BÀI TẬP TRÊN LỚP MÔN HỌC:

HỆ PHÂN TÁN CHƯƠNG 3: ĐỊNH DANH TRONG HỆ PHÂN TÁN

HỌ TÊN SV: Lưu Tuấn Hùng

MSSV: 20225131

Câu hỏi 1: Tại sao không thể lấy địa chỉ của Access Point để sử dụng như địa chỉ của thực thể?

Do:

Một thực thể có thể thay đổi địa chỉ Access point và có thể thay đổi nó khi thay đổi địa chỉ vật lý

Access point cũng có thể gán cho các thực thể khác

Một thực thể có thể có nhiều access point

**Câu hỏi 2:** Với việc sử dụng Định Danh, các vấn đề gì cần phải xem xét?

Không gian của định danh có giới hạn

Khi thực thể không tồn tại => tái sử dụng định danh ?

Trùng định danh

**Câu hỏi 3:** Xét một thực thể di chuyển từ vị trí A sang vị trí B. Trong quá trình di chuyển thực thể đó có đi qua các nút trung gian nhưng chỉ dừng lại ở đó khoảng thời gian ngắn. Khi đến B, thực thể đó dừng lại. Chúng ta biết rằng việc thay đổi địa chỉ trong một dịch vụ tổ chức vị trí phân cấp (hierarchical location service) là rất mất thời gian để hoàn thành, vì vậy cần tránh làm việc này khi thực thể tạm dừng ở các nút trung gian. Hãy đề xuất một mô hình kết hợp cả dịch vụ tổ chức vị trí phân cấp và cơ chế chuyển tiếp con trỏ (forwarding pointers) để có thể xác định được vị trí của thực thể khi nó ở các nút trung gian.

Đề xuất. Các nút trung gian sẽ phải có nhiệm vụ ghi lại con trỏ trong khoảng thời gian thực thể dịch chuyển sau đó khi đã đến đích thì chỉ cần lưu con trỏ ở A và xóa các con trỏ trung gian

**Câu hỏi 4:** Trình bày một số phương pháp ARP Spoofing để thấy được điểm yếu của phương pháp định danh sử dụng cơ chế quảng bá.

Tấn công man in the midde

Bị bắt trộm gói tin, không bảo mật

**Câu hỏi 5:** Vấn đề còn tồn tại đối với cơ chế *chuyển tiếp con trỏ* (Forwarding Pointer) là gì?

Vấn đề:

Chuỗi con trỏ quá dài

Nếu một nút trung gian bị mất =>mất dấu thực thể

Một nút trung gian có thể phải lưu con trỏ của nhiều thực thể khác

**Câu hỏi 6:** Nhược điểm của giải pháp Home-based là gì? Giải pháp nào để giải quyết nhược điểm đó?

Có thể thực thể đang ở rất gần nhưng để đến được đó lại phải gửi yêu cầu đến một nơi rất xa để nắm được vị trí của thực thể đó sau rồi mới truy cập được thực thể

Giải quyết :

Khi một thực thể đã đi xa home và ở lại một nơi ở một khoảng thời gian, tiến hành cập nhật con trỏ của thực thể đó tại home mới gần nó

**Câu hỏi 7:** Khi áp dụng giải pháp sử dụng hàm băm phân tán vào hệ thống Chord thì nó đã tối ưu cơ chế định danh như thế nào?

Áp dụng giải pháp sử dụng hàm băm phân tán vào hệ thống Chord

* Thiết lập vòng bằng các biến cụ bộ prev(n) và succ(n)
* Sử dụng bảng băm để xác định chỉ succ(k) của tên k
* Xét theo vòng, chiều kim đồng hồ
* Giá trị nằm giữa hai khóa liên tiếp nào thì chọn khó bé hơn
* Sau khi không chọn được ở giữa nào nữa thì đi thẳng tới khóa k
* Ở các vòng tiếp theo, “độ lớn” xếp theo thứ tự (ví dụ 1>28)

**Câu hỏi 8:** Trong *giải pháp phân cấp*, sử dụng cơ chế bộ đệm có tác dụng cải thiện hiệu năng như thế nào? Cho ví dụ.

Cơ chế bộ đêm giúp lưu lại nhưng thực thể được truy cập nhiều hoặc được truy cập gần đây để nếu người dùng có truy cập lại thì có ngay kết quả. Vì có thể việc tìm kiếm trong giải pháp phân cấp sẽ rất mất thời gian nếu đang truy cập một thực thể ở xa thực thể cần truy nhập

**Câu hỏi 9:** So sánh *liên kết vật lý* và *liên kết biểu tượng* trong hệ thống quản lý tệp của UNIX.

|  |  |
| --- | --- |
| Liên kết vật lý | Liên kết biểu tượng |
| * Một liên kết vật lý là quan hệ giữa một thư mục và inode * Có thể có nhiều liên kết vật lý với 1 inode * Liên kết cuối cùng tới inode bị xóa thì các khối liên quan tới inode cũng bị xóa | * Khi tạo liên kết biểu tượng một inode mới được tạo ra * Inode này có tên phân tử được chỉ tới * Khi xóa các file khác không bị ảnh hưởng |

**Câu hỏi 10:** Khi chúng ta thêm 1 node mới vào hệ thống Chord, chúng ta có cần phải cập nhật toàn bộ các bảng finger?

**Trả lời:**

Không cần phải cập nhật toàn bộ các bảng finger, chỉ cần cập nhật cho nút sau và nút trước vì trong hệ thống Chord, mỗi nút sẽ quản lý một khóa của nút trước đó

**Câu hỏi 11**: *Phân giải tên đệ qui* có ưu điểm gì so với *phân giải tên không đệ qui*?

Có đệ quy: Xử lý nhiều ở server, có tổ chức lưu trữ cache giúp trả lời client nhanh hơn, có thể hoạt động tốt hơn không đệ quy đặc biệt khi khoảng cách truyền xa.