**I, Introduction**

1.1 Ba mục đích chính của hệ điều hành là gì?

**Trả lời:**

• Cung cấp môi trường cho người dùng máy tính thực thi các chương trình trên phần cứng máy tính một cách thuận tiện và hiệu quả.

• Phân bổ các tài nguyên riêng biệt của máy tính khi cần thiết để giải quyết vấn đề được đưa ra. Quá trình phân bổ phải công bằng và hiệu quả nhất có thể.

• Là một chương trình điều khiển, nó phục vụ hai chức năng chính: (1) giám sát việc thực thi các chương trình của người dùng để ngăn ngừa lỗi và sử dụng máy tính không đúng cách, và (2) quản lý hoạt động và điều khiển các thiết bị I / O.(vào ra )

1.2 Sự khác biệt chính giữa hệ điều hành cho máy tính lớn và máy tính cá nhân là gì?

**Trả lời:**

Nói chung, hệ điều hành cho hệ thống lô (hàng loạt) có yêu cầu đơn giản hơn so với máy tính cá nhân. Hệ thống hàng loạt không phải quan tâm đến việc tương tác với người dùng nhiều như máy tính cá nhân. Do đó, hệ điều hành cho PC phải quan tâm đến thời gian phản hồi cho người dùng tương tác. Hệ thống hàng loạt không có các yêu cầu như vậy. Một hệ thống theo lô thuần túy cũng có thể không xử lý việc chia sẻ thời gian, trong khi một hệ điều hành phải chuyển đổi nhanh chóng giữa các công việc khác nhau.

1.3 Liệt kê bốn bước cần thiết để chạy một chương trình trên một máy hoàn toàn chuyên dụng.

**Trả lời:**

a. Dự trữ thời gian máy.

b. Tải chương trình vào bộ nhớ theo cách thủ công.

C. Tải địa chỉ bắt đầu và bắt đầu thực hiện.

d. Giám sát và điều khiển việc thực thi chương trình từ bảng điều khiển.

1.4 Chúng tôi đã nhấn mạnh sự cần thiết của một hệ điều hành để sử dụng hiệu quả phần cứng máy tính. Khi nào thì thích hợp để hệ điều hành từ bỏ nguyên tắc này và "lãng phí" tài nguyên? Tại sao một hệ thống như vậy không thực sự lãng phí?

**Trả lời:**

Hệ thống một người dùng nên sử dụng tối đa hệ thống cho người dùng. GUI có thể “lãng phí” chu kỳ CPU, nhưng nó tối ưu hóa sự tương tác của người dùng với hệ thống.

1.5 Khó khăn chính mà một lập trình viên phải vượt qua khi viết một hệ điều hành cho môi trường thời gian thực là gì?

**Trả lời:**

Khó khăn chính là giữ hệ điều hành trong các giới hạn thời gian cố định của hệ thống thời gian thực. Nếu hệ thống không hoàn thành một nhiệm vụ trong một khung thời gian nhất định, nó có thể gây ra sự cố cho toàn bộ hệ thống mà nó đang chạy. Do đó, khi viết hệ điều hành cho hệ thống thời gian thực, người viết phải chắc chắn rằng các lược đồ lập lịch của mình không cho phép thời gian phản hồi vượt quá giới hạn thời gian.

1.6 Xem xét các định nghĩa khác nhau của hệ điều hành. Xem xét liệu hệ điều hành có nên bao gồm các ứng dụng như trình duyệt Web và chương trình thư hay không. Tranh luận cả điều nên và điều gì không nên, và hỗ trợ câu trả lời của bạn.

**Trả lời:**

**Điểm**. Các ứng dụng như trình duyệt web và công cụ email đang ngày càng đóng vai trò quan trọng trong các hệ thống máy tính để bàn hiện đại. Để thực hiện vai trò này, chúng nên được kết hợp như một phần của hệ điều hành. Bằng cách đó, chúng có thể cung cấp hiệu suất tốt hơn và tích hợp tốt hơn với phần còn lại của hệ thống. Ngoài ra, các ứng dụng quan trọng này có thể có giao diện giống như phần mềm hệ điều hành.

**Điểm đối âm.** Vai trò cơ bản của hệ điều hành là quản lý tài nguyên hệ thống như CPU, bộ nhớ, thiết bị I / O, v.v. Ngoài ra, vai trò của nó là chạy các ứng dụng phần mềm như trình duyệt web và ứng dụng email. Bằng cách kết hợp các ứng dụng như vậy vào hệ điều hành, chúng tôi tạo gánh nặng cho hệ điều hành với chức năng bổ sung. Gánh nặng như vậy có thể dẫn đến việc hệ điều hành thực hiện công việc quản lý tài nguyên hệ thống kém khả quan. Ngoài ra, chúng tôi tăng kích thước của hệ điều hành do đó làm tăng khả năng hệ thống bị treo và vi phạm bảo mật.

1.7 Sự phân biệt giữa chế độ hạt nhân và chế độ người dùng hoạt động như một dạng hệ thống bảo vệ (bảo mật) thô sơ như thế nào?

**Trả lời:**

Sự khác biệt giữa chế độ hạt nhân và chế độ người dùng cung cấp một hình thức bảo vệ thô sơ theo cách sau. Một số lệnh chỉ có thể được thực thi khi CPU ở chế độ hạt nhân. Tương tự, các thiết bị phần cứng chỉ có thể được truy cập khi chương trình đang thực thi ở chế độ hạt nhân. Việc kiểm soát khi nào ngắt có thể bị ngắt hoặc bị vô hiệu hóa cũng chỉ có thể thực hiện được khi CPU ở chế độ hạt nhân.

Do đó, CPU có khả năng rất hạn chế khi thực thi ở chế độ người dùng, do đó thực thi bảo vệ các tài nguyên quan trọng.

1.8 Hướng dẫn nào sau đây nên được ưu tiên?

a. Đặt giá trị của bộ hẹn giờ. ( bộ định thời )

b. Đọc đồng hồ.

C. Dọn dẹp bộ nhớ.

d. Đưa ra lệnh bẫy.

e. Tắt ngắt.

f. Sửa đổi các mục nhập trong bảng trạng thái thiết bị.

g. Chuyển từ chế độ người dùng sang chế độ hạt nhân.

h. Truy cập thiết bị I / O.

**Trả lời:**

Các thao tác sau cần được đặc quyền: Đặt giá trị của bộ hẹn giờ, xóa bộ nhớ, tắt ngắt, sửa đổi các mục trong bảng trạng thái thiết bị, truy cập thiết bị I / O. Phần còn lại có thể được thực hiện ở chế độ người dùng.

1.9 Một số máy tính đời đầu đã bảo vệ hệ điều hành bằng cách đặt nó vào một phân vùng bộ nhớ mà công việc của người dùng hoặc chính hệ điều hành không thể sửa đổi được. Mô tả hai khó khăn mà bạn nghĩ có thể phát sinh với một kế hoạch như vậy.

**Trả lời:**

Dữ liệu được yêu cầu bởi hệ điều hành (mật khẩu, kiểm soát truy cập, thông tin kế toán, v.v.) sẽ phải được lưu trữ trong hoặc chuyển qua bộ nhớ không được bảo vệ và do đó người dùng trái phép có thể truy cập được.

1.10 Một số CPU cung cấp nhiều hơn hai chế độ hoạt động. Hai cách sử dụng có thể có của nhiều chế độ này là gì?

**Trả lời:**

Mặc dù hầu hết các hệ thống chỉ phân biệt giữa chế độ người dùng và hạt nhân, một số CPU đã hỗ trợ nhiều chế độ. Nhiều chế độ có thể được sử dụng để cung cấp chính sách bảo mật chi tiết hơn. Ví dụ, thay vì phân biệt giữa chế độ chỉ người dùng và hạt nhân, bạn có thể phân biệt giữa các loại chế độ người dùng khác nhau. Có lẽ những người dùng thuộc cùng một nhóm có thể thực thi mã của nhau. Máy sẽ chuyển sang chế độ được chỉ định khi một trong những người dùng này đang chạy mã. Khi máy ở chế độ này, một thành viên của

nhóm có thể chạy mã của bất kỳ ai khác trong nhóm.

Một khả năng khác là cung cấp các phân biệt khác nhau trong mã nhân. Ví dụ: một chế độ cụ thể có thể cho phép trình điều khiển thiết bị USB chạy. Điều này có nghĩa là các thiết bị USB có thể được bảo dưỡng mà không cần phải chuyển sang chế độ hạt nhân, do đó về cơ bản cho phép trình điều khiển thiết bị USB chạy ở chế độ bán người dùng / hạt nhân.

1.11 Bộ hẹn giờ có thể được sử dụng để tính thời gian hiện tại. Cung cấp một mô tả ngắn về cách có thể thực hiện điều này.

**Trả lời:**

Một chương trình có thể sử dụng cách tiếp cận sau để tính thời gian hiện tại bằng cách sử dụng ngắt bộ định thời. Chương trình có thể đặt bộ đếm thời gian trong tương lai và chuyển sang chế độ ngủ. Khi nó được đánh thức bởi ngắt, nó có thể cập nhật trạng thái cục bộ của nó, trạng thái mà nó đang sử dụng để theo dõi số lượng ngắt, nó đã nhận được cho đến nay. Sau đó, nó có thể lặp lại quá trình này liên tục thiết lập các ngắt hẹn giờ và cập nhật trạng thái cục bộ của nó khi các ngắt thực sự được nâng lên.

1.12 Internet là mạng LAN hay mạng WAN?

**Trả lời:**

Internet là một mạng WAN vì các máy tính khác nhau được đặt ở các vị trí địa lý khác nhau và được kết nối bằng các liên kết mạng đường dài

**II.Operating system structures**

2.1 Mục đích của lệnh gọi hệ thống là gì?

**Trả lời:**

Lệnh gọi hệ thống cho phép các tiến trình cấp người dùng yêu cầu các dịch vụ của hệ điều hành.

2.2 Năm hoạt động chính của một hệ điều hành liên quan đến quản lý quá trình là gì?

**Trả lời:**

a. Việc tạo và xóa cả quá trình của người dùng và hệ thống

b. Việc tạm ngừng và tiếp tục các tiến trình

C. Việc cung cấp các cơ chế để đồng bộ hóa tiến trình

d. Cung cấp các cơ chế để giao tiếp tiến trình

e. Cung cấp các cơ chế để xử lý bế tắc

2.3 Ba hoạt động chính của hệ điều hành liên quan đến quản lý bộ nhớ là gì?

**Trả lời:**

a. Theo dõi những phần nào của bộ nhớ hiện đang được sử dụng và bởi ai.

b. Quyết định quy trình nào sẽ được tải vào bộ nhớ khi có dung lượng bộ nhớ.

c. Phân bổ và phân bổ không gian bộ nhớ khi cần thiết.

2.4 Ba hoạt động chính của hệ điều hành liên quan đến quản lý bộ nhớ thứ cấp là gì?

**Trả lời:**

• Quản lý không gian trống.

• Phân bổ lưu trữ.

• Lập lịch đĩa.

2.5 Mục đích của trình thông dịch lệnh là gì? Tại sao nó thường tách biệt với hạt nhân?

**Trả lời:**

Nó đọc các lệnh từ người dùng hoặc từ một tệp lệnh và thực thi chúng, thường bằng cách chuyển chúng thành một hoặc nhiều lệnh gọi hệ thống. Nó thường không phải là một phần của hạt nhân vì trình thông dịch lệnh có thể thay đổi.

2.6 Lệnh gọi hệ thống nào phải được thực thi bởi trình thông dịch lệnh hoặc trình bao để bắt đầu một tiến trình mới?

**Trả lời:**

Trong hệ thống Unix, một lệnh gọi hệ thống rẽ nhánh theo sau một lệnh gọi hệ thống thực thi cần được thực hiện để bắt đầu một quy trình mới. Cuộc gọi fork sao chép quy trình hiện đang thực thi, trong khi lệnh gọi thực thi phủ lên một quy trình mới dựa trên một quy trình thực thi khác trên quy trình gọi.

2.7 Mục đích của các chương trình hệ thống là gì?

**Trả lời:**

Các chương trình hệ thống có thể được coi là một gói các lệnh gọi hệ thống hữu ích. Chúng cung cấp chức năng cơ bản cho người dùng để người dùng không cần phải viết chương trình của riêng mình để giải quyết các vấn đề thông thường.

2.8 Ưu điểm chính của cách tiếp cận phân lớp đối với thiết kế hệ thống là gì? Nhược điểm của việc sử dụng phương pháp phân lớp là gì?

**Trả lời:**

Như trong tất cả các trường hợp thiết kế mô-đun, thiết kế hệ điều hành theo cách mô-đun có một số ưu điểm. Hệ thống dễ gỡ lỗi và sửa đổi hơn vì các thay đổi chỉ ảnh hưởng đến các phần giới hạn của hệ thống chứ không phải chạm vào tất cả các phần của hệ điều hành. Thông tin chỉ được lưu giữ ở những nơi cần thiết và chỉ có thể truy cập được trong một khu vực được xác định và hạn chế, vì vậy bất kỳ lỗi nào ảnh hưởng đến dữ liệu đó phải được giới hạn trong một mô-đun hoặc lớp cụ thể.

2.9 Liệt kê năm dịch vụ được cung cấp bởi một hệ điều hành. Giải thích cách mỗi thứ cung cấp sự tiện lợi cho người dùng. Cũng giải thích những trường hợp nào thì các chương trình cấp người dùng sẽ không thể cung cấp các dịch vụ này.

**Trả lời:**

a. Thực hiện chương trình. Hệ điều hành tải nội dung (hoặc các phần) của tệp vào bộ nhớ và bắt đầu thực thi. Không thể tin cậy một chương trình cấp người dùng để phân bổ thời gian CPU đúng cách.

b. Hoạt động I / O. Đĩa, băng, đường truyền nối tiếp và các thiết bị khác phải được giao tiếp ở mức rất thấp. Người dùng chỉ cần chỉ định thiết bị và thao tác thực hiện trên thiết bị đó, trong khi hệ thống chuyển đổi yêu cầu đó thành các lệnh dành riêng cho thiết bị hoặc bộ điều khiển. Không thể tin cậy các chương trình cấp người dùng để chỉ truy cập vào các thiết bị mà họ phải có quyền truy cập và chỉ truy cập chúng khi chúng không được sử dụng.

c. Thao tác với hệ thống tệp. Có nhiều chi tiết trong việc tạo, xóa, cấp phát và đặt tên tệp mà người dùng không cần phải thực hiện. Các khối không gian đĩa được sử dụng bởi các tệp và phải được theo dõi. Xóa tệp yêu cầu xóa thông tin tệp tên và giải phóng các khối được cấp phát. Các biện pháp bảo vệ cũng phải được kiểm tra để đảm bảo quyền truy cập tệp thích hợp. Các chương trình người dùng không thể đảm bảo tuân thủ các phương pháp bảo vệ cũng như không được tin cậy để chỉ cấp phát các khối miễn phí và phân bổ khối khi xóa tệp.

d. Thông tin liên lạc. Việc truyền thông điệp giữa các hệ thống yêu cầu các thông điệp phải được chuyển thành các gói thông tin, được gửi đến bộ điều khiển mạng, được truyền qua một phương tiện truyền thông và được hệ thống đích tập hợp lại. Đặt hàng gói và sửa dữ liệu phải diễn ra. Một lần nữa, các chương trình người dùng có thể không điều phối quyền truy cập vào thiết bị mạng hoặc chúng có thể nhận các gói dành cho các quá trình khác.

e. Phát hiện lỗi. Phát hiện lỗi xảy ra ở cả cấp độ phần cứng và phần mềm. Ở cấp độ phần cứng, tất cả các quá trình truyền dữ liệu phải được kiểm tra để đảm bảo rằng dữ liệu không bị hỏng trong quá trình truyền. Tất cả dữ liệu trên phương tiện phải được kiểm tra để chắc chắn rằng chúng không thay đổi kể từ khi chúng được ghi vào phương tiện. Ở cấp độ phần mềm, phương tiện phải được kiểm tra tính nhất quán của dữ liệu; chẳng hạn, liệu số lượng khối lưu trữ được phân bổ và chưa được phân bổ có khớp với tổng số trên thiết bị hay không. Ở đó, các lỗi thường không phụ thuộc vào quy trình (ví dụ, dữ liệu bị hỏng trên đĩa), vì vậy phải có một chương trình toàn cầu (hệ điều hành) xử lý tất cả các loại lỗi

2.10 Mục đích của lệnh gọi hệ thống là gì?

**Trả lời:**

Các cuộc gọi hệ thống cho phép các quy trình cấp người dùng yêu cầu các dịch vụ của hệ điều hành.

2.11 Ưu điểm chính của phương pháp microkernel đối với thiết kế hệ thống là gì?

**Trả lời:**

Các lợi ích thường bao gồm những lợi ích sau (a) thêm một dịch vụ mới không yêu cầu sửa đổi hạt nhân, (b) nó an toàn hơn vì nhiều hoạt động được thực hiện ở chế độ người dùng hơn ở chế độ hạt nhân và (c) thiết kế và chức năng hạt nhân đơn giản hơn thường dẫn đến một hệ điều hành đáng tin cậy hơn.

2.12 Tại sao một số hệ thống lưu trữ hệ điều hành trong phần sụn và những hệ thống khác

trên đĩa?

**Trả lời:**

Đối với một số thiết bị nhất định, chẳng hạn như PDA cầm tay và điện thoại di động, đĩa có hệ thống tệp có thể không khả dụng cho thiết bị. Trong tình huống này, hệ điều hành phải được lưu trữ trong phần sụn.

2.13 Làm thế nào một hệ thống có thể được thiết kế để cho phép lựa chọn hệ điều hành để khởi động từ đó? Chương trình bootstrap cần làm gì?

**Trả lời:**

Hãy xem xét một hệ thống muốn chạy cả Windows XP và ba bản phân phối khác nhau của Linux (ví dụ: RedHat, Debian và Mandrake). Mỗi hệ điều hành sẽ được lưu trữ trên đĩa. Trong quá trình khởi động hệ thống, một chương trình đặc biệt (chúng ta sẽ gọi là trình quản lý khởi động) sẽ xác định hệ điều hành nào cần khởi động. Điều này có nghĩa là thay vì khởi động ban đầu vào hệ điều hành, trình quản lý khởi động sẽ chạy trước tiên trong quá trình khởi động hệ thống. Chính trình quản lý khởi động này chịu trách nhiệm xác định hệ thống sẽ khởi động vào. Thông thường, các trình quản lý khởi động phải được lưu trữ tại một số vị trí nhất định của đĩa cứng để được nhận dạng trong quá trình khởi động hệ thống. Trình quản lý khởi động thường cung cấp cho người dùng lựa chọn hệ thống để khởi động vào; trình quản lý khởi động cũng thường được thiết kế để khởi động vào hệ điều hành mặc định nếu người dùng không có lựa chọn nào được chọn.

**III. Processes:**

3.1 Palm OS không cung cấp phương tiện xử lý đồng thời. Thảo luận về ba phức tạp chính mà xử lý đồng thời thêm vào hệ điều hành.

**Trả lời:**

a. Một phương pháp chia sẻ thời gian phải được thực hiện để cho phép từng quy trình trong số một số quy trình có quyền truy cập vào hệ thống. Phương pháp này liên quan đến việc ưu tiên các quá trình không tự nguyện từ bỏ CPU (ví dụ: bằng cách sử dụng lệnh gọi hệ thống) và hạt nhân được đưa vào lại (vì vậy nhiều quá trình có thể đang thực thi mã hạt nhân đồng thời).

b. Các quy trình và tài nguyên hệ thống phải có các biện pháp bảo vệ và phải được bảo vệ lẫn nhau. Bất kỳ quy trình nhất định nào cũng phải bị giới hạn về dung lượng bộ nhớ mà nó có thể sử dụng và các hoạt động mà nó có thể thực hiện trên các thiết bị như đĩa.

c. Cần phải cẩn thận trong nhân để ngăn chặn bế tắc giữa các quy trình, vì vậy các quy trình không chờ đợi các tài nguyên được phân bổ của nhau.

3.2 Bộ xử lý Sun UltraSPARC có nhiều bộ thanh ghi. Mô tả các hành động của một chuyển đổi ngữ cảnh nếu ngữ cảnh mới đã được tải vào một trong các tập đăng ký. Điều gì khác phải xảy ra nếu ngữ cảnh mới nằm trong bộ nhớ chứ không phải trong một tập thanh ghi và tất cả các tập thanh ghi đang được sử dụng?

**Trả lời:**

Con trỏ bộ đăng ký hiện tại của CPU được thay đổi để trỏ đến bộ chứa ngữ cảnh mới, điều này mất rất ít thời gian. Nếu ngữ cảnh nằm trong bộ nhớ, một trong các ngữ cảnh trong tập thanh ghi phải được chọn và được chuyển vào bộ nhớ, và ngữ cảnh mới phải được tải từ bộ nhớ vào tập hợp. Quá trình này mất nhiều thời gian hơn một chút so với các hệ thống có một bộ thanh ghi, tùy thuộc vào cách chọn nạn nhân thay thế.

3.3 Khi một tiến trình tạo một tiến trình mới bằng thao tác fork (), trạng thái nào sau đây được chia sẻ giữa tiến trình mẹ và tiến trình con?

a. Cây rơm

b. Đống

C. Phân đoạn bộ nhớ được chia sẻ

**Trả lời:**

Chỉ các phân đoạn bộ nhớ được chia sẻ được chia sẻ giữa quy trình mẹ và quy trình con mới được phân nhánh. Các bản sao của ngăn xếp và đống được tạo cho quá trình mới được tạo.

3.4 Một lần nữa xem xét cơ chế RPC, hãy xem xét ngữ nghĩa “chính xác một lần”. Thuật toán triển khai ngữ nghĩa này có thực thi chính xác ngay cả khi thông báo “ACK” gửi lại máy khách bị mất do mạng không

vấn đề? Mô tả chuỗi tin nhắn và liệu "chính xác một lần" có còn được lưu giữ hay không.

**Trả lời:**

Ngữ nghĩa “chính xác một lần” đảm bảo rằng một thủ tục từ xa sẽ được thực thi chính xác một lần và chỉ một lần. Thuật toán chung để đảm bảo điều này kết hợp lược đồ xác nhận (ACK) kết hợp với dấu thời gian (hoặc một số bộ đếm gia tăng khác cho phép máy chủ phân biệt giữa các thông báo trùng lặp).

Chiến lược chung là máy khách gửi RPC đến máy chủ cùng với dấu thời gian. Máy khách cũng sẽ bắt đầu đồng hồ thời gian chờ. Sau đó, máy khách sẽ đợi một trong hai lần xuất hiện: (1) nó sẽ nhận được ACK từ máy chủ cho biết rằng thủ tục từ xa đã được thực hiện hoặc (2) nó sẽ hết thời gian. Nếu máy khách hết thời gian chờ, nó giả sử máy chủ không thể thực hiện thủ tục từ xa, do đó máy khách gọi RPC lần thứ hai, gửi một dấu thời gian sau đó. Máy khách có thể không nhận được ACK vì một trong hai lý do: (1) RPC ban đầu không bao giờ được máy chủ nhận, hoặc (2) RPC đã được nhận chính xác — và được thực hiện — bởi máy chủ nhưng ACK đã bị mất. Trong tình huống (1), việc sử dụng ACK cho phép máy chủ cuối cùng nhận và thực hiện RPC. Trong tình huống (2), máy chủ sẽ nhận được một RPC trùng lặp và nó sẽ sử dụng dấu thời gian để xác định nó là một bản sao để không thực hiện RPC lần thứ hai. Điều quan trọng cần lưu ý là máy chủ phải gửi lại ACK thứ hai cho máy khách để thông báo cho máy khách rằng RPC đã được thực hiện.

3.5 Giả sử rằng một hệ thống phân tán dễ bị lỗi máy chủ. Những cơ chế nào sẽ được yêu cầu để đảm bảo ngữ nghĩa “chính xác một lần” để thực thi RPC?

**Trả lời:**

Máy chủ phải theo dõi thông tin lưu trữ ổn định (chẳng hạn như nhật ký đĩa) về những gì các hoạt động RPC đã nhận được, liệu chúng có được thực hiện thành công hay không và kết quả liên quan đến các hoạt động. Khi xảy ra sự cố máy chủ và nhận được thông báo RPC, máy chủ có thể kiểm tra xem RPC đã được thực hiện trước đó hay chưa và do đó đảm bảo ngữ nghĩa "chính xác một lần" cho

thực thi các RPC.

**IV. Threads**

4.1 Cung cấp hai ví dụ lập trình trong đó đa luồng cung cấp hiệu suất tốt hơn giải pháp đơn luồng.

**Trả lời:**

(1) một máy chủ web phục vụ từng yêu cầu trong một chuỗi riêng biệt.

2) (Một ứng dụng song song như phép nhân ma trận trong đó (các phần khác nhau của ma trận có thể được thực hiện song song.

(3) Một (chương trình GUI tương tác chẳng hạn như trình gỡ lỗi trong đó một luồng được sử dụng (để giám sát đầu vào của người dùng, một luồng khác đại diện cho việc chạy (ứng dụng và luồng thứ ba giám sát hiệu suất).

4.2 Sự khác biệt giữa luồng cấp người dùng và luồng cấp nhân là gì? Trong những trường hợp nào thì loại này tốt hơn loại kia?

**Trả lời:**

(1) Hạt nhân không xác định được các luồng cấp người dùng, trong khi hạt nhân nhận biết các luồng nhân. (2) Trên các hệ thống sử dụng ánh xạ M: 1 hoặc M: N, các luồng người dùng được lập lịch bởi thư viện luồng và hạt nhân lập lịch cho các luồng nhân. (3) Các luồng nhân không cần phải được liên kết với một quy trình trong khi mọi luồng người dùng thuộc về một quy trình. Các luồng nhân thường đắt hơn để duy trì các luồng người dùng vì chúng phải được biểu diễn bằng cấu trúc dữ liệu nhân.

4.3 Mô tả các hành động được thực hiện bởi một nhân để chuyển đổi ngữ cảnh giữa các luồng cấp nhân.

**Trả lời:**

Việc chuyển đổi ngữ cảnh giữa các luồng nhân thường yêu cầu lưu giá trị của các thanh ghi CPU từ luồng đang được chuyển ra ngoài và khôi phục các thanh ghi CPU của luồng mới đang được lên lịch.

4.4 Những tài nguyên nào được sử dụng khi một luồng được tạo? Chúng khác với những gì được sử dụng khi một quy trình được tạo ra?

**Trả lời:**

Bởi vì một luồng nhỏ hơn một quy trình, việc tạo luồng thường sử dụng ít tài nguyên hơn so với tạo quy trình. Tạo một quy trình yêu cầu phân bổ một khối điều khiển quy trình (PCB), một cấu trúc dữ liệu khá lớn. PCB bao gồm một bản đồ bộ nhớ, danh sách các tệp đang mở và các biến môi trường. Phân bổ và quản lý bản đồ bộ nhớ thường là

hoạt động tốn nhiều thời gian nhất. Việc tạo luồng người dùng hoặc nhân bao gồm việc phân bổ cấu trúc dữ liệu nhỏ để chứa tập thanh ghi, ngăn xếp và mức độ ưu tiên.

4.5 Giả sử một hệ điều hành ánh xạ các luồng cấp người dùng tới hạt nhân bằng cách sử dụng mô hình nhiều-nhiều và việc ánh xạ được thực hiện thông qua các LWP. Hơn nữa, hệ thống cho phép các nhà phát triển tạo các luồng thời gian thực. Có cần thiết phải liên kết một chuỗi thời gian thực với một LWP không? Giải thích.

**Trả lời:**

Vâng. Thời gian là rất quan trọng đối với các ứng dụng thời gian thực. Nếu một luồng được đánh dấu là thời gian thực nhưng không bị ràng buộc với một LWP, thì luồng đó có thể phải đợi để được gắn vào một LWP trước khi chạy. Xem xét nếu một luồng thời gian thực đang chạy (được gắn với một LWP) và sau đó tiến hành chặn (tức là phải thực hiện I / O, đã được ưu tiên bởi một luồng thời gian thực có mức độ ưu tiên cao hơn, đang chờ khóa loại trừ lẫn nhau, vv) Trong khi luồng thời gian thực bị chặn, LWP mà nó được gắn vào đã được gán cho một luồng khác. Khi luồng thời gian thực đã được lên lịch để chạy lại, trước tiên nó phải đợi để được gắn vào một LWP. Bằng cách liên kết một LWP với một luồng thời gian thực, bạn đảm bảo rằng luồng sẽ có thể chạy với độ trễ tối thiểu khi nó được lên lịch.

4.6 Một chương trình Pthread thực hiện chức năng tổng kết đã được cung cấp trong Phần 4.3.1. Viết lại chương trình này bằng Java.

**Trả lời:**

Vui lòng tham khảo trang web hỗ trợ để biết giải pháp mã nguồn.

**V. CPU Scheduling:**

5.1 Thuật toán lập lịch CPU xác định thứ tự thực hiện các quy trình đã lên lịch của nó. Cho n quá trình được lên lịch trên một bộ xử lý, có bao nhiêu lịch trình khác nhau có thể có? Đưa ra công thức về n.

**Trả lời:**

n! (n giai thừa = n × n - 1 × n - 2 × ... × 2 × 1).

5.2 Xác định sự khác biệt giữa lập kế hoạch ưu tiên và không đặt trước.

**Trả lời:**

Lập lịch trước cho phép một quá trình bị gián đoạn khi đang thực thi, lấy CPU đi và phân bổ nó cho một quá trình khác. Lập lịch không ưu tiên đảm bảo rằng một quá trình chỉ từ bỏ quyền kiểm soát CPU khi nó kết thúc với sự bùng nổ CPU hiện tại.

5.3 Giả sử rằng các quy trình sau đây được thực thi vào thời điểm

chỉ ra. Mỗi quá trình sẽ chạy trong khoảng thời gian được liệt kê. Khi trả lời các câu hỏi, hãy sử dụng lập lịch trình không tính trước và dựa trên tất cả các quyết định dựa trên thông tin bạn có tại thời điểm phải đưa ra quyết định.

Quy trình Thời gian Đến Thời gian Burst

P1 0,0 8

P2 0,4 4

P3 1.0 1

Một. Thời gian quay vòng trung bình cho các quy trình này với thuật toán lập lịch FCFS là bao nhiêu?

b. Thời gian quay vòng trung bình cho các quy trình này với thuật toán lập lịch SJF là bao nhiêu?

C. Thuật toán SJF được cho là sẽ cải thiện hiệu suất, nhưng lưu ý rằng chúng tôi đã chọn chạy quy trình P1 tại thời điểm 0 vì chúng tôi không biết rằng hai quy trình ngắn hơn sẽ đến sớm. Tính toán thời gian quay vòng trung bình sẽ là bao nhiêu nếu CPU không hoạt động trong 1 đơn vị đầu tiên và sau đó lập lịch SJF được sử dụng. Hãy nhớ rằng các quá trình P1 và P2 đang chờ trong thời gian nhàn rỗi này, vì vậy thời gian chờ của chúng có thể tăng lên. Thuật toán này có thể được gọi là lập lịch tri thức trong tương lai.

**Trả lời:**

a. 10,53

b. 9.53

C. 6,86

Hãy nhớ rằng thời gian quay vòng là thời gian kết thúc trừ thời gian đến, vì vậy bạn phải trừ thời gian đến để tính thời gian quay vòng. FCFS là 11 nếu bạn quên trừ thời gian đến.

5.4 Có lợi thế gì khi có các kích thước lượng tử thời gian khác nhau ở các cấp độ khác nhau của hệ thống xếp hàng đa cấp?

**Trả lời:**

Các quy trình cần được bảo dưỡng thường xuyên hơn, chẳng hạn như các quy trình tương tác như trình chỉnh sửa, có thể nằm trong hàng đợi với lượng tử thời gian nhỏ. Các quy trình không cần bảo dưỡng thường xuyên có thể nằm trong hàng đợi với lượng tử lớn hơn, yêu cầu ít công tắc ngữ cảnh hơn để hoàn thành quá trình xử lý và do đó sử dụng máy tính hiệu quả hơn.

5.5 Nhiều thuật toán lập lịch trình CPU được tham số hóa. Ví dụ, thuật toán RR yêu cầu một tham số để chỉ ra lát thời gian. Hàng đợi phản hồi đa cấp yêu cầu các tham số để xác định số lượng hàng đợi, các thuật toán lập lịch cho mỗi hàng đợi, các tiêu chí được sử dụng để di chuyển các quy trình giữa các hàng đợi, v.v. Do đó, những thuật toán này thực sự là một tập hợp các thuật toán (ví dụ, tập hợp các thuật toán RR cho mọi lát cắt thời gian, v.v.). Một bộ thuật toán có thể bao gồm một bộ thuật toán khác (ví dụ, thuật toán FCFS là thuật toán RR với lượng tử thời gian vô hạn). Mối quan hệ nào (nếu có) giữ giữa các cặp tập hợp thuật toán sau đây?

a. Ưu tiên và SJF

b. Hàng đợi phản hồi đa cấp và FCFS

C. Ưu tiên và FCFS

d. RR và SJF

**Trả lời:**

a. Công việc ngắn nhất có mức độ ưu tiên cao nhất.

b. Cấp thấp nhất của MLFQ là FCFS.

C. FCFS dành ưu tiên cao nhất cho công việc đã tồn tại

dài nhất.

d. Không có.

5.6 Giả sử rằng một thuật toán lập lịch (ở cấp độ lập lịch CPU ngắn hạn) ủng hộ những quá trình đã sử dụng ít thời gian xử lý nhất trong quá khứ gần đây. Tại sao thuật toán này lại ưu tiên các chương trình ràng buộc I / O nhưng lại không bỏ đói các chương trình ràng buộc CPU vĩnh viễn?

**Trả lời:**

Nó sẽ ưu tiên các chương trình ràng buộc I / O vì yêu cầu cụm CPU tương đối ngắn của chúng; tuy nhiên, các chương trình ràng buộc CPU sẽ không chết đói vì các chương trình ràng buộc I / O sẽ tương đối thường xuyên rời bỏ CPU để thực hiện I / O của chúng.

5.7 Phân biệt giữa lập lịch PCS và SCS.

**Trả lời:**

Lập lịch PCS được thực hiện cục bộ cho quá trình. Đó là cách thư viện luồng lên lịch cho các luồng trên các LWP có sẵn. Lập lịch SCS là tình huống hệ điều hành lập lịch cho các luồng nhân. Trên các hệ thống sử dụng nhiều-một hoặc nhiều-nhiều, hai mô hình lập kế hoạch về cơ bản là khác nhau. Trên các hệ thống sử dụng một-một, PCS và SCS đều giống nhau.

5.8 Giả sử một hệ điều hành ánh xạ các luồng cấp người dùng tới hạt nhân bằng cách sử dụng mô hình nhiều-nhiều trong đó ánh xạ được thực hiện thông qua việc sử dụng LWP. Hơn nữa, hệ thống cho phép các nhà phát triển chương trình tạo các luồng thời gian thực. Có cần thiết phải liên kết một chuỗi thời gian thực với một LWP không?

**Trả lời:**

Có, nếu không thì một luồng người dùng có thể phải cạnh tranh để giành được một LWP có sẵn trước khi được lên lịch thực sự. Bằng cách liên kết luồng người dùng với một LWP, không có độ trễ khi chờ đợi một LWP có sẵn; luồng người dùng thời gian thực có thể được lên lịch ngay lập tức.