CISC 434: Naming – II

- Naming and name resolution mechanisms
  - Names, Identifiers, and Addresses
  - Flat Naming
  - Structured Naming
  - Attribute-Based Naming
Naming Types

- **Flat Naming**
  - Systems need to resolve an identifier to the address of its associated entity – an identifier does not know entity location

- **Structured Naming**
  - Organized in a name space (naming graph) in which a node represents a named entity and the label on an edge represents the name under which that entity is known

- **Attribute-Based Naming**
  - Entities are described by a collection of (attribute, value) pairs
Flat Naming – Hierarchical Approaches

- Hierarchical organization of a location service into domains, each having an associated root (directory) node
- Each root node will have a location record for each entity
  - Each record stores a pointer to the directory of the next lower-level sub-domains where that record’s associated entity is currently located

![Diagram of hierarchical organization]

The root directory node dir(T)

Top-level domain T

Directory node dir(S) of domain S

A subdomain S of top-level domain T (S is contained in T)

A leaf domain, contained in S
Hierarchical Approaches – contd.

- An entity may have multiple addresses, for example, if it is replicated – smallest domain containing all those subdomains will have pointers for each subdomain containing an address.
- An example of two addresses in different leaf domains.
Hierarchical Approaches – contd.

Looking up a location

Node knows about E, so request is forwarded to child

Node has no record for E, so that request is forwarded to parent

Look-up request

Domain D
Hierarchical Approaches – contd.

(a) An insert request is forwarded to the first node that knows about entity $E$

(b) A chain of forwarding pointers to the leaf node is created
Structured Naming

- Structured Naming
  - Organized in a name space (naming graph)
    - node represents a named entity
    - label on an edge represents the name under which that entity is known
Structured Naming – Name Spaces

A mechanism for storing and retrieving information about entities by means of names

- Leaf node – a named entity without any outgoing edge
- Directory node – has one or more outgoing edges labeled with name
- Path name – sequence of labels corresponding to the edges in that path
- Absolute path name – if the first name of the naming graph is root of the naming graph
- Relative path name – otherwise
Name Spaces – contd.

- A general naming graph with a single root node
- Directed acyclic graph – can have more than one incoming edge, but no cycle
Name Resolution – Looking Up a Name

- **Closure mechanism**
  - Knowing how and where to start name resolution, specifically deals with finding the initial node in a name space

- **Linking** – using aliases (another name for the same entity)
  - Hard links (in Unix terminology) – allow multiple absolute path names to refer to the same node in the graph (previous diagram)
  - Symbolic link – represent an entity by leaf node (next diagram)
Symbolic Link

The concept of a symbolic link explained in a naming graph
Mounting

Thus far, we have discussed name resolution within a single name space.

Mounted file system - a directory node stores the identifier of the directory node from a different node space (foreign name space).

The stored node identifier is called a mount point, while the directory node in the foreign name space is called a mounting point — usually the root of the foreign name space.
Linking and Mounting – contd.

- Required information for mounting a foreign name space in distributed system
  - The name of an access protocol
  - The name of the server
  - The name of the mounting point in the foreign name space
Mounting remote name spaces through a specific access protocol
The Implementation of a Name Space

- **Implementation by partitioning into layers**
  - Global layer
  - Administrative layer
  - Managerial layer
The Implementation of a Name Space - Partitioned in layers

- **Global layer**
  - Highest level of nodes (the roots and other directory nodes are close to root)
  - Rarely change—stable
  - May represent organizations, groups of organizations, for which names are stored in the name space

- **Administrational layer**
  - Formed by directory nodes managed within a single organization
  - Represents group of entities of same organization or administrative unit
  - Less stable than global layer
Managerial layer

- Includes nodes representing hosts in local area network (shared files such as libraries and binaries, user defined directories and files)
- Typically change regularly
- Maintained by both system administrators and end users
An example partitioning of the DNS name space, including Internet-accessible files, into three layers

Zones – a part of the name space that is implemented by a separate name server
## Comparison of Name Servers in Different Layers

<table>
<thead>
<tr>
<th>Item</th>
<th>Global</th>
<th>Administrative</th>
<th>Managerial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographical scale of network</td>
<td>Worldwide</td>
<td>Organization</td>
<td>Department</td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>Few</td>
<td>Many</td>
<td>Vast numbers</td>
</tr>
<tr>
<td>Responsiveness to lookups</td>
<td>Seconds</td>
<td>Milliseconds</td>
<td>Immediate</td>
</tr>
<tr>
<td>Update propagation</td>
<td>Lazy</td>
<td>Immediate</td>
<td>Immediate</td>
</tr>
<tr>
<td>Number of replicas</td>
<td>Many</td>
<td>None or few</td>
<td>None</td>
</tr>
<tr>
<td>Is client-side caching applied?</td>
<td>Yes</td>
<td>Yes</td>
<td>Sometimes</td>
</tr>
</tbody>
</table>
Summary

Naming and name resolution mechanisms
  - Names, Identifiers, and Addresses
  - Flat Naming
  - Structured Naming
  - Attribute-Based Naming