

Name Services (İsim Servisleri)

- Introduction (Giriş)
- Names, Identifiers and Addresses
- Name services and DNS (İsim servisleri ve DNS)
- Discovery services (Keşif servisleri)
- Summary (Özet)

Amaç

- İsim ve İsimlendirme
- İsim servisini tanıtmak,
- İstemciler neden isim servislerine ihtiyaç duyar
- İsmi verilen kaynak veya nesnelerin adresleri gibi niteliklerini elde etmek
- Ayrı bir hizmet olarak isim servisi.

Learning objectives (Öğrenme hedefleri)

- Dağıtık sistemlerde adlandırma sistemlerinin gereğini anlamak
- İsim alanlarının yapı ve yönetimi gibi dizayn gereksinimleriyle ve desteklenen işlemlerle tanışık olmak.
- İnternet adlandırma servisinin işlevini anlamak – DNS (Domain Name Service)

Names, Identifiers and Addresses in DS

- Dağıtık bir sistemde isimler, çeşitli kaynaklara işaret etmek için kullanılır:
 - Bilgisayarlar, servisler, uzak nesnelere, dosyalar, kullanıcılar.
- Adlandırma, iletişim ve kaynak paylaşımını kolaylaştırdığı için, DS dizaynında önemlidir.
 - Belirli bir web sayfasına erişmek için, URL biçiminde bir isim gerekir.
 - Processler, yönetilen belirli kaynakları tutarlı olarak adlandırılmazlarsa paylaşamazlar
 - Bir DS üzerindeki kullanıcılar, birbirlerini e-mail adresleriyle adlandırmazlarsa iletişim kuramazlar.
- Kimlik belirlemenin yanında nitelikleri tarif etmek için de isimlendirme yapılır

Names, Ids and Addresses: Computers vs. Humans

- Humans like to use meaningful *names* (a string of chars/bits) to refer to an entity (processes, users, etc.)
 - www.csc.liv.ac.uk, My Documents,
- Entities (Internet hosts, printers, routers, files, disks, services) are accessed using *identifiers* (numbers)
 - IP address
 - File descriptor
 - Port number ...
- We need **services** that *bind* names and identifiers
 - To operate on an entity, need to access it (need an access point).
Access point is a special kind of entity in DS (*an address*)

Identifiers:

- A name for an entity that is independent from its addresses (location) is easy and flexible to use (such a name is called location independent). Server running on diff servers
- A Service should be known by a separate name independent of address of associated server.
- Identifiers: special type of (usually, computer readable) name with the following properties:
 - An id *refers* to at most one entity
 - Each entity is referred by at most one id
 - An id always refers to the same entity (never reused)
- Identifier includes or can be transformed to an address for an object. (NFS file handle, Java RMI remote object reference, etc)

Names

- A **name** is human-readable value (usually a string) that can be **resolved** to an identifier or address
 - Internet domain name, file pathname, process number
 - E.g. /etc/passwd, <http://www.csc.liv.ac.uk/>
- For many purposes, names are preferable to identifiers
 - because the binding of the named resource to a physical location is deferred and can be changed (Kaynağın fiziksel konumla bağlantısı değişebilir)
 - because they are more meaningful to users (İsimler kullanıcılar açısından daha anlamlıdır)
- Resource names are **resolved** by name services
 - to give identifiers and other useful attributes
 - A naming system maintains a **name-to-address binding (i.e., a table)**

Examples

<u>Resource</u>	<u>Name</u>	<u>Identifiers</u>
File	Pathname	File w/in a given FS
Process	Process ID	Process on a given computer
Port	Port number	IP port on a given computer
Web page	URL	Resource ID

- Names (e.g. URLs) are bound to objects (e.g. web pages). Names must be resolved before the corresponding objects can be invoked.
- Any process that requires access to a specific resource must possess a name or identifier for it. Ex: URL `http://www.cdk3.net/`
- A name has to be looked up before it can be used.
- A name is said to be **resolved** when it is translated into data about the resource or object

Names and Binding

- Names are bound to the attributes of named objects (and not to any specific implementations.)
- The association between a name and an object is called binding.
- Services are written to map between names and the attributes of objects they refer to.
- Example:
 - domain name → Domain name Service (DNS) maps → attributes of the host computer

Uniform Resource Locator (URL)

- Used for identifying resources in the Internet
- Typed by the protocol field (http, ftp, nfs, etc.)
- Part of the name is service-specific

Name Service

- A name service stores a collection of one or more naming contexts – sets of bindings between textual names and attributes for objects.
- Provides a general naming scheme for entities (such as users and services) that are beyond the scope of a single service.
- Major operation: resolve a name - to look up attributes from a given name
- Other operations required: creating new binding, deleting bindings, listing bound names and adding and deleting contexts.

Name Services

- Name services store information in a place that users, workstations, and applications must have to communicate across the network such as:
 - Machine (host) names and addresses
 - User names
 - Passwords
 - Access permissions
- Name service provides functionality:
 - Associates (**binds**) names with objects
 - Resolves names to objects
 - Removes bindings
 - Lists names
 - Renames

Composed naming domains used to access a resource from a URL

URL'den kaynağa erişmek için, adlandırma alanları kullanılır

