

Bài tập thực hành

Môn học: TH Lập Trình Mạng

Chương 4: Socket APIs

1. MỤC TIÊU

Bài thực hành đưa ra với mục tiêu cho sinh viên thực hành với các hàm cơ bản liên quan đến socket như: `socket()`, `bind()`, `listen()`, `accept()`, `connect()`, v.v... Ngoài ra, sinh viên sẽ biết cách xây dựng chương trình client-server trong đó server chạy theo mô hình đa tiến trình, đồng thời bước đầu tìm hiểu cơ chế xử lý các child process rơi vào trạng thái zombie.

2. YÊU CẦU

- Kiến thức cơ bản về lập trình C
- Kiến thức cơ bản về Mạng máy tính
- Máy tính cài đặt Hệ điều hành Linux (khuyến khích distro Ubuntu)
- Sử dụng các hàm cơ bản trong bộ thư viện `sys/socket.h`

3. BÀI THỰC HÀNH

Bài 1: Xử lý tiến trình zombie

Trong bài tập này, chúng ta sẽ thực hiện các nhiệm vụ sau:

Yêu cầu 1: Hãy viết một chương trình đơn giản sử dụng hàm `fork()` để tạo tiến trình con mà không dùng trình xử lý tín hiệu (signal handler). Trong đó tiến trình cha tạo ra một tiến trình con bằng cách sử dụng `fork()`. Tiến trình cha gọi hàm `sleep(30)`; để ngủ 30s và không chờ tiến trình con kết thúc, điều này có thể dẫn đến tiến trình con trở thành zombie.

Trong khi tiến trình cha đang ngủ (30 giây), bạn có thể dùng lệnh sau (mở một cửa sổ lệnh khác) để kiểm tra tiến trình zombie:

```
$ ps aux | grep Z
```

Hoặc lệnh:

```
$ top
```

Câu hỏi 1: kết quả thu được là gì? Tiến trình nào đang ở trạng thái Z? Từ khóa <defunct> nghĩa là gì?

Hãy bổ sung đoạn code sau để dọn dẹp (thu gom) các tiến trình zombie:

```
// Thu gom tiến trình zombie

wait(NULL); // Chờ tiến trình con kết thúc

printf("Tiến trình cha đã thu gom tiến trình zombie.\n");
```

Yêu cầu 2: Làm lại các bước ở trên và cho biết có còn tiến trình zombie nào xuất hiện không? vì sao?

Bài 2: Viết một chương trình client-server trong đó:

- Server là một chương trình đa tiến trình, có thể nhận kết nối với nhiều client cùng lúc, với mỗi client thì server sẽ dùng hàm fork() để sinh ra một tiến trình con để trao đổi thông tin với client đó. Server có kiểm soát thu hồi tiến trình zombie client.
- Server có lưu trữ 10 câu hỏi trắc nghiệm, mỗi câu hỏi trắc nghiệm có 4 đáp án, trong đó có một đáp án đúng.
- Sau khi một client liên kết với server thì sẽ nhận được lần lượt các câu hỏi kèm theo 4 câu trả lời theo thứ tự ngẫu nhiên.
- Client nhận được câu hỏi và danh sách 4 câu trả lời thì sẽ cho người dùng gõ vào câu trả lời và gửi lại cho server. Mỗi câu trả lời đúng thì client đó được 1 điểm.
- Sau khi trả lời xong 10 câu hỏi thì chương trình sẽ gửi lại điểm của client trong lần chơi đó.

Yêu cầu 3: Viết chương trình trên và upload lên hệ thống.