**Trắc nghiệm**

1. Tính mở của hệ phân tán:

**B**: hệ thống đảm bảo các thành phần được tạo bởi các bên khác nhau vẫn có thể kết nối và làm việc với nhau.

1. Phân tán ngang là gì

**B**: Hệ thống chỉ có 1 mức và “the load” được phân chia vào các máy.

1. Đặc điểm hệ thống CAN

**B**: là hệ thống không tập trung và có cấu trúc. (**Giống chord**)

1. Lợi thế của lightweight processes

**B**: Khi tiến trình có đủ số lightweight process thì nếu có 1 lời gọi blocking system, nó sẽ không tắt toàn bộ tiến trình.

1. Lợi thế của thread-per-connection

Không tốn nhiều overload để hủy và tạo luồng.

1. Vấn đề của truyền tham biến trong RPC

**B**: Client và server có cách biểu diễn dữ liệu khác nhau

1. Trong java, class nào dùng để khởi tạo socket trong giao thức UDP

**A**: DatagramSocket

1. Tính chất nào quyết định RMI ít phổ biến hơn RPC

**A:** Tính mo

1. Không phải nhược điểm của cơ chế chuyển tiếp con trỏ

**C**: Khi một server stub không trỏ đến bất kỳ 1 client stub nào thì nó sẽ bị loại bỏ.

1. Giao thức thông dụng trong broadcating mechanism

**B:** ARP

1. Công dụng của finger table trong chord
2. Toi uu hoa toc do tim kiem…

**C**: lưu trữ vị trí của successer

1. Mechanism nào mobile agent dựa vào

**C**: Autonomous Code

**Tự luận**

**Câu 1:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Distributed OS** | | **Network OS** |
| **Multiproc.** | **Multicomp.** |
| Mức độ trong suốt | Rất cao | Cao | Thấp |
| Một HĐH trên các nút | Có | Có | không |
| Số lượng bản HĐH | 1 | N | N |
| Trao đổi thông tin | Bộ nhớ chia sẻ | Chuyển thông báo | Tệp |
| Quản lý tài nguyên | Toàn cục tập trung | Toàn cục phân tán | Theo nút |
| Co giãn | Không | Có thể | Có |
| Mở | Đóng | Đóng | Mở |

Tại sao DOS không được ứng dụng trong những hệ thống thực tế?

Vì HĐH DOS không có tính co giãn, tính mở rộng và sử dụng 1 HĐH trên các nút. Các hệ thống phân tán hiện nay có rất nhiều máy tính khác nhau về phần cứng, phần mềm nên nếu DOS không có các tính chất nêu trên thì sẽ không thể ứng dụng vào các hệ thống thực tế.

**Câu 2**

Server machine

a)

2.Request service

Daemon

Server

Client

1.Ask for end point

Table of end point

Mỗi máy chủ sẽ có mỗi daemon lưu giữ các end point của các dịch vụ của máy chủ. Khi các Client gửi yêu cầu đến, Daemon sẽ trả lại end point của dịch vụ đó. Sau đố Client sẽ request đến end point đó

-Nhược điểm của nó:

Trong hệ thống UNIX, có rất nhiều server chạy đồng thời , mỗi server có 1 daemon ,điều đó sẽ gây lãng phí ,thay vào đó chỉ cần 1 super server để lắng nghe các yêu cầu end-point

b)

2.Request service

Server

Client

Super Server

1.Ask for end point

Table of end point

Máy khách gửi yêu cầu dịch vụ đến super server. Super server sẽ tìm xem server nào thực hiện dịch vụ mà khách yêu cầu, rồi chuyển yêu cầu đó đến máy chủ tương ứng .Sau đó khách hàng và máy chủ đó sẽ trao đổi qua lại.

Ví du: trong hệ thống UNIX, nó có rất nhiều máy chủ đồng thời và hâu như nó đều thụ động chờ đợi yêu cầu từ client. Thay vì theo dõi rất nhiều quá trình thụ động, sẽ hiệu quả hơn nếu có 1 super server lắng nghe cho mỗi end-point được gắn với 1 dịch vụ cụ thể. Trong UNIX inettd daemon được sử dụng. Inettd lắng nghe các cổng thường được sử dung internet như (http,ftp,..) .Khi 1 yêu cầu đến daemon tạo ra 1 tiến trình để xử lý têu cầu đó.Khi tiến trình thực hiệnx ong, nó sẽ được giải phóng

**Câu 3:**

Bước 1: P gửi thời gian yêu cầu đến S.

Bước 2: Sau khi nhân, S gửi thông điệp với thời gian gửi TQ của đồng hồ vật lý của nó

Bước 3: P nhận thông điệp và cập nhật đồng hồ của nó Tp = TQ + RTT/2

Biết RTT là thời gian cần để gửi thông điệp + thời gian cần để nhận thông điệp.

1. Giá trị Tp của tiến trình P có tuyệt đối chính xác không? Tại sao
2. Gọi độ lệch thời gian δ, có nghĩa: Tp = (TQ + RTT/2) ± δ . Tìm công thức tính δ

**Trả lời**

1. Giá trị TP không là chính xác tuyệt đối vì: Thời gian để nhận và để gửi có thể lệch nhau nên thời gian nhận sẽ ≠ RTT/2. Công thức trên chỉ đúng khi thời gian gửi = thời gian nhận.
2. Có sơ đồ

T3

T2

S

T4

T1

Time nhận

Time gửi

P

Có: RTT = (T2-T1) + (T4-T3)

=> TQ = T3

=> TP = T3 + ± δ = T4

=> =

câu 1 ( chắc thế ) : vì RPC được sur dụng để tăng tính mềm dẻo,trong suốt và hiệu quả cho hệ thống, tạo giao diện và cung cấp dịch vụ cho tầng vận chuyển phía dứoi

Tai Tran

sau đó mô tả tổng quan RPC để minh họa