một giải pháp cho vấn đề này là xác định cấu trúc tệp bao gồm một vùng xác định ban đầu (có kích thước xác định). Nếu vùng này được lấp đầy, hệ điều hành sẽ tự động xác định vùng tràn được liên kết với tiếp giáp khu vực. Nếu các tràn ra khu vực Là điền, khác tràn ra khu vực được phân bổ. So sánh việc triển khai tệp này với tệp liên kết và liền kề tiêu chuẩn triển khai.

**Trả lời:** Phương pháp này yêu cầu chi phí cao hơn so với phân bổ chuẩn. Nó yêu cầu ít chi phí hơn so với phân bổ được liên kết tiêu chuẩn.

* 1. Làm cách nào để bộ nhớ đệm giúp cải thiện hiệu suất? Tại sao hệ thống không sử dụng nhiều bộ nhớ đệm hơn hoặc lớn hơn nếu chúng như vậy hữu ích?

**Trả lời:** Bộ nhớ đệm cho phép các thành phần của khác nhau tốc độ đến giao tiếp hiệu quả hơn bằng cách tạm thời lưu trữ dữ liệu từ thiết bị chậm hơn trong một thiết bị nhanh hơn (bộ nhớ đệm). Theo định nghĩa, bộ nhớ đệm đắt hơn hơn các thiết bị họ Chúng tôi bộ nhớ đệm vì, Vì thế tăng các số lượng hoặc kích thước bộ nhớ đệm sẽ làm tăng hệ thống Giá cả.

* 1. Tại sao Là nó thuận lợi vì các người sử dụng vì một điều hành hệ thống đến năng động chỉ định nó là Nội bộ những cái bàn? Gì Chúng tôi các hình phạt đến các hệ điều hành để làm Vì thế?

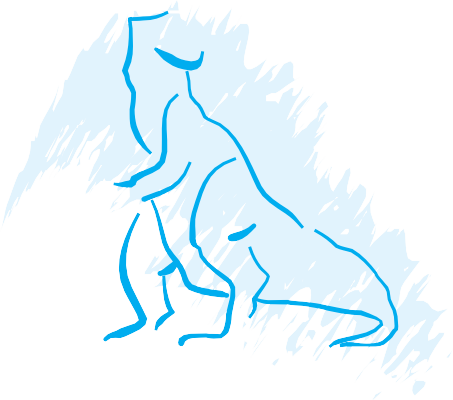
**Trả lời:** Bảng động cho phép linh hoạt hơn trong việc tăng trưởng sử dụng hệ thống - bảng Chúng tôi không bao giờ vượt quá, tránh né nhân tạo sử dụng Hạn mức. Thật không may, cấu trúc hạt nhân và mã phức tạp hơn, vì vậy có nhiều tiềm năng vì lỗi. Các sử dụng của một nguồn có thể lấy xa hơn tài nguyên hệ thống (bằng cách phát triển để đáp ứng các yêu cầu) so với các bảng tĩnh.

**Thực hành Bài tập 41**

* 1. Giải thích Làm sao các VFS lớp cho phép một điều hành hệ thống một cách dễ dàng đến hỗ trợ nhiều loại tệp các hệ thống.

**Trả lời:** VFS giới thiệu một lớp chuyển hướng trong quá trình triển khai hệ thống tệp. Ở trong nhiều cách, nó Là giống đến hướng đối tượng kỹ thuật lập trình. Các cuộc gọi hệ thống có thể được thực hiện một cách chung chung (không phụ thuộc vào loại hệ thống tệp). Mỗi loại hệ thống tệp cung cấp các lệnh gọi hàm của nó và cấu trúc dữ liệu của lớp VFS . Một lệnh gọi hệ thống được dịch thành các chức năng cụ thể thích hợp cho hệ thống tệp đích ở lớp VFS . Chương trình gọi không có mã dành riêng cho hệ thống tệp và các cấp trên của cấu trúc lệnh gọi hệ thống cũng tương tự như vậy là độc lập với hệ thống tệp. Các dịch tại các Lớp VFS lần lượt này chung cuộc gọi vào trong tập tin-hệ thống cụ thể hoạt động.

## Cấu trúc bộ nhớ chung



C*1*H A*2*PTER

### Bài tập thực hành

* 1. Tìm kiếm tăng tốc được mô tả trong Bài tập 12.3 là điển hình của ổ đĩa cứng. Ngược lại, đĩa mềm (và nhiều đĩa cứng được sản xuất trước giữa những năm 1980) thường tìm kiếm ở một tốc độ cố định. Giả sử rằng đĩa trong Bài tập 12.3 có một tìm kiếm tốc độ không đổi chứ không phải một tìm kiếm gia tốc không đổi, vì vậy thời gian tìm kiếm có dạng *tx yL* , trong đó *t* là thời gian tính bằng mili giây và *L* là khoảng cách tìm kiếm. Giả sử rằng thời gian đến tìm kiếm đến một liền kề hình trụ Là 1 mili giây, bằng trước, và Là

= +

0,5 mili giây cho mỗi xi lanh bổ sung.

* + 1. Viết phương trình cho thời gian tìm kiếm này dưới dạng hàm của khoảng cách tìm kiếm.
    2. Sử dụng các tìm kiếm thời gian chức năng từ phần Một, tính toán các toàn bộ tìm kiếm thời gian cho mỗi lịch trình trong Bài tập 12.2. Câu trả lời của bạn có giống với câu trả lời cho Bài tập không 12,3 (c)?
    3. Gì Là các tỷ lệ phần trăm tăng tốc của các nhanh nhất lịch trình kết thúc FCFS trong này trường hợp?

#### Trả lời:

Một. *t* = 0 *.* 95 + 0 *.* 05 *L*

b. FCFS 362.60; SSTF 95,80; QUÉT 497,95; NHÌN 174.50; C-QUÉT 500,15;

(và C-LOOK 176.70). SSTF vẫn là người chiến thắng, và LOOK là á quân.

C. (362. *60* 95. 80 *) /* 362 *.* 60 0 *.* 74 Phần trăm tốc độ tăng của SSTF trên FCFS Là 74%, với kính trọng đến các tìm kiếm thời gian. Nếu chúng tôi bao gồm chi phí của độ trễ quay và truyền dữ liệu, tốc độ phần trăm sẽ là ít hơn.

− =

**43**

**44 Chương 12 Ổ đĩa chung Kết cấu**

* 1. Lập lịch đĩa, khác với lập lịch FCFS , có hữu ích trong môi trường một người dùng không? Giải thích của bạn trả lời.

**Trả lời:** Trong môi trường một người dùng, hàng đợi I / O thường trống. Các yêu cầu thường đến từ một quy trình duy nhất cho một khối hoặc cho một chuỗi các khối liên tiếp. Trong những trường hợp này, FCFS là một phương pháp lập lịch đĩa tiết kiệm. Nhưng LOOK gần như dễ lập trình và sẽ cho hiệu suất tốt hơn nhiều khi nhiều quy trình đang thực hiện đồng thời I / O , như là bằng khi nào một trang web trình duyệt lấy lại dữ liệu trong nền trong khi hệ điều hành đang phân trang và một ứng dụng khác đang hoạt động trong vấn đề xung quanh.

* 1. Giải thích lý do tại sao lập lịch SSTF có xu hướng ưu tiên hình trụ giữa hơn hình trụ trong cùng và ngoài cùng xi lanh.

**Trả lời:** Tâm của đĩa là vị trí có khoảng cách trung bình nhỏ nhất đến tất cả các rãnh còn lại. Do đó đầu đĩa có xu hướng di chuyển ra xa từ các các cạnh của các đĩa. Đây Là khác đường đến nghĩ của nó. Vị trí hiện tại của đầu chia các xi lanh thành hai nhóm. Nếu phần đầu không nằm ở trung tâm của đĩa và một yêu cầu mới đến, thì yêu cầu mới có nhiều khả năng nằm trong nhóm bao gồm trung tâm của đĩa; do đó, đầu có nhiều khả năng di chuyển trong đó phương hướng.

* 1. Tại sao độ trễ quay thường không được xem xét trong lập lịch đĩa? Bạn sẽ sửa đổi SSTF , SCAN và C-SCAN như thế nào để bao gồm tối ưu hóa độ trễ?

**Trả lời:** Hầu hết đĩa làm không phải xuất khẩu của chúng luân phiên Chức vụ thông tin đến máy chủ. Ngay cả khi họ đã làm như vậy, thời gian để thông tin này đến được với bộ lập lịch sẽ không chính xác và thời gian mà bộ lập lịch sử dụng có thể thay đổi, vì vậy thông tin vị trí quay sẽ trở thành không đúng. Thêm nữa, các đĩa yêu cầu Chúng tôi thường xuyên được cho trong thuật ngữ của số khối logic và ánh xạ giữa các khối logic và vị trí vật lý là rất phức tạp.

* 1. Làm sao sẽ sử dụng của Một RAM đĩa có ảnh hưởng đến của bạn sự lựa chọn của Một thuật toán lập lịch đĩa? Bạn cần xem xét những yếu tố nào? Thực hiện các cân nhắc tương tự áp dụng đến ổ đĩa cứng lập lịch trình, được cho điều đó các tập tin hệ thống lưu trữ các khối được sử dụng gần đây trong một bộ đệm đệm trong bộ nhớ chính? **Trả lời:** Lập lịch đĩa cố gắng giảm thời gian định vị đầu đĩa chung. Vì đĩa RAM có thời gian truy cập đồng nhất nên việc lập lịch biểu phần lớn là không cần thiết. Việc so sánh giữa đĩa RAM và bộ nhớ đệm trên đĩa chính không có ý nghĩa gì đối với việc lập lịch đĩa cứng vì chúng tôi chỉ lập lịch cho các lần bỏ sót bộ đệm đệm chứ không phải các yêu cầu tìm thấy dữ liệu của chúng trong bộ nhớ đệm chính. kỉ niệm.
  2. Tại sao điều quan trọng là phải cân bằng I / O hệ thống tệp giữa các đĩa và bộ điều khiển trên một hệ thống trong chế độ đa nhiệm Môi trường?

**Trả lời:** Một hệ thống chỉ có thể thực hiện ở tốc độ cổ chai chậm nhất của nó. Đĩa hoặc đĩa bộ điều khiển Chúng tôi thường xuyên các nút cổ chai trong hệ thống hiện đại vì hiệu suất riêng lẻ của chúng không thể theo kịp với CPU và hệ thống xe buýt. Qua thăng bằng I / O giữa đĩa và bộ điều khiển, không một đĩa riêng lẻ hay một bộ điều khiển nào bị quá tải, vì vậy nút cổ chai đó là tránh được.

**Thực hành Bài tập 45**

* 1. Sự cân bằng liên quan đến việc đọc lại các trang mã từ hệ thống tệp so với việc sử dụng không gian hoán đổi để lưu trữ là gì họ?

**Trả lời:** Nếu mã số trang Chúng tôi cất giữ trong tráo đổi khoảng trống, họ có thể là chuyển nhiều hơn Mau đến chủ yếu kỉ niệm (tại vì tráo đổi khoảng trống sự phân bổ Là điều chỉnh cho nhanh hơn màn biểu diễn hơn chung tập tin hệ thống phân bổ). Sử dụng không gian hoán đổi có thể yêu cầu thời gian khởi động nếu các trang được sao chép ở đó khi gọi quy trình thay vì chỉ được phân trang để hoán đổi không gian theo yêu cầu. Cũng thế, hơn tráo đổi khoảng trống cần phải là phân bổ nếu nó Là đã sử dụng vì cả hai mã số và dữ liệu các trang.

* 1. Có cách nào để triển khai bộ nhớ thực sự ổn định không? Giải thích ý kiến của bạn .

**Trả lời:** Bộ nhớ thực sự ổn định sẽ không bao giờ mất dữ liệu. Kỹ thuật cơ bản để lưu trữ ổn định là duy trì nhiều bản sao của dữ liệu, để nếu một bản sao bị phá hủy, một số bản sao khác vẫn có sẵn để sử dụng. Nhưng đối với bất kỳ kế hoạch nào, chúng ta có thể tưởng tượng một thảm họa đủ lớn mà tất cả các bản sao đều bị phá hủy.

* 1. Thuật ngữ “ SCSI-II tốc độ rộng nhanh ” biểu thị một bus SCSI hoạt động với tốc độ dữ liệu 20 megabyte mỗi giây khi nó di chuyển một gói byte giữa máy chủ và thiết bị. Giả sử rằng một đĩa SCSI-II rộng nhanh lái xe quay tại 7200 RPM , có Một khu vực kích cỡ của 512 byte, và nắm giữ 160 các cung trên mỗi bản nhạc.
     1. Ước tính tốc độ truyền liên tục của ổ đĩa này tính bằng megabyte mỗi thứ hai.
     2. Giả sử điều đó các lái xe có 7000 xi lanh, 20 bài hát mỗi hình trụ, thời gian chuyển đổi đầu (từ đĩa này sang đĩa khác) là 0,5 mili giây và thời gian tìm kiếm hình trụ liền kề là 2 mili giây. Sử dụng thông tin bổ sung này để đưa ra ước tính chính xác về tốc độ truyền được duy trì cho một khoản chuyển lớn.
     3. Giả sử điều đó các Trung bình cộng tìm kiếm thời gian vì các lái xe Là số 8 mili giây. Ước tính I / O s mỗi giây và tốc độ truyền tải hiệu quả cho một truy cập ngẫu nhiên khối lượng công việc điều đó đọc cá nhân các ngành điều đó nằm rải rác trên đĩa.
     4. I / O truy cập ngẫu nhiên s mỗi giây và tốc độ truyền cho các kích thước I / O là 4 kilobyte, 8 kilobyte và 64 kilobyte.
     5. Nếu nhiều yêu cầu Chúng tôi trong các xếp hàng, Một lập kế hoạch thuật toán nhu la QUÉT nên là có thể đến giảm bớt các Trung bình cộng tìm kiếm khoảng cách. Giả sử điều đó Một truy cập ngẫu nhiên khối lượng công việc Là đọc hiểu 8 kilobyte trang, độ dài hàng đợi trung bình là 10 và thuật toán lập lịch giảm thời gian tìm kiếm trung bình xuống còn 3 mili giây. Bây giờ hãy tính I / O s mỗi giây và tốc độ truyền tải hiệu quả của lái xe.

#### Trả lời:

1. Đĩa quay 120 lần mỗi giây và mỗi vòng quay chuyển một bản nhạc của 80 KB \_ Vì vậy, các duy trì chuyển khoản tỷ lệ có thể là xấp xỉ là 9600 KB / s.

**46 Chương 12 Ổ đĩa chung Kết cấu**

1. Giả sử điều đó 100 xi lanh Là Một to lớn chuyển khoản. Các chuyển khoản tỷ lệ là tổng số byte chia cho tổng thời gian. Số byte: 100 cyl \* 20 trk / cyl \* 80 KB / trk, I E, 160.000 KB \_ Thời gian: Vòng xoay thời gian + theo dõi công tắc thời gian + xi lanh công tắc thời gian. Vòng xoay thời gian Là 2000 trks / 120 trks mỗi giây, tức là, 16,667 S. Theo dõi công tắc thời gian Là 19 công tắc mỗi cyl \* 100 cyl \* 0,5 ms, tức là 950 ms. Thời gian chuyển đổi xi lanh là 99 \* 2 ms, tức là 198 ms. Vì vậy, các toàn bộ thời gian Là 16,667 + 0,950 + 0,198, I E, 17.815 S. (Chúng tôi đang bỏ qua mọi tìm kiếm ban đầu và độ trễ xoay vòng , điều này có thể làm tăng thêm xung quanh 12 bệnh đa xơ cứng đến các lịch trình, I E 0,1%.) Như vậy các chuyển khoản tốc độ là 8981,2 KB / s. Chi phí của việc chuyển đổi đường ray và xi lanh là về 6,5%.
2. Các thời gian mỗi chuyển khoản Là số 8 bệnh đa xơ cứng đến tìm kiếm + 4.167 bệnh đa xơ cứng Trung bình cộng độ trễ quay + 0,052 bệnh đa xơ cứng (tính toán từ 1 / (120 trk mỗi thứ hai \* 160 sector trên mỗi trk)) để xoay một sector qua đầu đĩa trong khi đọc. Chúng tôi tính số lần chuyển mỗi giây là 1 / (0,012219), tức là 81,8. Vì mỗi lần truyền là 0,5 KB nên tốc độ truyền là 40,9 KB / s.
3. chúng tôi phớt lờ theo dõi và hình trụ băng qua đường vì sự đơn giản. Vì đọc có kích thước 4 KB , 8 KB , và 64 KB , các tương ứng I / O s mỗi thứ hai được tính toán từ các tìm kiếm, luân phiên độ trễ, và luân phiên chuyển khoản thời gian bằng trong các Trước mục, cho (tương ứng) 1 / (0,0126), 1 / (0,013), và 1 / (0,019). Như vậy chúng tôi lấy 79,4, 76,9, và 52,6 chuyển đổi trên giây, tương ứng. Tỷ lệ chuyển khoản thu được từ 4, 8 và 64 lần các tốc độ I / O này , cho 318 KB / s, 615 KB / s và 3366 KB / s, tương ứng.
4. Từ 1 / (3 + 4,167 + 0,83) chúng tôi được 125 I / O s mỗi thứ hai. Từ số 8

KB mỗi I / O , chúng tôi thu được 1000 KB / s.

* 1. Có thể gắn nhiều ổ đĩa vào một bus SCSI . Đặc biệt, một bus SCSI-II rộng nhanh (xem Bài tập 12.9) có thể được kết nối với nhiều nhất 15 ổ đĩa. Nhớ lại rằng xe buýt này có băng thông 20 megabyte mỗi thứ hai. Tại bất cứ thời gian, chỉ một một gói tin có thể là chuyển nhượng trên các bus giữa một số bộ nhớ cache bên trong của đĩa và máy chủ. Tuy nhiên, một đĩa có thể di chuyển nhánh đĩa của nó trong khi một số đĩa khác đang chuyển một gói trên bus. Ngoài ra, một đĩa có thể truyền dữ liệu giữa các đĩa từ và bộ nhớ đệm bên trong của nó trong khi một số đĩa khác đang chuyển một gói trên bus. Xem xét tốc độ chuyển tiền mà bạn đã tính toán các nhiều khối lượng công việc trong Thể dục 12,9, bàn luận Làm sao nhiều đĩa có thể được sử dụng hiệu quả bởi một SCSI-II rộng nhanh xe buýt.

**Trả lời:** Đối với số 8 KB ngẫu nhiên I / O s trên Một nhẹ nhàng nạp vào đĩa, ở đâu các ngẫu nhiên thời gian truy cập được tính là khoảng 13 ms (xem Bài tập 12.9), tốc độ truyền hiệu dụng là khoảng 615 MB / s. Trong trường hợp này, 15 đĩa sẽ có tổng chuyển khoản tỷ lệ của ít hơn hơn 10 MB / s, cái mà Nên không phải bão hòa xe buýt. Đối với các lần đọc ngẫu nhiên 64 KB vào đĩa được tải nhẹ, tốc độ truyền là khoảng 3,4 MB / s, do đó, năm ổ đĩa trở xuống sẽ bão hòa bus. Đối với các lần đọc 8 KB với hàng đợi đủ lớn để giảm tìm kiếm trung bình xuống 3 ms, tốc độ truyền là khoảng 1 MB / s, do đó băng thông bus có thể đủ để đáp ứng 15 đĩa.

**Thực hành Bài tập 47**

* 1. Việc loại bỏ các khối xấu bằng cách tiết kiệm lĩnh vực hoặc trượt lĩnh vực có thể ảnh hưởng đến hiệu suất. Giả sử rằng ổ đĩa trong bài tập 12.9 có tổng của 100 tồi tệ các ngành tại ngẫu nhiên địa điểm và điều đó mỗi tồi tệ khu vực được ánh xạ tới một phụ tùng nằm trên một đường ray khác, nhưng trong cùng một trụ. Ước tính số lượng I / O mỗi giây và hiệu quả chuyển khoản tỷ lệ vì Một truy cập ngẫu nhiên khối lượng công việc bao gồm của Các lần đọc 8 kilobyte, với độ dài hàng đợi là 1 (nghĩa là lựa chọn thuật toán lập lịch Là không phải Một hệ số). Gì Là các hiệu ứng của Một tồi tệ khu vực trên màn biểu diễn? **Trả lời:** Vì đĩa chứa 22.400.000 cung nên xác suất yêu cầu một trong 100 lĩnh vực được ánh xạ lại là rất nhỏ. Ví dụ về trường hợp xấu nhất là chúng tôi cố đọc một trang 8 KB , nhưng một cung từ giữa bị lỗi và đã được ánh xạ lại đến vị trí xấu nhất có thể trên một rãnh khác trong hình trụ đó. Trong trường hợp này, thời gian vì các thu hồi có thể là số 8 bệnh đa xơ cứng đến tìm kiếm, thêm hai theo dõi công tắc và hai độ trễ xoay đầy đủ. Có khả năng là một bộ điều khiển hiện đại sẽ đọc tất cả các yêu cầu tốt các ngành từ các nguyên bản theo dõi trước khi chuyển sang rãnh dự phòng để truy xuất khu vực được ánh xạ lại và do đó sẽ chỉ phải chịu một chuyển đổi rãnh và độ trễ quay. Vì vậy, thời gian sẽ là 8 ms tìm kiếm + độ trễ quay trung bình 4,17 ms + theo dõi 0,05 ms công tắc + 8,3 bệnh đa xơ cứng luân phiên độ trễ + 0,83 bệnh đa xơ cứng đọc thời gian (số 8 KB là 16 cung, 1/10 vòng quay theo dõi). Do đó, thời gian để phục vụ yêu cầu này sẽ là 21,8 mili giây, cho tốc độ I / O là 45,9 yêu cầu mỗi giây và băng thông hiệu dụng là 367 KB / s. Đối với một ứng dụng bị giới hạn thời gian nghiêm trọng, điều này có thể thành vấn đề, nhưng tác động tổng thể trong mức trung bình có trọng số của 100 ánh xạ lại các ngành và 22.4 triệu tốt các ngành Là không.
  2. Trong máy hát tự động đĩa, tác động của việc có nhiều tệp đang mở hơn số ổ đĩa trong máy hát tự động?

**Trả lời:** Hai kết quả xấu có thể dẫn đến. Một khả năng là chết đói các các ứng dụng điều đó phát hành chặn lại I / O s đến băng điều đó Chúng tôi không phải gắn vào ổ đĩa. Khác khả năng Là đập mạnh, bằng các máy hát tự động Là lệnh chuyển đổi băng sau mỗi I / O hoạt động.

* 1. Nếu từ tính siêng năng đĩa sau cùng có các tương tự Giá cả mỗi gigabyte bằng làm băng, băng sẽ trở nên lỗi thời, hay chúng vẫn còn cần thiết? Giải thích của bạn trả lời.

**Trả lời:** Băng có thể tháo rời dễ dàng, vì vậy chúng rất hữu ích cho việc sao lưu bên ngoài trang web và để truyền dữ liệu hàng loạt (bằng cách gửi hộp mực). Theo quy luật, đĩa cứng từ tính không phải là phương tiện di động.

* 1. Đôi khi người ta nói rằng băng là một phương tiện truy cập tuần tự, trong khi đĩa từ là một phương tiện truy cập ngẫu nhiên. Trên thực tế, mức độ phù hợp của thiết bị lưu trữ đối với truy cập ngẫu nhiên phụ thuộc vào kích thước truyền. Thuật ngữ *tốc độ truyền trực tuyến* biểu thị tốc độ dữ liệu cho quá trình truyền đang được thực hiện, loại trừ ảnh hưởng của độ trễ truy cập. Ngược lại, *hiệu quả chuyển khoản tỷ lệ* Là các tỉ lệ của toàn bộ byte mỗi toàn bộ giây, bao gồm cả thời gian trên không, chẳng hạn như truy cập độ trễ.

Giả sử điều đó, trong Một máy tính, các cấp độ 2 bộ nhớ đệm có một truy cập độ trễ 8 nano giây và tốc độ truyền trực tuyến 800 megabyte mỗi giây, các chủ yếu kỉ niệm có một truy cập độ trễ của 60 nano giây và một phát trực tuyến chuyển khoản tỷ lệ của 80 megabyte mỗi thứ hai, các từ tính đĩa

**48 Chương 12 Ổ đĩa chung Kết cấu**

có một truy cập độ trễ của 15 mili giây và Một phát trực tuyến chuyển khoản tỷ lệ là 5 megabyte mỗi giây và ổ băng có độ trễ truy cập là 60 giây và tốc độ truyền trực tuyến là 2 megabyte mỗi giây.

1. Truy cập ngẫu nhiên khiến tốc độ truyền tải hiệu quả của một thiết bị giảm xuống, tại vì không dữ liệu Chúng tôi chuyển nhượng suốt trong các truy cập thời gian. Đối với đĩa được mô tả, tốc độ truyền hiệu quả là bao nhiêu nếu một truy cập trung bình được theo sau bởi truyền trực tuyến 512 byte, 8 kilobyte, 1 megabyte và 16 megabyte?
2. Việc sử dụng một thiết bị là tỷ số giữa tốc độ truyền tải hiệu quả với tốc độ truyền trực tuyến. Tính toán việc sử dụng ổ đĩa để truy cập ngẫu nhiên thực hiện chuyển theo từng kích thước trong số bốn kích thước đã cho trong một phần Một.
3. Giả sử điều đó Một sử dụng của 25 phần trăm (hoặc cao hơn) Là được coi là có thể chấp nhận được. Sử dụng các số liệu hiệu suất đã cho, tính toán kích thước truyền tải nhỏ nhất cho đĩa có thể chấp nhận được sự sử dụng.
4. Hoàn thành câu sau: Đĩa là phân quyền truy cập ngẫu nhiên cho các lần truyền lớn hơn byte, và là một thiết bị truy cập tuần tự cho nhỏ hơn chuyển nhượng.
5. Tính toán các kích thước truyền tải tối thiểu mang lại khả năng tối ưu hóa có thể chấp nhận được cho bộ nhớ đệm, bộ nhớ và băng keo.
6. Khi nào Là Một băng Một truy cập ngẫu nhiên thiết bị, và khi nào Là nó Một truy cập tuần tự thiết bị?

#### Trả lời:

1. Đối với 512 byte, tốc độ truyền tải hiệu quả được tính là theo sau.

ETR = quy mô chuyển nhượng / thời gian chuyển nhượng.

Nếu X là kích thước truyền, thì thời gian truyền là (( X / STR ) + độ trễ). Thời gian truyền là 15ms + ( 512B / 5MB mỗi giây) = 15.0097ms . Có hiệu lực chuyển khoản tỷ lệ Là vì thế 512B / 15,0097 ms = 33.12 KB / giây. ETR cho 8KB = .47MB / giây.

ETR cho 1MB = 4,65MB / giây. ETR cho 16MB = 4,98MB / giây.

1. Sử dụng của các thiết bị vì 512B = 33.12 KB / giây / 5MB / giây =

.0064 = .64

Vì 8KB = 9,4%.

Đối với 1MB = 93%.

Đối với 16MB = 99,6%.

1. Tính .25 = ETR / STR , giải quyết kích thước truyền X.

STR = 5MB , vì vậy 1,25MB / S = ETR . 1.25MB / S \* ((X / 5) + .015) = X.

.25X + .01875 = X. X = .025 MB.

1. MỘT đĩa Là Một truy cập ngẫu nhiên thiết bị vì chuyển tiền lớn hơn hơn K byte (ở đâu K *>* đĩa khối kích cỡ), và Là Một truy cập tuần tự thiết bị đối với chuyển khoản nhỏ hơn.

**Thực hành Bài tập 49**

1. Tính toán kích thước truyền tối thiểu để sử dụng bộ nhớ đệm có thể chấp nhận được kỉ niệm:

STR = 800 MB , ETR = 200, độ trễ = số 8 \* 10 - 9 . 200 ( XMB / 800 + số 8 X 10 - 9 ) = XMB .

.25XMB + 1600 \* 10 - 9 = XMB .

X = 2,24 byte.

Tính toán cho bộ nhớ:

STR = 80 MB , ETR = 20, L = 60 \* 10 - 9 . 20 ( XMB / 80 + 60 \* 10 - 9 ) = XMB .

.25XMB + 1200 \* 10 - 9 = XMB .

X = 1,68 byte.

Tính toán cho băng:

STR = 2MB , ETR = 0,5, L = 60 giây.

.5 ( XMB / 2 + 60) = XMB .

.25XMB + 30 = XMB . X = 40 MB .

1. Nó phụ thuộc vào cách nó được sử dụng. Giả sử chúng ta đang sử dụng băng để khôi phục bản sao lưu. Trong trường hợp này, băng đóng vai trò như một thiết bị truy cập tuần tự, nơi chúng ta đọc tuần tự nội dung của băng. Một ví dụ khác, giả sử chúng ta đang sử dụng băng để truy cập nhiều bản ghi được lưu trữ trên băng. Trong trường hợp này, quyền truy cập vào băng là tùy ý và do đó được coi là ngẫu nhiên.
   1. Giả sử rằng chúng ta đồng ý rằng 1 kilobyte là 1,024 byte, 1 megabyte là 1,024 2 byte, và 1 gigabyte Là 1,024 3 byte. Cái này sự tiến triển tiếp tục thông qua terabyte, petabyte, và exabyte (1,024 6 ). Ở đó Chúng tôi hiện tại một số Mới đề xuất thuộc về khoa học dự án điều đó kế hoạch đến ghi và cửa hàng một vài exabyte dữ liệu trong thập kỷ tới. Để trả lời những câu hỏi sau, bạn sẽ cần phải đưa ra một vài giả định hợp lý; nêu các giả định rằng bạn làm.
      1. Cần bao nhiêu ổ đĩa để chứa 4 exabyte dữ liệu?
      2. Làm sao nhiều từ tính băng sẽ là yêu cầu đến tổ chức 4 exabyte trong số dữ liệu?
      3. Cần bao nhiêu băng quang để chứa 4 exabyte dữ liệu (xem Bài tập 12,21)?
      4. Cần bao nhiêu hộp mực lưu trữ ba chiều để chứa 4 exabyte dữ liệu (xem Bài tập 12,20)?
      5. Mỗi tùy chọn sẽ chiếm bao nhiêu feet khối dung lượng lưu trữ?

#### Trả lời:

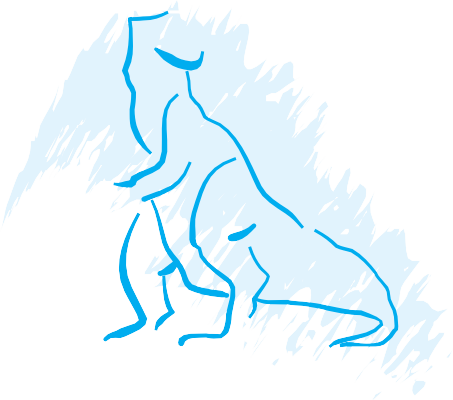
1. Giả sử rằng ổ đĩa chứa 10 GB . Sau đó 100 đĩa chứa 1 TB , 100.000 đĩa tổ chức 1 PB , và 100.000.000 đĩa tổ chức 1 EB . Để lưu trữ 4 EB sẽ cần khoảng 400 triệu đĩa. Nếu một băng từ nắm giữ 40 GB , chỉ một 100 triệu băng sẽ là yêu cầu. Nếu

**50 Chương 12 Ổ đĩa chung Kết cấu**

một quang học băng nắm giữ 50 lần hơn dữ liệu hơn Một từ tính băng, 2 triệu quang học băng sẽ đủ. Nếu Một hình ba chiều hộp đạn có thể lưu trữ 180 GB , xung quanh 22,2 triệu hộp mực sẽ là yêu cầu.

1. Ổ đĩa 3,5 "cao khoảng 1", rộng 4 "và sâu 6". Tính theo feet, giá trị này là 1/12 x 1/3 x 1/2 hoặc 1/72 feet khối. Được đóng gói dày đặc, 400 triệu đĩa sẽ chiếm 5,6 triệu feet khối. Nếu chúng ta cho phép hệ số hai đối với không gian không khí và không gian cho nguồn cung cấp điện, thì công suất cần thiết là khoảng 11 triệu khối đôi chân.
2. Hộp băng 1/2 "có kích thước khoảng 1" cao và 4,5 "hình vuông. Thể tích là khoảng 1/85 feet khối. Đối với 100 triệu cuộn băng từ được đóng gói dày đặc, thể tích khoảng 1,2 triệu feet khối. Đối với quang học 2 triệu băng, khối lượng là 23.400 khối đôi chân.
3. Một đĩa CD-ROM có đường kính 4,75 "và dày khoảng 1/16". Nếu giả sử rằng một đĩa holostore được lưu trữ trong một khe thư viện rộng 5 "vuông và rộng 1/8", chúng tôi tính thể tích của 22,2 triệu đĩa là khoảng 40.000 khối. đôi chân.

# Hệ thống I / O



C1H A3PTER

### Bài tập thực hành

* 1. Nêu ba ưu điểm của việc đặt chức năng trong bộ điều khiển thiết bị, thay vì trong hạt nhân. Tiểu bang ba những bất lợi.

**Trả lời:** Ba ưu điểm: Lỗi ít có khả năng gây ra sự cố hệ điều hành hơn

Hiệu suất có thể được cải thiện bằng cách sử dụng phần cứng chuyên dụng và các thuật toán được mã hóa cứng

Kernel được đơn giản hóa bằng cách chuyển các thuật toán ra khỏi nó

Ba nhược điểm: Lỗi khó sửa hơn — cần có phiên bản chương trình cơ sở mới hoặc phần cứng mới

Việc cải thiện các thuật toán cũng yêu cầu cập nhật phần cứng thay vì chỉ cập nhật nhân hoặc trình điều khiển thiết bị

Các thuật toán nhúng có thể xung đột với việc sử dụng chức năng của ứng dụng, gây giảm hiệu suất.

* 1. Các ví dụ của bắt tay trong Tiết diện 13,2 đã sử dụng 2 chút ít: Một b usy chút và một c ommand đã sẵn sàng chút. Là nó có thể được đến thực hiện cái này bắt tay chỉ với 1 bit? Nếu có, hãy mô tả giao thức. Nếu không, hãy giải thích tại sao 1 bit không đủ.

**Trả lời:** Có thể, bằng cách sử dụng thuật toán sau. Hãy giả sử chúng ta chỉ sử dụng bit bận (hoặc bit sẵn sàng cho lệnh; câu trả lời này giống nhau bất kể). Khi bit tắt, bộ điều khiển không hoạt động. Máy chủ ghi vào dữ liệu ra và đặt bit để báo hiệu rằng một hoạt động đã sẵn sàng (tương đương với việc đặt bit sẵn sàng ra lệnh). Khi bộ điều khiển kết thúc, nó sẽ xóa bit bận. Máy chủ sau đó bắt đầu tiếp theo hoạt động.

Giải pháp này yêu cầu cả máy chủ và bộ điều khiển đều có quyền truy cập đọc và ghi vào cùng một bit, điều này có thể làm phức tạp mạch và tăng chi phí của bộ điều khiển.

**51**

**52 Chương 13 I / O Hệ thống**

* 1. I / O hướng ngắt để quản lý một cổng nối tiếp duy nhất, Nhưng thăm dò ý kiến I / O đến quản lý Một đầu cuối bộ xử lý, như là bằng Một thiết bị đầu cuối tập trung?

**Trả lời:** Thăm dò ý kiến có thể là hơn Có hiệu quả hơn Gián đoạn hướng I / O. Đây là các trường hợp khi nào các I / O Là thường xuyên và của ngắn khoảng thời gian. Thậm chí mặc dù một cổng nối tiếp duy nhất sẽ thực hiện I / O tương đối không thường xuyên và do đó sử dụng ngắt quãng, Một thu thập của nối tiếp cổng như là bằng những, cái đó trong Một thiết bị tập trung đầu cuối có thể sản xuất Một nhiều của ngắn I / O hoạt động, và làm gián đoạn cho từng cái có thể tạo ra một tải nặng cho hệ thống. Vòng lặp thăm dò đúng thời gian có thể giảm bớt tải mà không lãng phí nhiều tài nguyên thông qua vòng lặp không có I / O cần thiết.

* 1. Việc thăm dò ý kiến để hoàn thành I / O có thể lãng phí một số lượng lớn chu kỳ CPU nếu các bộ xử lý lặp lại Một bận-đợi vòng nhiều lần trước các I / O hoàn tất. Nhưng mà nếu các I / O thiết bị Là Sẵn sàng vì dịch vụ, thăm dò ý kiến có thể là hiệu quả hơn nhiều so với việc bắt và điều phối một ngắt. Mô tả một chiến lược kết hợp kết hợp bỏ phiếu, ngủ và ngắt cho I / O thiết bị dịch vụ. Vì mỗi của này ba chiến lược (nguyên chất thăm dò ý kiến, ngắt thuần túy, kết hợp), mô tả một môi trường máy tính trong đó chiến lược đó hiệu quả hơn một trong hai khác.

**Trả lời:** Một phương pháp kết hợp có thể chuyển đổi giữa bỏ phiếu và liên kết tùy thuộc vào thời gian chờ thao tác I / O. Ví dụ: chúng ta có thể thăm dò và lặp lại N lần, và nếu thiết bị vẫn bận ở N + 1, chúng ta có thể đặt ngắt và ngủ. Cách tiếp cận này sẽ tránh các chu kỳ chờ đợi lâu. Phương pháp này sẽ tốt nhất cho thời gian bận rộn rất dài hoặc rất ngắn. Sẽ không hiệu quả nếu I / O hoàn thành ở N + T (trong đó T là một số chu kỳ nhỏ) do chi phí bỏ phiếu cộng với thiết lập và bắt ngắt.

Nguyên chất thăm dò ý kiến Là tốt nhất với hết sức ngắn Chờ đã lần. Ngắt Chúng tôi tốt nhất với sự chờ đợi lâu lần.

* 1. DMA làm tăng tính đồng thời của hệ thống như thế nào? Nó làm phức tạp phần cứng như thế nào thiết kế?

**Trả lời:** DMA làm tăng tính đồng thời của hệ thống bằng cách cho phép CPU thực hiện các tác vụ trong khi hệ thống DMA truyền dữ liệu qua các bus hệ thống và bộ nhớ. Thiết kế phần cứng rất phức tạp vì bộ điều khiển DMA phải được tích hợp vào hệ thống và hệ thống phải cho phép bộ điều khiển DMA làm bộ điều khiển bus . Đánh cắp chu kỳ cũng có thể cần thiết để cho phép CPU và bộ điều khiển DMA chia sẻ quyền sử dụng bộ nhớ xe buýt.

* 1. Tại sao Là nó quan trọng đến tỉ lệ hướng lên hệ thống xe buýt và thiết bị tốc độ bằng các

CPU tăng lên?

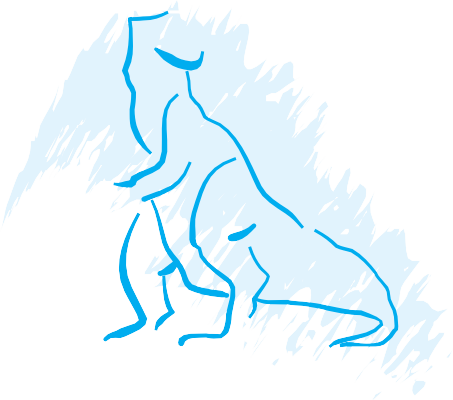
**Trả lời:** Hãy xem xét một hệ thống thực hiện 50% I / O và 50%. Tăng gấp đôi hiệu suất CPU trên hệ thống này sẽ chỉ tăng tổng hiệu suất hệ thống lên 50%. Nhân đôi cả hai khía cạnh hệ thống sẽ tăng hiệu suất lên 100%. Nói chung, điều quan trọng là phải giải quyết nút thắt cổ chai của hệ thống hiện tại và tăng hiệu suất tổng thể của hệ thống, thay vì tăng hiệu suất của các thành phần hệ thống riêng lẻ một cách mù quáng.

* 1. Phân biệt giữa trình điều khiển STREAMS và STREAMS mô-đun.

**Thực hành Bài tập 53**

**Trả lời:** Trình điều khiển STREAMS điều khiển một thiết bị vật lý có thể tham gia vào hoạt động STREAMS . Mô -đun STREAMS sửa đổi luồng dữ liệu giữa phần đầu (giao diện người dùng) và trình điều khiển.

*Sự bảo vệ*



C*1*H A*4*PTER

### Thực hành Bài tập

* 1. Gì Chúng tôi các chủ yếu sự khác biệt giữa khả năng danh sách và truy cập danh sách? **Trả lời:** Một truy cập danh sách Là Một danh sách vì mỗi vật bao gồm của các tên miền có tập hợp quyền truy cập không có gì cho đối tượng đó. Danh sách khả năng là danh sách các đối tượng và các hoạt động được phép trên các đối tượng đó cho mỗi miền.
  2. MCP Burroughs B7000 / B6000 có thể được gắn thẻ là dữ liệu nhạy cảm. Khi một tệp như vậy bị xóa, vùng lưu trữ của nó sẽ bị ghi đè bởi một số bit ngẫu nhiên. Vì mục đích gì mà một kế hoạch như vậy sẽ hữu ích? **Trả lời:** Cái này sẽ là hữu ích bằng một thêm Bảo vệ đo lường Vì thế điều đó nội dung cũ của bộ nhớ không thể được truy cập, dù cố ý hay tình cờ, bởi một chương trình khác. Điều này đặc biệt hữu ích cho bất kỳ thông tin được phân loại cao.
  3. Trong hệ thống bảo vệ vòng, cấp 0 có quyền truy cập lớn nhất vào các đối tượng và mức độ *n* (lớn hơn hơn số không) có ít hơn truy cập các quyền. Các truy cập quyền của Một chương trình tại Một cụ thể mức độ trong các nhẫn kết cấu Chúng tôi xem xét như một tập hợp các khả năng. Mối quan hệ giữa các khả năng của miền cấp *j* và miền cấp *i* với một đối tượng (đối với *j > i* ) là gì? **Đáp án:** *D j* là một tập hợp con của *D i* .
  4. Các RC 4000 hệ thống (và khác hệ thống) có xác định Một cây của quy trình (được gọi là Một quy trình cây) như là điều đó tất cả các hậu duệ của Một quy trình Chúng tôi các tài nguyên (đối tượng) đã cho và chỉ có quyền truy cập bởi tổ tiên của chúng . Vì vậy, một hậu duệ không bao giờ có thể có khả năng làm bất cứ điều gì mà tổ tiên không làm được. Gốc của cây là hệ điều hành, có khả năng làm bất cứ điều gì. Giả sử tập hợp các quyền truy cập được biểu diễn bằng một ma trận truy cập , *A. A* ( *x* , *y* ) xác định quyền truy cập của quá trình *x* đến

**55**

**56 chương 14 Sự bảo vệ**

vật *y* . Nếu *x* Là Một con cháu của *z* , Cái gì Là các mối quan hệ giữa *A* ( *x* , *y* ) và *A* ( *z* , *y* ) cho một đối tượng tùy ý *y* ?

**Trả lời:** *A* ( *x, y* ) là tập con của *A* ( *z, y* ).

* 1. Những vấn đề bảo vệ nào có thể phát sinh nếu một ngăn xếp dùng chung được sử dụng cho máy đo đi qua?

**Trả lời:** Nội dung của ngăn xếp có thể bị xâm phạm bởi (các) quy trình khác chia sẻ ngăn xếp.

* 1. Hãy xem xét một môi trường máy tính nơi một số duy nhất được kết hợp với từng quy trình và từng đối tượng trong hệ thống. Giả sử rằng chúng tôi cho phép Một quy trình với con số *n* đến truy cập một vật với con số *m* chỉ khi *n > m* . Loại cấu trúc bảo vệ nào chúng ta có?

**Trả lời:** Cấu trúc thứ bậc.

* 1. Hãy xem xét một môi trường máy tính trong đó một quá trình được trao quyền ưu tiên chỉ truy cập một đối tượng *n* lần. Đề xuất một kế hoạch để thực hiện điều này chính sách.

**Trả lời:** Thêm một bộ đếm số nguyên với khả năng.

* 1. Nếu tất cả các truy cập quyền lợi đến một vật Chúng tôi đã xóa, các vật có thể không còn được truy cập. Tại thời điểm này, đối tượng cũng sẽ bị xóa và khoảng trắng nó nơi làm việc Nên là trả lại đến các hệ thống. Gợi ý một thực hiện hiệu quả điều này kế hoạch.

**Trả lời:** Số lượng tham chiếu.

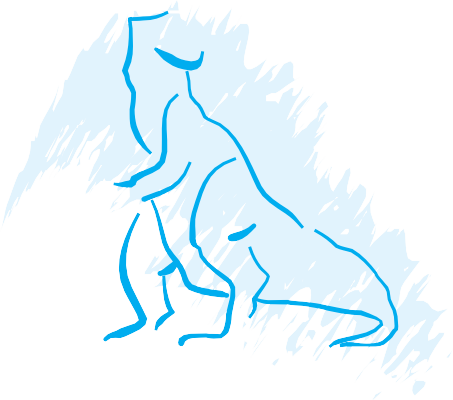
* 1. Tại sao lại khó bảo vệ một hệ thống trong đó người dùng được phép làm việc riêng của họ I / O ?

**Trả lời:** Ở trong sớm hơn chương chúng tôi xác định Một sự phân biệt giữa kernel và chế độ người dùng trong đó chế độ kernel được sử dụng để thực hiện các hoạt động đặc quyền như I / O. Một lý do tại sao I / O phải được thực hiện trong chế độ hạt nhân là I / O yêu cầu truy cập phần cứng và quyền truy cập thích hợp vào phần cứng là cần thiết cho tính toàn vẹn của hệ thống. Nếu chúng tôi cho phép người dùng thực hiện I / O của riêng họ , chúng tôi không thể đảm bảo hệ thống thanh Liêm.

* 1. Danh sách khả năng thường được giữ trong không gian địa chỉ của người dùng. Làm thế nào để hệ thống đảm bảo rằng người dùng không thể sửa đổi nội dung của danh sách?

**Trả lời:** Danh sách khả năng được coi là “ đối tượng được bảo vệ ” và chỉ được truy cập gián tiếp bởi người dùng. Hệ điều hành đảm bảo người dùng không thể truy cập trực tiếp vào danh sách khả năng.

# Cấu trúc hệ thống phân tán



C1H A6PTER

### Bài tập thực hành

* 1. Hầu hết các mạng WAN chỉ sử dụng cấu trúc liên kết được kết nối một phần . Tại sao cái này rất?

**Trả lời:** Chi phí. Một mạng được kết nối đầy đủ yêu cầu liên kết giữa mọi nút trong mạng. Đối với mạng WAN , điều này có thể quá tốn kém vì các liên kết thông tin giữa các máy chủ ở xa về mặt vật lý có thể tốn kém.

* 1. Trong những trường hợp nào thì mạng truyền mã thông báo hiệu quả hơn Ethernet mạng?

**Trả lời:** Vòng mã thông báo rất hiệu quả trong điều kiện tải duy trì cao, vì không có xung đột nào có thể xảy ra và mỗi vị trí có thể được sử dụng để mang một thông điệp, cung cấp thông lượng cao. Vòng mã thông báo kém hiệu quả hơn khi tải Là ánh sáng (mã thông báo Chế biến nhận lâu hơn hơn xe buýt truy cập, Vì thế bất cứ một gói có thể mất nhiều thời gian hơn để đến đích) hoặc rời rạc.

* 1. Tại sao việc các cổng truyền các gói tin quảng bá giữa các mạng lại là một ý tưởng tồi? Lợi ích của việc làm như vậy là gì? **Trả lời:** Tất cả các chương trình phát sóng sẽ được truyền tới tất cả các mạng, gây ra *rất nhiều* lưu lượng mạng. Nếu lưu lượng truyền phát bị giới hạn ở dữ liệu quan trọng (và rất ít trong số đó), thì quá trình truyền phát sẽ tiết kiệm các cổng khỏi phải chạy phần mềm đặc biệt để theo dõi dữ liệu này (chẳng hạn như thông tin định tuyến mạng) và phát lại nó.
  2. Thảo luận về ưu điểm và nhược điểm của việc chuyển đổi tên bộ đệm cho các máy tính đặt ở xa các miền.

**Trả lời:** Có một lợi thế về hiệu suất đối với tên bộ nhớ đệm transla- tions cho máy tính đặt trong miền từ xa: phân giải lặp lại có cùng tên từ các máy tính khác nhau nằm trong miền cục bộ có thể được thực hiện cục bộ mà không yêu cầu thao tác tra cứu tên từ xa. Các bất lợi Là điều đó ở đó có thể là mâu thuẫn trong các

**59**

**60 Chương 16 Hệ thống phân tán cấu trúc**

Tên bản dịch khi nào cập nhật Chúng tôi làm ra trong các lập bản đồ của những cái tên tới IP các địa chỉ. Này Tính nhất quán các vấn đề có thể là đã giải quyết qua làm mất hiệu lực bản dịch, yêu cầu trạng thái được quản lý liên quan đến máy tính nào đang lưu vào bộ nhớ đệm một bản dịch nhất định và cũng sẽ tạo ra một số thông báo vô hiệu hoặc bằng cách sử dụng các hợp đồng thuê bộ nhớ đệm thực thể làm mất hiệu lực Một dịch sau Một chắc chắn giai đoạn = Stage của thời gian. Các sau này cách tiếp cận đòi hỏi ít hơn tình trạng và không sự vô hiệu tin nhắn nhưng có thể bị mâu thuẫn tạm thời.

* 1. Gì Chúng tôi các thuận lợi và nhược điểm của sử dụng mạch điện chuyển đổi? Đối với những loại ứng dụng nào, chuyển mạch kênh là một chiến lược khả thi? **Trả lời:** Chuyển mạch kênh đảm bảo rằng tài nguyên mạng yêu cầu chuyển giao được bảo lưu trước khi quá trình truyền diễn ra. Cái này đảm bảo điều đó gói tin sẽ không phải là giảm và của chúng chuyển sẽ đáp ứng các yêu cầu về chất lượng dịch vụ. Nhược điểm của chuyển mạch kênh là nó yêu cầu một bản tin khứ hồi để thiết lập các dự trữ và nó cũng có thể cung cấp quá mức tài nguyên, do đó dẫn đến việc sử dụng tài nguyên dưới mức tối ưu. Chuyển mạch kênh là một chiến lược khả thi đối với các ứng dụng có nhu cầu liên tục về tài nguyên mạng và sẽ yêu cầu tài nguyên trong thời gian dài, do đó khấu hao ban đầu chi phí chung.
  2. Gì Chúng tôi hai ghê gớm các vấn đề điều đó nhà thiết kế cần phải giải quyết đến áp dụng một hệ thống mạng trong suốt?

**Trả lời:** Một vấn đề như vậy là làm cho tất cả các bộ xử lý và thiết bị lưu trữ có vẻ trong suốt trên toàn mạng. Nói cách khác, hệ thống phân bổ phải xuất hiện như một hệ thống tập trung cho người dùng. Hệ thống tệp Andrew và NFS cung cấp tính năng này: hệ thống tệp phân tán xuất hiện với người dùng dưới dạng một hệ thống tệp duy nhất nhưng trên thực tế, nó có thể được phân phối trên một mạng.

Một vấn đề khác liên quan đến tính di động của người dùng. Chúng tôi muốn cho phép người dùng kết nối với “ hệ thống ” chứ không phải với một máy cụ thể (mặc dù trên thực tế, họ có thể đang đăng nhập vào một máy cụ thể ở đâu đó trong hệ thống phân tán).

* 1. Quá trình di chuyển trong một mạng không đồng nhất thường là không thể, do sự khác biệt về kiến trúc và hệ điều hành. Ghi lại một phương pháp để di chuyển quy trình trên các kiến trúc khác nhau đang chạy:
     1. Cùng một hoạt động hệ thống
     2. Hệ điều hành khác nhau

**Trả lời:** Đối với cùng một hệ điều hành, quá trình di chuyển là tương đối thẳng thắn, bằng các tình trạng của các quy trình nhu cầu đến di cư từ một bộ xử lý đến khác. Cái này liên quan đến động các Địa chỉ khoảng trống, trạng thái của các thanh ghi CPU , và mở các tệp từ hệ thống nguồn đến mô tả. Tuy nhiên, nó Là quan trọng điều đó giống hệt nhau bản sao của các hệ điều hành đang chạy trên các hệ thống khác nhau để đảm bảo tính tương thích. Nếu hệ điều hành giống nhau, nhưng có lẽ các phiên bản khác nhau đang chạy trên các tách rời hệ thống, sau đó di cư quy trình cần phải là

**Thực hành Bài tập 61**

Chắc chắn đến theo lập trình hướng dẫn điều đó Chúng tôi đặc giữa các phiên bản khác nhau của hoạt động hệ thống.

Các ứng dụng Java cung cấp một ví dụ hay về quá trình di chuyển giữa các ứng dụng khác nhau điều hành các hệ thống. Đến giấu sự khác biệt trong các cơ bản hệ thống, quá trình được di chuyển (tức là một ứng dụng Java) chạy trên một máy ảo chứ không phải một hệ điều hành cụ thể. Tất cả những gì được yêu cầu là dành cho ảo máy móc đến là đang chạy trên các hệ thống các quy trình di cư đến.

* 1. Để xây dựng một hệ thống phân tán mạnh mẽ, bạn phải biết những loại lỗi nào có thể xảy ra xảy ra.
     1. Liệt kê ba loại lỗi có thể xảy ra trong một phân phối hệ thống.
     2. Chỉ định mục nào trong danh sách của bạn cũng có thể áp dụng cho hệ thống tập trung.

**Trả lời:** Ba lỗi phổ biến trong hệ thống phân tán bao gồm:

(1) lỗi liên kết mạng, (2) lỗi máy chủ, (3) lỗi phương tiện lưu trữ. Cả (2) và (3) đều là lỗi cũng có thể xảy ra trong hệ thống tập trung, trong khi lỗi liên kết mạng chỉ có thể xảy ra trong hệ thống được phân phối mạng.

* 1. Là nó luôn chủ yếu đến biết điều đó các tin nhắn bạn có gởi có đã đến đích một cách an toàn? Nếu câu trả lời của bạn là *có, hãy* giải thích tại sao. Nếu câu trả lời của bạn là *không,* hãy đưa ra các ví dụ.

**Trả lời:** Không. Nhiều thu thập trạng thái chương trình công việc từ các giả thiết điều đó gói tin có thể không phải là nhận qua các Nơi Đến hệ thống. Các chương trình này thường *phát* một gói tin và giả định rằng ít nhất một số hệ thống khác trên mạng của họ sẽ nhận được thông tin. Đối với lập trường, một daemon trên mỗi hệ thống có thể phát đi mức trung bình tải của hệ thống và số lượng người dùng. Thông tin này có thể được sử dụng để lựa chọn mục tiêu di cư theo quy trình. Một ví dụ khác là một chương trình xác định nếu Một Xa xôi Địa điểm Là cả hai đang chạy và có thể truy cập kết thúc các mạng. Nếu nó gửi một truy vấn và không nhận được trả lời, nó biết rằng hệ thống hiện không thể đạt.

* 1. Hiện nay một thuật toán vì tái tạo lại Một hợp lý nhẫn sau Một quy trình trong chiếc nhẫn không thành công.

**Trả lời:** Các hệ thống phân tán thông thường sử dụng một quy trình điều phối viên thực hiện các chức năng cần thiết của các quy trình khác trong hệ thống. Điều này sẽ bao gồm việc thực thi loại trừ lẫn nhau và —trong trường hợp này là một chiếc nhẫn

—Đặt lại mã thông báo bị mất.

Một lược đồ tương tự như thuật toán vòng được trình bày trong Phần 18.6.2 có thể được sử dụng. Thuật toán là sau:

Các **nhẫn thuật toán** giả định điều đó các liên kết Chúng tôi một chiều và quy trình đó gửi của chúng tin nhắn đến các hàng xóm trên của chúng bên phải. Các Dữ liệu chính kết cấu đã sử dụng qua các thuật toán Là các **tích cực danh sách** , Một danh sách điều đó chứa các số ưu tiên của tất cả các tiến trình đang hoạt động trong hệ thống khi thuật toán kết thúc; mỗi quy trình duy trì danh sách hoạt động của riêng nó.

* + 1. Nếu quá trình *P i* phát hiện lỗi điều phối viên, nó tạo ra một danh sách hoạt động ban đầu trống. Sau đó, nó sẽ gửi một thông điệp *chọn* ( *i* ) đến người hàng xóm bên phải của nó và thêm số *i* vào hoạt động của nó danh sách.

**62 Chương 16 Hệ thống phân tán cấu trúc**

* + 1. Nếu *P i* nhận được một thông báo *chọn* ( *j* ) từ quy trình bên trái, nó phải trả lời một trong ba cách:
       1. Nếu cái này Là các đầu tiên *trúng tuyển* tin nhắn nó có đã xem hoặc gởi, *P i* tạo ra một Mới tích cực danh sách với các con số *tôi* và *j.* Nó sau đó gửi tin nhắn *chọn* ( *i* ), tiếp theo là tin nhắn *bầu* ( *j* ).
       2. Nếu *ij* , tức là tin nhắn nhận được không chứa *P i số* của , sau đó là *P i* thêm *j* vào danh sách hoạt động của nó và chuyển tiếp tin nhắn sang bên phải của nó hàng xóm.

/=

* + - 1. Nếu *i* = *j* , tức là *P i* nhận được thông báo *chọn* ( *i* ), sau đó hoạt động danh sách vì *P i* Hiện nay chứa đựng các con số của tất cả các các quy trình hoạt động trong hệ thống. Quy trình *P i* bây giờ có thể xác định số lượng lớn nhất trong danh sách hoạt động để xác định điều phối viên mới quy trình.
  1. Xem xét Một phân phối hệ thống với hai các trang web, MỘT và B. Xem xét liệu trang web A có thể phân biệt giữa các tiếp theo:
     1. B đi xuống.
     2. Liên kết giữa A và B đi xuống.
     3. B cực kỳ quá tải và thời gian phản hồi của nó lâu hơn 100 lần thông thường.

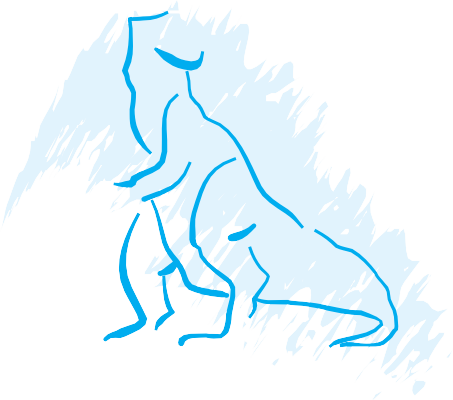
Câu trả lời của bạn có ý nghĩa gì đối với việc khôi phục trong hệ thống phân tán?

**Trả lời:** Một kỹ thuật sẽ là vì B đến định kỳ gửi Một Tin nhắn *I-up* cho A cho biết nó vẫn còn sống. Nếu A không nhận được *thông* báo xác nhận, nó có thể cho rằng B — hoặc liên kết mạng — bị ngắt. Ghi chú điều đó một *Tôi đang lên* tin nhắn làm không phải cho phép MỘT đến phân biệt giữa từng loại hư hỏng. Một kỹ thuật cho phép A tốt hơn để xác định xem mạng có bị trục trặc hay không là gửi một tin nhắn *Có* sẵn cho B bằng cách sử dụng một tuyến đường thay thế. Nếu nó nhận được phản hồi, nó có thể xác định rằng thực sự liên kết mạng bị ngắt và B là hướng lên.

Nếu chúng ta giả sử rằng A biết B đang hoạt động và có thể truy cập được (thông qua cơ chế *I-up* ) và rằng A có một số giá trị *N* cho biết một phản hồi bình thường thời gian, MỘT có thể màn hình các phản ứng thời gian từ B và so sánh các giá trị với *N* , cho phép A xác định xem B có bị quá tải hay không không phải.

Ý nghĩa của cả hai kỹ thuật này là A có thể chọn một máy chủ khác — có thể nói là C — trong hệ thống nếu B gặp sự cố, không thể truy cập hoặc quá tải.

*Hệ thống Linux*



C*2*HA P*1*TER

### Bài tập thực hành

* 1. Các mô-đun nhân có thể tải động mang lại sự linh hoạt khi trình điều khiển được thêm vào hệ thống, nhưng chúng cũng có nhược điểm? Trong bất kì trường hợp nào sẽ Một hạt nhân là biên dịch vào trong Một Độc thân nhị phân tập tin, và khi nào thì tốt hơn nếu giữ nó được chia thành các mô-đun? Giải thích câu trả lời của bạn.

**Trả lời:** Có hai hạn chế chính với việc sử dụng các mô-đun. Đầu tiên là kích thước: quản lý mô-đun sử dụng bộ nhớ nhân không thể gắn thẻ và một nhân cơ bản với một số mô-đun được tải sẽ tiêu tốn nhiều bộ nhớ hơn một nhân tương đương với các trình điều khiển được biên dịch thành chính hình ảnh nhân. Đây có thể là một vấn đề rất nghiêm trọng trên các máy có vật lý hạn chế kỉ niệm.

Hạn chế thứ hai là các mô-đun có thể làm tăng độ phức tạp của quá trình khởi động hạt nhân. Thật khó để tải một tập hợp các mô-đun từ đĩa nếu trình điều khiển cần thiết để truy cập vào đĩa đó một mô-đun cần được tải. Do đó, việc quản lý kernel bootstrap bằng các mô-đun có thể đòi hỏi người quản trị phải làm việc thêm: các mô-đun cần thiết để khởi động cần được đặt vào một ảnh đĩa ram được tải cùng với ảnh hạt nhân ban đầu khi hệ thống được khởi tạo.

Trong một số trường hợp nhất định, tốt hơn là sử dụng nhân mô-đun và trong các trường hợp khác nó Là tốt hơn đến sử dụng Một hạt nhân với nó là thiết bị người lái xe được liên kết trước. Trong đó việc giảm thiểu kích thước của hạt nhân là quan trọng, sự lựa chọn sẽ phụ thuộc vào tần suất sử dụng các trình điều khiển thiết bị khác nhau. Nếu chúng được sử dụng phổ biến, thì các mô-đun không phù hợp. Điều này đặc biệt đúng khi trình điều khiển cần thiết cho chính quá trình khởi động. Mặt khác, nếu một số người lái xe Chúng tôi không phải luôn cần thiết, sau đó các mô-đun cơ chế al-

**71**

**72 Chương 21 Linux Hệ thống**

hạ thấp những trình điều khiển đó được tải và không tải theo yêu cầu, có khả năng cung cấp một khoản tiết kiệm ròng về vật lý kỉ niệm.

Khi một hạt nhân được xây dựng phải có thể sử dụng được trên nhiều loại máy rất khác nhau, thì việc xây dựng nó bằng các mô-đun rõ ràng là tốt hơn so với việc sử dụng một hạt nhân duy nhất với hàng tá trình điều khiển không cần thiết tiêu tốn bộ nhớ. Điều này đặc biệt xảy ra đối với các hạt nhân được thương mại hóa, nơi ưu tiên hỗ trợ nhiều loại phần cứng nhất có thể theo cách đơn giản nhất có thể.

Tuy nhiên, nếu một nhân đang được xây dựng cho một máy có cấu hình Là đã biết trong tiến bộ, sau đó biên dịch và sử dụng mô-đun có thể đơn giản là một sự phức tạp không cần thiết. Trong những trường hợp như thế này, việc sử dụng các mô-đun có thể là một vấn đề nếm.

* 1. Đa luồng Là Một thông thường đã sử dụng lập trình kỹ thuật. Mô tả ba cách khác nhau mà các luồng có thể được thực hiện. Giải thích làm thế nào những cách đối chiếu đến các Linux **dòng vô tính** cơ chế. Khi nào có thể mỗi cơ chế thay thế tốt hơn hoặc tệ hơn so với việc sử dụng nhái?

**Trả lời: Các** triển khai chủ đề có thể được phân loại rộng rãi thành hai nhóm: dựa trên hạt nhân chủ đề và chế độ người dùng chủ đề. Chế độ người dùng các gói luồng phụ thuộc vào một số hỗ trợ hạt nhân — chẳng hạn, chúng có thể yêu cầu các phương tiện liên kết hẹn giờ — nhưng việc lập lịch giữa các luồng không được thực hiện qua các hạt nhân Nhưng qua một số thư viện của chế độ người dùng mã số. Nhiều luồng trong quá trình triển khai như vậy xuất hiện trong hệ điều hành như một ngữ cảnh thực thi duy nhất. Khi quá trình đa luồng đang chạy, nó quyết định vì chinh no cái mà của nó là chủ đề đến hành hình, sử dụng các bước nhảy không cục bộ để chuyển đổi giữa các luồng theo các quy tắc lập lịch trước hoặc không đặt trước của riêng nó.

Ngoài ra, nhân hệ điều hành có thể cung cấp hỗ trợ cho các luồng chinh no. Ở trong cái này trường hợp, các chủ đề có thể là thực hiện bằng các quy trình riêng biệt chia sẻ một không gian địa chỉ chung hoàn chỉnh hoặc một phần hoặc chúng có thể được triển khai dưới dạng các ngữ cảnh thực thi riêng biệt trong Một Độc thân quy trình. Cái nào đường các chủ đề Chúng tôi được tổ chức, chúng xuất hiện dưới dạng bối cảnh thực thi hoàn toàn độc lập với ứng dụng.

Việc triển khai kết hợp cũng có thể thực hiện được, trong đó một số lượng lớn các luồng được tạo sẵn cho ứng dụng bằng cách sử dụng một số lượng nhỏ hơn các luồng hạt nhân. Các luồng người dùng có thể chạy được chạy bởi luồng nhân có sẵn đầu tiên.

Trong Linux, các luồng được thực hiện bên trong hạt nhân bằng một cơ chế sao chép tạo ra một quy trình mới trong cùng một máy ảo Không gian địa chỉ bằng các cha mẹ quy trình. không giống một số dựa trên hạt nhân chủ đề các gói, nhân Linux không tạo ra bất kỳ sự phân biệt nào giữa các luồng và các quy trình: Một chủ đề Là đơn giản Một quy trình điều đó đã làm không phải tạo ra Một Mới không gian địa chỉ ảo khi nó là khởi tạo.

Ưu điểm chính của việc triển khai các luồng trong nhân thay vì trong thư viện chế độ người dùng là:

* + - hệ thống luồng nhân có thể tận dụng nhiều bộ xử lý nếu chúng có sẵn; và
    - nếu một luồng chặn trong quy trình dịch vụ nhân (ví dụ: một cuộc gọi hệ thống hoặc lỗi trang), các chuỗi khác vẫn có thể chạy.

**Thực hành Bài tập 73**

Một lợi thế ít hơn là khả năng gán các thuộc tính bảo mật khác nhau cho mỗi luồng.

Chế độ người dùng triển khai làm không phải có này thuận lợi. Bởi vì như vậy triển khai chạy toàn bộ ở trong Một Độc thân hạt nhân chấp hành bối cảnh, chỉ một một chủ đề có thể bao giờ là đang chạy tại Một lần, thậm chí nếu nhiều CPU có sẵn. Vì lý do tương tự, nếu một luồng tham gia lệnh gọi hệ thống, không luồng nào khác có thể chạy cho đến khi lệnh gọi hệ thống đó hoàn tất. Do đó, một luồng thực hiện đọc đĩa chặn sẽ giữ mọi luồng trong ứng dụng. Tuy nhiên, việc triển khai chế độ người dùng có những lợi thế riêng. Rõ ràng nhất là hiệu suất: việc gọi bộ lập lịch riêng của hạt nhân để chuyển đổi giữa các luồng liên quan đến việc nhập một bảo vệ mới miền bằng các CPU công tắc đến hạt nhân chế độ, nhưng trái lại Việc chuyển đổi giữa các luồng trong chế độ người dùng có thể thực hiện được đơn giản bằng cách lưu và khôi phục các thanh ghi CPU chính. Các luồng chế độ người dùng cũng có thể chiếm ít bộ nhớ hệ thống hơn: hầu hết các hệ thống UNIX sẽ dành ít nhất một trang đầy đủ cho ngăn xếp nhân cho mỗi luồng nhân và ngăn xếp này có thể không có thể phân trang.

Phương pháp kết hợp, triển khai nhiều luồng người dùng trên một con số của hạt nhân chủ đề, cho phép Một sự cân bằng giữa này những đánh đổi cần đạt được. Các luồng nhân sẽ cho phép nhiều luồng tham gia chặn các cuộc gọi nhân cùng một lúc và sẽ cho phép chạy trên nhiều CPU và chuyển đổi luồng chế độ người dùng có thể xảy ra trong mỗi luồng nhân để thực hiện phân luồng nhẹ mà không cần phải có quá nhiều luồng nhân. . Nhược điểm của phương pháp này là phức tạp: việc trao quyền kiểm soát sự cân bằng sẽ làm phức tạp người dùng của thư viện luồng giao diện.

* 1. Nhân Linux không cho phép phân trang ngoài bộ nhớ nhân. Hạn chế này có ảnh hưởng gì đến thiết kế của nhân? Hai ưu điểm và nhược điểm của thiết kế này là gì quyết định?

**Trả lời:** Các sơ cấp va chạm của không cho phép phân trang của hạt nhân kỉ niệm trong Linux là tính không ưu tiên của hạt nhân là bảo quản. Bất cứ quy trình đang lấy Một trang lỗi, cho dù trong hạt nhân hoặc trong người sử dụng chế độ, rủi ro hiện tại lên lịch lại trong khi các yêu cầu dữ liệu Là phân trang trong từ đĩa. Là- vì hạt nhân có thể dựa vào việc không được lên lịch lại trong quá trình truy cập đến nó là cấu trúc dữ liệu chính, các yêu cầu khóa để bảo vệ thanh Liêm của những cấu trúc dữ liệu đó được đơn giản hóa rất nhiều. Mặc dù thiết kế bản thân sự đơn giản là một lợi ích, nó cũng cung cấp một quan trọng perfor- mance thuận lợi trên bộ xử lý đơn máy móc quá hạn đến các thực tế điều đó nó Là không phải cần thiết đến làm thêm vào khóa trên hầu hết Nội bộ dữ liệu cấu trúc. Có một số nhược điểm khi thiếu có thể phân trang hạt nhân bộ nhớ, tuy nhiên. Trước hết, nó áp đặt các ràng buộc đối với số lượng của kỉ niệm điều đó các hạt nhân có thể sử dụng. Nó Là không hợp lý đến giữ hết sức lớn dữ liệu cấu trúc trong không phân trang được kỉ niệm, từ điều đó đại diện thuộc thân thể kỉ niệm điều đó chắc chắn rồi không thể là đã sử dụng vì bất cứ điều gì khác. Cái này có hai tác động: đầu tiên của tất cả, các hạt nhân cần phải tỉa cành trở lại nhiều của nó là Nội bộ dữ liệu cấu trúc thủ công, thay thế của hiện tại có thể đến phụ thuộc trên Một Độc thân ảo- kỉ niệm cơ chế đến giữ thuộc thân thể kỉ niệm sử dụng Dưới điều khiển. Thứ hai, nó làm cho nó không khả thi đến thực hiện chắc chắn Tính năng, đặc điểm điều đó yêu cầu lớn lượng của ảo kỉ niệm trong các hạt nhân, như là bằng các / tmp-

**74 Chương 21 Linux Hệ thống**

hệ thống tệp (một hệ thống tệp dựa trên bộ nhớ ảo nhanh được tìm thấy trên một số

UNIX ).

Lưu ý rằng sự phức tạp của việc quản lý lỗi trang trong khi chạy mã nhân không phải là vấn đề ở đây. Mã nhân Linux đã có thể xử lý các lỗi trang: nó cần có khả năng xử lý các lệnh gọi hệ thống có các đối số tham chiếu bộ nhớ người dùng có thể được phân trang ra đĩa.

* 1. Ba ưu điểm của liên kết động (dùng chung) của các thư viện so với liên kết tĩnh là gì? Hai trường hợp mà liên kết tĩnh được ưu tiên là gì?

**Trả lời:** Ưu điểm chính của thư viện chia sẻ là chúng giảm các kỉ niệm và đĩa khoảng trống đã sử dụng qua Một hệ thống, và họ tăng cường khả năng bảo trì.

Khi các thư viện được chia sẻ đang được sử dụng bởi tất cả các chương trình đang chạy, chỉ có một phiên bản của mỗi quy trình thư viện hệ thống trên đĩa và tại hầu hết một ví dụ trong thuộc thân thể kỉ niệm. Khi nào các thư viện trong câu hỏi là một câu hỏi được sử dụng bởi nhiều ứng dụng và chương trình, khi đó khả năng tiết kiệm đĩa và bộ nhớ có thể khá đáng kể. Ngoài ra, việc khởi động Thời gian để chạy các chương trình mới có thể được giảm bớt, vì nhiều chức năng phổ biến cần thiết của chương trình đó có thể đã được tải vào kỉ niệm.

Khả năng bảo trì cũng là một lợi thế chính của liên kết động so với liên kết tĩnh. Nếu tất cả các chương trình đang chạy đều sử dụng thư viện chia sẻ để truy cập các quy trình thư viện hệ thống của chúng, thì hãy nâng cấp các quy trình đó, hoặc để thêm chức năng mới hoặc đến sửa chữa lỗi, có thể là xong đơn giản qua thay thế điều đó thư viện chia sẻ . Không cần biên dịch lại hoặc liên kết lại bất kỳ ứng dụng nào; bất kỳ chương trình nào được tải sau khi nâng cấp hoàn tất sẽ tự động nhận các phiên bản mới của các thư viện.

Ở đó Chúng tôi khác thuận lợi cũng vậy. MỘT chương trình điều đó sử dụng đã chia sẻ thư viện thường có thể được điều chỉnh cho các mục đích cụ thể đơn giản bằng cách thay thế một hoặc nhiều thư viện của nó, hoặc thậm chí (nếu hệ thống cho phép nó và hầu hết các UNIX bao gồm cả Linux đều có) thêm một thư viện mới vào lúc chạy. Ví dụ, một gỡ lỗi thư viện có thể là thay thế vì Một thông thường một đến dấu vết Một vấn đề trong một ứng dụng. Thư viện được chia sẻ cũng cho phép các tệp nhị phân chương trình được liên kết với mã thư viện thương mại, độc quyền mà không thực sự bao gồm bất cứ của điều đó mã số trong các chương trình của cuối cùng có thể thực thi tập tin. Cái này là quan trọng tại vì trên hầu hết UNIX hệ thống, nhiều của các Tiêu chuẩn thư viện được chia sẻ là sở hữu độc quyền và các vấn đề cấp phép có thể ngăn cản việc đưa mã đó vào các tệp thực thi được phân phối cho thứ ba các bữa tiệc.

Ở trong một số nơi, Tuy nhiên, tĩnh sự liên kết Là thích hợp. Một ví dụ trong giải thoát môi trường vì hệ thống các nhà quản trị. Nếu Một hệ thống người quản lý làm cho Một sai lầm, điều sai, ngộ nhận trong khi cài đặt bất cứ Mới thư viện, hoặc nếu phần cứng phát triển vấn đề, rất có thể các thư viện chia sẻ hiện có bị hỏng. Do đó, thường một tập hợp các tiện ích cứu hộ cơ bản được liên kết tĩnh, để có cơ hội sửa lỗi mà không cần phải dựa vào các thư viện chia sẻ đang hoạt động. một cách chính xác.

Ngoài ra còn có những lợi thế về hiệu suất mà đôi khi làm cho liên kết tĩnh thích hợp hơn trong các trường hợp đặc biệt. Để bắt đầu, liên kết động tăng lên các khởi động thời gian vì Một chương trình, bằng các liên kết cần phải Hiện nay là

**Thực hành Bài tập 75**

xong tại chạy thời gian hơn hơn tại biên dịch thời gian. Năng động sự liên kết có thể đôi khi cũng tăng kích thước tập hợp làm việc tối đa của một chương trình (tổng con số của thuộc thân thể trang của kỉ niệm yêu cầu đến chạy các chương trình). Trong thư viện được chia sẻ, người dùng không có quyền kiểm soát vị trí trong tệp nhị phân thư viện mà các hàm khác nhau nằm ở đâu. Vì hầu hết các hàm không điền chính xác vào một trang đầy đủ hoặc các trang của thư viện, nên việc tải một hàm thường sẽ kết quả trong Đang tải trong các bộ phận của các bao quanh chức năng, cũng vậy. Với liên kết tĩnh, tuyệt đối không có chức năng nào không được ứng dụng tham chiếu (trực tiếp hoặc gián tiếp) cần được tải vào kỉ niệm.

Các vấn đề khác xung quanh liên kết tĩnh bao gồm dễ phân phối: việc phân phối tệp thực thi với liên kết tĩnh sẽ dễ dàng hơn so với liên kết động nếu nhà phân phối không chắc liệu người nhận có cài đặt trước các thư viện chính xác hay không. Cũng có thể có thương mại những hạn chế chống lại phân phối lại một số nhị phân bằng thư viện dùng chung. Ví dụ: giấy phép cho môi trường đồ họa UNIX “ Motif ” cho phép các tệp nhị phân sử dụng Motif được phân phối tự do miễn là chúng được liên kết tĩnh, nhưng các thư viện được chia sẻ có thể không được sử dụng nếu không có giấy phép.

* 1. Đối chiếu các sử dụng của mạng lưới ổ cắm với các sử dụng của đã chia sẻ ký ức như Một cơ chế vì giao tiếp dữ liệu giữa quy trình trên Một máy tính duy nhất. Gì Chúng tôi các thuận lợi của mỗi phương pháp? Khi nào có thể mỗi người ưa thích?

**Trả lời:** Sử dụng ổ cắm mạng thay vì bộ nhớ dùng chung để giao tiếp cục bộ có một số ưu điểm. Ưu điểm chính là giao diện lập trình socket có một tập hợp các tính năng đồng bộ hóa phong phú. Một quy trình có thể dễ dàng xác định thời điểm dữ liệu mới đến trên kết nối socket, lượng dữ liệu hiện có và ai đã gửi dữ liệu đó. Các quá trình có thể chặn cho đến khi dữ liệu mới đến trên ổ cắm hoặc chúng có thể yêu cầu gửi tín hiệu khi dữ liệu đến. Một ổ cắm cũng quản lý các kết nối riêng biệt. Một quy trình có một ổ cắm mở để hoạt động lại có thể chấp nhận nhiều kết nối đến ổ cắm đó và sẽ được thông báo khi các quy trình mới cố gắng kết nối hoặc khi các quy trình cũ ngắt kết nối của chúng.

Bộ nhớ dùng chung không cung cấp các tính năng này. Không có cách nào để một quá trình xác định liệu một quá trình khác đã phân phối hoặc thay đổi dữ liệu trong bộ nhớ dùng chung ngoài việc xem xét nội dung của bộ nhớ đó. Không thể có quy trình chặn và yêu cầu đánh thức khi bộ nhớ dùng chung được phân phối và không có tiêu chuẩn cơ chế vì khác quy trình đến thành lập Một đã chia sẻ liên kết bộ nhớ với một hiện có quy trình.

Tuy nhiên, bộ nhớ chia sẻ có ưu điểm là nó nhanh hơn rất nhiều so với giao tiếp socket trong nhiều trường hợp. Khi dữ liệu được gửi qua một ổ cắm, nó thường được sao chép từ bộ nhớ sang bộ nhớ nhiều lần. Cập nhật bộ nhớ dùng chung không yêu cầu bản sao dữ liệu: nếu một quy trình cập nhật Một dữ liệu kết cấu trong đã chia sẻ kỉ niệm, điều đó cập nhật Là có thể nhìn thấy ngay lập tức đến tất cả khác quy trình chia sẻ điều đó kỉ niệm. Gửi hoặc nhận dữ liệu qua một ổ cắm yêu cầu phải thực hiện lệnh gọi dịch vụ hệ thống hạt nhân khởi xướng các chuyển khoản, Nhưng đã chia sẻ kỉ niệm liên lạc có thể là per- được hình thành hoàn toàn ở chế độ người dùng mà không cần chuyển giao quyền kiểm soát yêu cầu.

**76 Chương 21 Linux Hệ thống**

Ổ cắm liên lạc Là tiêu biểu ưa thích khi nào sự liên quan quản lý là quan trọng hoặc khi có yêu cầu đồng bộ hóa người gửi và người nhận. Ví dụ, các quy trình máy chủ thường sẽ thiết lập một ổ cắm lắng nghe mà máy khách có thể kết nối khi họ muốn sử dụng dịch vụ đó. Sau khi ổ cắm được thiết lập, các yêu cầu riêng lẻ cũng được gửi bằng ổ cắm, để máy chủ có thể dễ dàng xác định khi nào một yêu cầu mới đến và ai là người mà nó đến từ.

Ở trong một số các trường hợp, Tuy nhiên, đã chia sẻ kỉ niệm Là được ưu tiên. Được chia sẻ bộ nhớ thường là giải pháp tốt hơn khi cần chuyển một lượng lớn dữ liệu hoặc khi hai quy trình cần truy cập ngẫu nhiên vào một tập dữ liệu chung lớn. Tuy nhiên, trong trường hợp này , các quá trình giao tiếp vẫn có thể cần một cơ chế bổ sung ngoài bộ nhớ dùng chung để đạt được đồng bộ hóa giữa chúng tôi. Các X Cửa sổ Hệ thống, một đồ họa trưng bày Môi trường vì UNIX , Là Một tốt ví dụ của cái này: hầu hết các yêu cầu đồ họa được gửi qua các ổ cắm, nhưng bộ nhớ dùng chung được cung cấp như một phương tiện truyền tải bổ sung trong các trường hợp đặc biệt, nơi các bitmap lớn sẽ được hiển thị trên các màn. Ở trong cái này trường hợp, Một yêu cầu đến trưng bày các bitmap sẽ vẫn được gửi qua socket, nhưng dữ liệu lớn của chính bitmap sẽ được gửi qua chia sẻ kỉ niệm.

* 1. UNIX thường sử dụng tối ưu hóa bố cục đĩa dựa trên vị trí điều hành của dữ liệu đĩa, nhưng các triển khai hiện đại, bao gồm cả Linux, chỉ cần tối ưu hóa để truy cập dữ liệu tuần tự. Tại sao họ làm như vậy? Truy cập tuần tự có đặc điểm phần cứng nào? Tại sao tối ưu hóa xoay vòng không còn như vậy hữu ích?

**Trả lời:** Đặc tính hiệu suất của phần cứng đĩa đã thay đổi đáng kể trong những năm gần đây. Đặc biệt, nhiều cải tiến đã được giới thiệu để tăng băng thông tối đa có thể đạt được trên đĩa. Trong một hệ thống hiện đại, có thể có một đường ống dài giữa hệ điều hành và đầu đọc-ghi của đĩa. Một yêu cầu I / O đĩa phải chuyển qua bộ điều khiển đĩa cục bộ của máy tính , qua bus logic đến chính ổ đĩa, rồi đến bên trong đĩa, ở đâu ở đó Là rất có thể đến là Một phức tạp người điều khiển điều đó có thể bộ nhớ đệm truy cập dữ liệu và có khả năng tối ưu hóa thứ tự của I / O các yêu cầu.

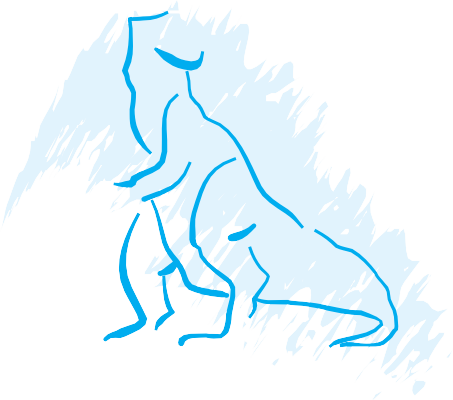
Do sự phức tạp này, thời gian cần thiết để một yêu cầu I / O được xác nhận và yêu cầu tiếp theo được tạo và nhận bởi đĩa có thể vượt xa khoảng thời gian giữa một khu vực đĩa đi qua Dưới các đọc viết cái đầu và các tiếp theo khu vực tiêu đề đến nơi. Để có thể đọc nhiều sector cùng một lúc một cách hiệu quả, các đĩa sẽ sử dụng một bộ nhớ đệm trên đầu đọc. Trong khi một sector đang được chuyển trở lại máy tính chủ, đĩa sẽ bận đọc các sector tiếp theo để đề phòng của Một yêu cầu đến đọc họ. Nếu đọc yêu cầu bắt đầu đến nơi theo thứ tự phá vỡ đường ống dẫn đầu đọc này, hiệu suất sẽ giảm. Do đó, hiệu suất mang lại lợi ích đáng kể nếu hệ điều hành cố gắng giữ các yêu cầu I / O theo trình tự nghiêm ngặt đơn hàng.

Đặc điểm thứ hai của đĩa hiện đại là hình học của chúng có thể rất phức tạp. Số lượng cung trên mỗi xi lanh có thể thay đổi tùy theo vị trí của hình trụ: nhiều dữ liệu có thể bị dồn vào các rãnh dài gần mép đĩa hơn là ở tâm đĩa. Đối với hệ điều hành để tối ưu hóa vị trí luân chuyển của dữ liệu trên như là

**Thực hành Bài tập 77**

đĩa, nó sẽ phải có hiểu biết đầy đủ về hình học này, như ổn bằng các thời gian đặc trưng của các đĩa và nó là bộ điều khiển. Ở trong chung, chỉ một các đĩa của Nội bộ Hợp lý có thể quyết tâm các tối ưu lập lịch I / O và hình dạng của đĩa có khả năng đánh bại bất kỳ nỗ lực nào của hệ điều hành để thực hiện quay tối ưu hóa.

## Windows XP



*2*CH A*2*PTER

### Bài tập thực hành

* 1. XP là loại hệ điều hành nào? Mô tả hai trong số chính của nó Tính năng, đặc điểm.

**Trả lời:** Hệ điều hành đa nhiệm ưu tiên 32/64 bit hỗ trợ nhiều người dùng. (1) Khả năng tự động sửa chữa các sự cố ứng dụng và hệ điều hành. (2) Kết nối mạng tốt hơn và trải nghiệm thiết bị (bao gồm cả nhiếp ảnh kỹ thuật số và băng hình).

* 1. Liệt kê các mục tiêu thiết kế của Windows XP . Mô tả hai trong chi tiết.

**Trả lời:** Thiết kế bàn thắng bao gồm Bảo vệ, độ tin cậy, các cửa sổ và Khả năng tương thích ứng dụng POSIX , hiệu suất cao, khả năng mở rộng, tính di động và hỗ trợ quốc tế. (1) Độ tin cậy được coi là một yêu cầu nghiêm ngặt và bao gồm xác minh trình điều khiển rộng rãi, phương tiện để bắt lỗi lập trình trong mã cấp người dùng và quy trình chứng nhận nghiêm ngặt cho trình điều khiển, ứng dụng và thiết bị của bên thứ ba. (2) Đạt được cao màn biểu diễn yêu cầu kiểm tra của vừa qua vấn đề khu vực như là bằng I / O màn biểu diễn, máy chủ CPU tắc nghẽn, và các khả năng mở rộng của đa luồng và đa xử lý các môi trường.

* 1. Mô tả quá trình khởi động Windows XP hệ thống.

**Trả lời:** (1) Khi phần cứng bật nguồn, BIOS bắt đầu thực thi từ ROM và tải và thi hành các bootstrap người nạp từ các đĩa. (2) Chương trình NTLDR được tải từ thư mục gốc của hệ thống đã xác định thiết bị và xác định cái mà khởi động thiết bị chứa đựng các hệ điều hành. (3) NTLDR tải thư viện HAL , hạt nhân và tổ chức hệ thống. Hệ thống hive chỉ ra các trình điều khiển khởi động cần thiết và tải chúng. (4) Quá trình thực thi hạt nhân bắt đầu bằng cách khởi tạo hệ thống và tạo hai quy trình: quy trình hệ thống chứa tất cả các luồng nhân viên nội bộ và quy trình khởi tạo chế độ người dùng đầu tiên: SMSS . (5) SMSS hơn nữa

**79**

**80 chương 22 Windows XP**

khởi tạo hệ thống bằng cách thiết lập tệp hoán trang và tải trình điều khiển thiết bị. (6) SMSS tạo ra hai quy trình: WINLOGON , mang đến phần còn lại của hệ thống và CSRSS (quy trình hệ thống con Win32).

* 1. Mô tả ba lớp kiến trúc chính của Windows XP .

**Trả lời:** (1) HAL (Lớp trừu tượng phần cứng) tạo ra tính di động của hệ thống khi vận hành bằng cách che giấu sự khác biệt phần cứng so với các lớp trên của hệ điều hành. Các chi tiết quản trị của các tiện ích cấp thấp được cung cấp bởi các giao diện HAL . HAL giới thiệu một giao diện máy ảo điều đó Là đã sử dụng qua các hạt nhân điều phối, các điều hành và trình điều khiển thiết bị. (2) Lớp nhân cung cấp nền tảng cho các chức năng điều hành và hệ thống con chế độ người dùng. Kernel vẫn còn trong bộ nhớ và không bao giờ được sử dụng trước. Các trách nhiệm của nó là lập lịch luồng, xử lý nội dung và ngoại lệ, đồng bộ hóa bộ xử lý mức thấp, và sức mạnh sự thất bại hồi phục. (3) Các điều hành lớp cung cấp Một đặt của dịch vụ đã sử dụng qua tất cả hệ thống con: vật người quản lý, ảo kỉ niệm trình quản lý, trình quản lý quy trình, cơ sở gọi thủ tục cục bộ, trình quản lý I / O , giám sát bảo mật, trình quản lý plug-and-play, đăng ký và khởi động máy.

* 1. Công việc của đối tượng là gì người quản lý?

**Trả lời:** Các đối tượng hiện nay Một chung đặt của hạt nhân chế độ giao diện đến chế độ người dùng các chương trình. Các đối tượng Chúng tôi thao túng qua các tầng điều hành người quản lý đối tượng. Công việc của người quản lý đối tượng là giám sát việc phân bổ và sử dụng tất cả các các đối tượng.

* 1. Người quản lý quy trình cung cấp những loại dịch vụ nào? Thủ tục địa phương là gì gọi?

**Trả lời:** Trình quản lý quy trình cung cấp các dịch vụ để tạo, xóa và sử dụng các quy trình, luồng và công việc. Trình quản lý tiến trình cũng thực hiện xếp hàng và phân phối các lệnh gọi thủ tục không đồng bộ tới các luồng. Cuộc gọi thủ tục cục bộ ( LPC ) là một hệ thống truyền thông điệp. Hệ điều hành sử dụng LPC để chuyển các yêu cầu và kết quả khách hàng và máy chủ quy trình ở trong Một Độc thân máy móc, trong cụ thể là giữa Windows XP hệ thống con.

* 1. Các trách nhiệm của I / O là gì người quản lý?

**Trả lời:** Trình quản lý I / O chịu trách nhiệm về hệ thống tệp, trình điều khiển thiết bị và trình điều khiển mạng. Trình quản lý I / O theo dõi trình điều khiển thiết bị, trình điều khiển bộ lọc và hệ thống tệp nào được tải và quản lý bộ đệm cho các yêu cầu I / O. Ngoài ra, nó còn hỗ trợ cung cấp I / O tệp được ánh xạ bộ nhớ và kiểm soát trình quản lý bộ nhớ cache cho toàn bộ hệ thống I / O.

* 1. Windows XP có cung cấp bất kỳ quy trình chế độ người dùng nào cho phép nó chạy các chương trình không đã phát triển vì khác điều hành hệ thống? Diễn tả hai của các hệ thống con này.

**Trả lời:** Thuộc về môi trường hệ thống con Chúng tôi chế độ người dùng quy trình xếp lớp trên các dịch vụ thực thi gốc để cho phép Windows XP chạy các chương trình được phát triển cho các hệ điều hành khác. (1) Một ứng dụng Win32 được gọi là máy DOS ảo ( VDM ) được cung cấp như một quy trình chế độ người dùng đến chạy MS-DOS các ứng dụng. Các VDM có thể hành hình hoặc thi đua Intel 486 hướng dẫn và cũng thế cung cấp thói quen đến thi đua MS-DOS BIOS

**Thực hành Bài tập 81**

dịch vụ và cung cấp trình điều khiển ảo cho màn hình, bàn phím và giao tiếp các cổng. (2) Windows-on-windows ( WOW32 ) cung cấp nhân và sơ khai thói quen vì các cửa sổ 3.1 chức năng. Các sơ khai thói quen gọi các chương trình con Win32 thích hợp , chuyển đổi địa chỉ 16 bit thành 32 bit các địa chỉ.

* 1. Windows XP hỗ trợ những loại mạng nào ? Làm thế nào để Win- dows XP thực hiện chuyên chở các giao thức? Diễn tả hai mạng lưới pro- tocol.

**Trả lời:** Hỗ trợ được cung cấp cho cả mạng ngang hàng và máy khách-máy chủ. Các giao thức vận tải được thực hiện dưới dạng trình điều khiển. (1) Gói TCP / IP bao gồm hỗ trợ SNMP , DHCP , WINS và N et BIOS . (2) Điểm - điểm đào hầm giao thức Là cung cấp đến giao tiếp giữa truy cập từ xa mô-đun đang chạy trên các cửa sổ XP may chủ và khác hệ thống khách hàng kết nối kết thúc các Internet. Sử dụng cái này kế hoạch, mạng riêng ảo đa giao thức ( VPN ) được hỗ trợ qua Internet.

* 1. tên NTFS được tổ chức như thế nào? Diễn tả.

**Trả lời:** Các NTFS không gian tên Là được tổ chức bằng Một hệ thống cấp bậc của các thư mục trong đó mỗi thư mục sử dụng cấu trúc dữ liệu cây B + để lưu chỉ mục tên tệp trong thư mục đó . Gốc chỉ mục của một thư mục chứa cấp cao nhất của cây B +. Mỗi mục trong thư mục chứa tên và tập tin thẩm quyền giải quyết của các tập tin bằng ổn bằng các cập nhật dấu thời gian và tập tin kích cỡ.

* 1. NTFS xử lý cấu trúc dữ liệu như thế nào? Làm cách nào để phục hồi NTFS sau sự cố hệ thống? Điều gì được đảm bảo sau khi quá trình khôi phục diễn ra? **Trả lời:** Ở trong NTFS , tất cả hệ thống tập tin dữ liệu kết cấu cập nhật Chúng tôi thực hiện các giao dịch bên trong. Trước khi cấu trúc dữ liệu bị thay đổi, giao dịch sẽ ghi một bản ghi nhật ký chứa thông tin làm lại và hoàn tác. Bản ghi cam kết được ghi vào nhật ký sau khi giao dịch thành công. Sau một vụ va chạm các tập tin hệ thống có thể là phục hồi đến Một đặc tình trạng qua xử lý khúc gỗ Hồ sơ, đầu tiên làm lại hoạt động vì cam kết giao dịch và hoàn tác các hoạt động đối với các giao dịch không được cam kết thành công. Cái này kế hoạch làm không phải Bảo hành điều đó người sử dụng tập tin nội dung Chúng tôi Chính xác sau một hồi phục, Nhưng hơn điều đó các hệ thống tập tin dữ liệu cấu trúc (tập tin siêu dữ liệu) là không bị hư hại và phản chiếu một số đặc tình trạng điều đó tồn tại trước vụ va chạm.
  2. Windows XP phân bổ người dùng như thế nào kỉ niệm?

**Trả lời:** Người sử dụng kỉ niệm có thể là phân bổ theo đến vài lược đồ: bộ nhớ ảo , tệp ánh xạ bộ nhớ, đống, và tỷ lệ cục bộ luồng.

* 1. Mô tả một số cách ứng dụng có thể sử dụng bộ nhớ thông qua Win32 API .

**Trả lời:** (1) Bộ nhớ ảo cung cấp một số chức năng cho phép ứng dụng dự trữ và giải phóng bộ nhớ, chỉ định địa chỉ ảo mà bộ nhớ được cấp phát. (2) Một tệp có thể được ánh xạ bộ nhớ vào không gian địa chỉ, cung cấp một phương tiện để hai tiến trình chia sẻ bộ nhớ. (3) Khi một quy trình Win32 được khởi tạo, nó được tạo bằng Một mặc định đống. Riêng tư đống có thể là tạo điều đó cung cấp vùng của

**82, chương 22 Windows XP**

để dành Địa chỉ khoảng trống vì các ứng dụng. Chủ đề sự quản lý các chức năng được cung cấp để cấp phát và kiểm soát quyền truy cập luồng vào các heap riêng tư. (4) Cơ chế lưu trữ cục bộ luồng cung cấp cách thức cho dữ liệu toàn cục và dữ liệu tĩnh hoạt động bình thường trong môi trường đa luồng. Lưu trữ khóa luồng phân bổ lưu trữ toàn cầu trên mỗi luồng nền tảng.