

CHƯƠNG 4: Trao đổi thông tin trong hệ phân tán

Họ tên sinh viên: Phan Thành Đạt

MSSV:20173001

Mã Lớp: 118636

Mã học phần: IT4611

Câu 1:

Không nên triển khai một hệ thống mà tất cả các header của các tầng vào chung một phần header rồi xử lý chung là vì:

- Làm giảm tính trong suốt của các tầng.
- Sự thay đổi ở tầng này thì các tầng khác cũng thấy được.
- Nếu như một tầng thay đổi giao thức thì ta sẽ phải thiết kế lại toàn bộ hệ thống xử lý header mà nếu như không tách riêng tầng phần header ra thì sẽ chỉ thiết kế lại phần xử lý header của tầng đó thôi.

Câu 2:

Trong lời gọi hàm tham chiếu, giá trị của biến bị thay đổi thông qua hàm và ra khỏi hàm giá trị của biến bị thay đổi nên lần gọi thứ 2 giá trị của biến tăng lên là 2.

Trong lời gọi sao chép-phục hồi, giá trị của biến chỉ thay đổi trong hàm và ra ngoài hàm không bị thay đổi nên lần gọi thứ 2 giá trị của biến là 1.

Câu 3:

Một kết nối socket cần 4 thông tin:

- IP máy gửi
- Port máy gửi
- IP máy nhận
- Port máy nhận

Phải đủ 4 thông tin này thì mới phân biệt được các kết nối khác nhau.

Câu 4:

Giao thức request-reply được coi là đồng bộ vì:

- Sau khi client gửi request sang cho server sẽ tự block mình để chờ câu trả lời từ server. Server thực hiện xong request và gửi phản hồi thì client mới thực hiện tiếp.

Giao thức request-reply được coi là tin cậy vì:

- Mỗi request sẽ có 1 reply tương ứng trả lời nên ta sẽ coi gói tin reply đóng vai trò như 1 ACK.

Câu 5:

Hai vấn đề của RPC là:

- Hệ thống không đồng nhất: Do các máy không đồng nhất nên sẽ có không gian nhớ khác nhau, cách biểu diễn thông tin khác nhau, từ đó sẽ gây ra vấn đề.
- Có lỗi xảy ra trong quá trình gọi RPC, ví dụ như một máy bị hỏng hoặc vấn đề mạng có vấn đề => máy gọi sẽ block mãi mãi.

Câu 6:

Vấn đề đối với tham biến:

- Chỉ hoạt động tốt khi hệ thống là đồng nhất.
- Xuất hiện vấn đề khi biểu diễn dữ liệu của 2 hệ thống là khác nhau (ví dụ như 1 bên đầu to 1 bên đầu nhỏ)
- Các kiểu dữ liệu không cùng thuộc 1 kiểu, các kiểu dữ liệu khác nhau được biểu diễn khác nhau (ví dụ 1 bên dùng UTF-8 1 bên dùng UTF-16).

Vấn đề đối với tham chiếu:

- Hai máy khác nhau, không gian nhớ hoàn toàn khác nhau, nếu truyền địa chỉ của máy này sang máy kia thì hoàn toàn không có tác dụng
- Nếu dùng copy/phục hồi sẽ gây tốn băng thông.
- Không thực hiện được tham chiếu đối với những kiểu dữ liệu có cấu trúc, do người lập trình tự định nghĩa ra.

Giải pháp:

- Đặc tả tham số: Sử dụng interface. 2 bên gửi và nhận cùng phải thống nhất về đặc tả tham số(tuân thủ 1 kiểu giao thức), các yếu tố quan trọng để thống nhất.

Câu 7:

	RMI	RPC
Giống	<ul style="list-style-type: none">• Cùng hỗ trợ lập trình với các interface• Dựa trên giao thức yêu cầu/trả lời• Mức độ trong suốt với ứng dụng	
Khác nhau	<ul style="list-style-type: none">• Định danh duy nhất -> Truyền tham chiếu đối tượng• Khai thác hết điểm mạnh của OOP• Chỉ dùng được cho Java	<ul style="list-style-type: none">• Giao diện thống nhất giữa server và client• Đơn giản hóa việc lập trình.• Hỗ trợ mọi ngôn ngữ lập trình.

Câu 8:

Ý nghĩa của tham số backlog: Server quản lý 2 hàng đợi, hàng đợi 1 là hàng đợi chưa hoàn thành, hàng đợi 2 là hàng đợi đã hoàn thành. Tham số backlog để thiết lập số bản ghi lớn nhất của 2 hàng đợi, tức là giới hạn hàng đợi cho các kết nối đến.

Câu 9:

Cơ chế thực thi QoS được thực hiện ở tầng ứng dụng.

Differentiated services: Phân loại lưu lượng và áp dụng các chính sách ưu tiên khác nhau nhằm đảm bảo lưu lượng IP Telephony phải có độ ưu tiên đường truyền cao hơn, thậm chí thông suốt kể cả khi mạng bị nghẽn so với dịch vụ khác.