Câu hỏi bài tập chương 3: Đặt tên trong hệ thống phân tán

Câu 1:

a) Định nghĩa "tên" (name) trong hệ thống máy tính phân tán

Định nghĩa:

Tên (name) là chuỗi ký hiệu (symbolic string) được dùng để tham chiếu tới một đối tượng (object) trong hệ thống phân tán. Đối tượng này có thể là:

- Tập tin, thư mục
- Máy chủ, dịch vụ
- Ổ đĩa mạng, máy in
- Người dùng, tiến trình, phiên làm việc

Tên không nhất thiết phải chỉ rõ vị trí của đối tượng, mà thường chỉ là cách để nhận dạng và truy cập đến nó thông qua một dịch vụ phân giải tên (name resolution service), ví dụ như DNS.

📌 Ví dụ:

- "www.example.com" là tên đại diện cho một máy chủ web.
- "\\Server\Share\Docs" là tên của tài nguyên chia sẻ trong mạng nội bộ.

---

b) Phân biệt giữa "tên thân thiện" và "định danh"

Tiêu chí	Tên thân thiện (Friendly Name)
Khái niệm	Là tên dễ đọc, dễ nhớ, thân thiện với người dùng.
Đặc điểm	Có thể thay đổi, không nhất thiết phải duy nhất tuyệt đối.
Mục đích sử dụng	Phục vụ người dùng tương tác với hệ thống.
Phân giải (Resolution)	Cần được ánh xạ sang định danh hoặc địa chỉ bởi hệ thống tên (ví dụ: DNS, LDAP).

Câu 2:

a) Phân biệt ba loại tên: phi cấu trúc, có cấu trúc và dựa trên thuộc tính

Loại tên	Đặc điểm chính	Ưu điểm	Hạn chế

1. Tên phi cấu trúc (Flat name)	- Là chuỗi ký hiệu không có cấu trúc nội tại (không phân cấp, không theo quy tắc định nghĩa).- Thường là định danh duy nhất.	- Dễ sinh tự động bằng hệ thống (ví dụ: hash, UUID).- Tránh trùng lặp.	- Không dễ nhớ với con người.- Khó tìm kiếm hoặc phân loại.
2. Tên có cấu trúc (Structured name)	- Có tổ chức phân cấp hoặc quy tắc (hierarchical or human-readable).- Có thể chia nhỏ thành các thành phần.	- Dễ hiểu, dễ nhớ.- Hỗ trợ phân giải theo từng mức độ (như DNS).	- Có thể trùng lặp nếu không quản lý theo không gian tên.- Cần hệ thống phân giải tên phức tạp hơn.
3. Tên dựa trên thuộc tính (Attribute-based name)	- Không dựa vào chuỗi cụ thể, mà dùng tập hợp các thuộc tính mô tả đối tượng để định danh.	- Linh hoạt, mạnh mẽ trong việc tìm kiếm (truy vấn theo điều kiện).- Hữu ích trong môi trường động.	- Không đảm bảo duy nhất.- Cần cơ chế truy vấn và so khớp phức tạp.

b) Ví dụ minh họa cho từng loại tên

1. Tên phi cấu trúc (Flat Name):

- Ví dụ: 5f2d4b9e18af4c22a3f9d9a1efc8a76d
- Ngữ cảnh sử dụng:
    - Object ID trong hệ thống lưu trữ phân tán như Amazon S3, IPFS.
    - UUID (Universally Unique Identifier) trong các hệ thống nhận dạng đối tượng.

2. Tên có cấu trúc (Structured Name):

- Ví dụ: mail.google.com hoặc \\server01\shared\docs\file.txt
- Ngữ cảnh sử dụng:
    - Hệ thống tên miền (DNS).
    - Đường dẫn trong hệ thống file (UNIX: /home/user/docs, Windows: C:\Users\Admin\file.docx).
    - X.500 / LDAP: cn=John Smith, ou=HR, dc=example, dc=com.

3. Tên dựa trên thuộc tính (Attribute-Based Name):

- Ví dụ: {type=printer, location=buildingA, color=true}
- Ngữ cảnh sử dụng:
    - Truy vấn dịch vụ trong middleware hoặc mạng cảm biến: tìm "máy in màu ở tầng 3".
    - Hệ thống phát hiện tài nguyên theo metadata, chẳng hạn như trong UPnP, service discovery hoặc IoT networks.

Câu 4:

a) Giải thích cơ chế con trỏ chuyển tiếp (forwarding pointer)

Khái niệm thực thể di động (mobile entity):

Là đối tượng (ví dụ: tiến trình, agent, thiết bị, dịch vụ…) có thể di chuyển vị trí vật lý hoặc mạng trong hệ thống phân tán — chẳng hạn như:

- Người dùng di chuyển giữa các trạm làm việc (roaming user)
- Thiết bị di động chuyển đổi kết nối
- Mobile agent di chuyển giữa các máy chủ

Cơ chế con trỏ chuyển tiếp (forwarding pointer):

Khi thực thể di chuyển từ nút A đến nút B:

1. Nút A (vị trí cũ) không xóa hoàn toàn thông tin về thực thể.
2. Thay vào đó, nó giữ một con trỏ chuyển tiếp (forwarding pointer) trỏ đến nút B (vị trí mới).
3. Nếu một tiến trình cố gắng truy cập thực thể qua nút A, nó sẽ được chuyển tiếp đến nút B.

👉 Đây là cơ chế đơn giản để duy trì khả năng định vị thực thể khi nó di chuyển, mà không cần cập nhật toàn bộ hệ thống tên toàn cục.

---

b) Phân tích tác động khi chuỗi con trỏ dài hoặc có lỗi

1. Tác động đến tính trong suốt (Transparency):

- Tích cực: Cơ chế forwarding pointer giữ được sự trong suốt về di chuyển (location transparency) – người dùng không cần biết đối tượng đã di chuyển.
- Tiêu cực: Khi chuỗi forwarding trở nên dài (vì thực thể di chuyển nhiều lần), nó làm lộ ra quá trình chuyển tiếp, làm mất tính trong suốt từng phần.

2. Tác động đến hiệu năng (Performance):

- Khi chuỗi con trỏ ngắn: hiệu năng chấp nhận được, độ trễ nhỏ.
- Khi chuỗi dài (nút A → B → C → D...):
    - Mỗi truy cập cần đi qua nhiều bước → tăng độ trễ truy cập.
    - Gây tắc nghẽn tại các nút trung gian.
    - Làm chậm hệ thống truy tìm hoặc định vị.

Một giải pháp là rút ngắn chuỗi bằng cách cập nhật lại trực tiếp từ nút gốc → nút cuối cùng (lazy update hoặc shortening).

3. Tác động đến tính sẵn sàng (Availability):

- Nếu một nút trong chuỗi forwarding bị lỗi hoặc ngắt kết nối:
    - Toàn bộ chuỗi bị gián đoạn.
    - Thực thể không thể được truy cập, mặc dù vẫn đang hoạt động ở nút mới.
    - → Làm giảm tính sẵn sàng của hệ thống.

Cần có cơ chế phát hiện lỗi hoặc cập nhật fallback, ví dụ như cập nhật bảng định danh toàn cục khi di chuyển quá xa.

Câu 9:

a) Định nghĩa "không gian tên có cấu trúc" (structured namespace)

Định nghĩa:

Không gian tên có cấu trúc (Structured Namespace) là tập hợp các tên được tổ chức theo một cấu trúc phân cấp (thường là dạng cây hoặc đồ thị có hướng), trong đó mỗi tên được xác định thông qua một đường dẫn bao gồm các phần tử liên kết (thường là tên thư mục/con).

👉 Cấu trúc này tương tự hệ thống thư mục trong các hệ điều hành (UNIX, Windows), DNS, LDAP…

Đặc điểm chính:

- Có mối quan hệ cha – con giữa các tên.
- Tên được biểu diễn theo đường dẫn (path): ví dụ /home/user/docs/file.txt.
- Hỗ trợ khả năng định danh rõ ràng, dễ tìm kiếm, dễ phân giải.

Lợi ích của namespace có cấu trúc:

- Quản lý tài nguyên hiệu quả, có thể phân quyền, phân vùng.
- Dễ mở rộng: thêm tên mới vào cây hoặc đồ thị mà không ảnh hưởng đến các nhánh khác.

---

b) Phân biệt nút thư mục và nút lá trong đồ thị tên có cấu trúc

Trong một đồ thị không gian tên (naming graph) – thường là cây hoặc DAG (Directed Acyclic Graph), có hai loại nút chính:

1. Nút thư mục (Directory Node):

Đặc điểm	Mô tả
Là nút trung gian trong cây.	Có ít nhất một cạnh đi đến các nút con.
Đại diện cho nhóm/tập hợp tài nguyên hoặc thư mục cha.	Không chứa dữ liệu cuối cùng, mà chứa liên kết đến các nút khác.
Có thể chứa các nút thư mục khác hoặc nút lá.	

2. Nút lá (Leaf Node):

Đặc điểm	Mô tả
Là nút cuối cùng trong đường dẫn.	Không có nút con.
Đại diện cho đối tượng thực sự cần truy cập (tập tin, dịch vụ, thiết bị...).	
Có thể liên kết với dữ liệu cụ thể.	

Câu 10:

a) Khái niệm đường dẫn tuyệt đối và đường dẫn tương đối

1. Đường dẫn tuyệt đối (Absolute Path):

Là đường dẫn xác định vị trí một đối tượng bắt đầu từ nút gốc của không gian tên (tức là từ root), dẫn đến đối tượng đích theo cách đầy đủ và duy nhất.

- Luôn bắt đầu bằng ký hiệu xác định gốc:
  - UNIX/Linux: bắt đầu bằng /
  - Windows: bắt đầu bằng ký hiệu ổ đĩa như C:\

📌 Ví dụ:

- /home/user/docs/file.txt (trong hệ thống file Linux)
- C:\Users\Admin\Desktop\file.docx (trong Windows)
- dns:/www.example.com (trong hệ thống phân giải tên DNS)

2. Đường dẫn tương đối (Relative Path):

Là đường dẫn được xác định tương đối với vị trí hiện tại (current working directory hoặc context), không bắt đầu từ gốc.

- Dựa vào vị trí hiện tại để xác định đích.
- Có thể dùng các ký hiệu như:
  - . (vị trí hiện tại)
  - .. (thư mục cha)

📌 Ví dụ:

- Nếu đang ở thư mục /home/user/, đường dẫn docs/file.txt là tương đối.
- Trong Windows: nếu đang ở C:\Users\Admin, đường dẫn ..\Public\file.txt là tương đối.

---

b) Sự khác nhau giữa tên toàn cục và tên cục bộ

Loại tên	Tên toàn cục (Global Name)	Tên cục bộ (Local Name)
Định nghĩa	Tên có thể được phân giải và truy cập từ bất kỳ đâu trong hệ thống phân tán.	Tên chỉ có ý nghĩa trong một ngữ cảnh (context) cụ thể.
Phạm vi sử dụng	Toàn bộ hệ thống hoặc internet.	Chỉ trong một tiến trình, không gian tên, thư mục hoặc host.
Tính duy nhất	Thường là duy nhất toàn cục.	Có thể trùng lặp ở các không gian khác nhau.
Ví dụ	www.google.com, /home/user/file.txt, C:\Windows\System32	file.txt, printer1, temp

Câu 15:

a) Định nghĩa đặt tên dựa trên thuộc tính (Attribute-Based Naming)

Định nghĩa:

Đặt tên dựa trên thuộc tính là một cơ chế định danh trong đó các đối tượng được truy xuất không phải qua tên cụ thể, mà thông qua một tập các thuộc tính mô tả đặc trưng của đối tượng đó.

Ví dụ:

- {type=printer, location=floor3, color=true}: tìm máy in màu ở tầng 3.
- {owner=admin, extension=.docx, lastModified<2023-01-01}: tìm các file Word cũ của quản trị viên.

b) So sánh ưu và nhược điểm với các kiểu đặt tên khác

Tiêu chí	Đặt tên dựa trên thuộc tính	Đặt tên có cấu trúc	Đặt tên phi cấu trúc
Tính linh hoạt	Cao: dễ tìm theo nhiều tiêu chí	Trung bình: theo phân cấp định sẵn	Thấp: chỉ có thể khớp chính xác
Tính dễ nhớ với người	Trung bình – khó nhớ do không cố định	Dễ nhớ: tên phân cấp, thân thiện	Khó nhớ: thường là chuỗi UUID
Tính duy nhất	Không đảm bảo – có thể có nhiều đối tượng khớp cùng lúc	Có thể tổ chức để đảm bảo duy nhất	Dễ đảm bảo (nếu dùng UUID, hash)
Hiệu quả truy xuất	Tìm kiếm cần so khớp thuộc tính → tốn tài nguyên	Nhanh: dễ phân giải theo cây	Rất nhanh nếu ánh xạ trực tiếp
Phù hợp truy vấn mờ	Rất phù hợp (cho phép tìm theo điều kiện)	Không hỗ trợ tốt truy vấn mờ	Không hỗ trợ truy vấn theo tiêu chí
Tính mở rộng	Cao: thêm thuộc tính không ảnh hưởng đến tên cũ	Trung bình: giới hạn theo phân cấp	Cao nhưng khó tổ chức lâu dài

Câu 17:

1. Kiến trúc phân cấp của dịch vụ thư mục

Dịch vụ thư mục (Directory Service) là hệ thống tổ chức, lưu trữ và cung cấp truy cập đến thông tin định danh và thuộc tính của thực thể trong mạng như người dùng, thiết bị, tài nguyên...

Tổ chức phân cấp:

- Thông tin được sắp xếp trong một cấu trúc cây phân cấp gọi là DIT (Directory Information Tree).
- Mỗi nút trong cây là một entry có DN (Distinguished Name) duy nhất.
- Cây này phản ánh tổ chức theo tên miền, phòng ban, vị trí địa lý...

---

2. Vai trò của DSA và DIT Partition

DSA (Directory System Agent):

- Là thành phần phần mềm chịu trách nhiệm xử lý truy vấn và cập nhật dữ liệu thư mục trong một vùng dữ liệu (DIT cụ thể).
- Tương đương như "máy chủ thư mục" phục vụ một phần không gian tên.

Ví dụ: Mỗi DSA có thể quản lý một phần cây như dc=company,dc=com, hoặc ou=HR,dc=company,dc=com.

DIT Partition (Phân vùng DIT):

- Là cách chia DIT thành nhiều phần nhỏ để:
  - Phân tán dữ liệu giữa các DSA.
  - Tăng khả năng mở rộng và hiệu năng.
  - Giảm tải và hỗ trợ quản trị phân quyền theo vùng.

Một DIT partition được xử lý bởi một DSA.
Cây thư mục tổng thể có thể trải dài qua nhiều DSA khác nhau, mỗi DSA phụ trách một phân đoạn.

---

3. Liên kết nhiều cây (rừng – forest) và điều phối bằng Catalog Server

Forest (Rừng):

- Tập hợp nhiều cây thư mục (tree) độc lập nhau nhưng có mối quan hệ tin cậy (trust).
- Mỗi tree có DIT riêng, tên miền gốc riêng (ví dụ: abc.com, xyz.com).

Một forest có thể chứa nhiều tổ chức (doanh nghiệp con) với cấu trúc riêng nhưng vẫn kết nối với nhau.

Global Catalog Server:

- Là máy chủ đặc biệt lưu trữ thông tin tổng quát (các thuộc tính phổ biến) của mọi đối tượng trong toàn bộ forest.
- Dùng để:
  - Tìm kiếm nhanh thông tin trên toàn bộ forest (ví dụ tìm người dùng trong tổ chức khác).
  - Phân giải định danh chéo domain.
  - Hỗ trợ xác thực trong môi trường nhiều domain.