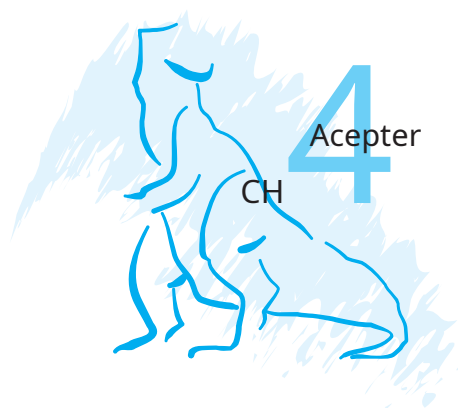


Chủ đề



Bài tập thực hành

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) Máy chủ AWeb 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 làm việc song song, đang chạy (ứng dụng và luồng thứ ba giám sát hiệu suất.

4.2 Hai 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 sang ngữ cảnh chuyển đổi 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 bố khối điều khiển **quá 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 bố và quản lý bản đồ bộ nhớ thường là hoạt động tiêu tốn nhiều thời gian nhất. Việc tạo luồng người dùng hoặc nhân bao gồm việc phân bố 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 với 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 trước 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 sau 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.