

分散システム 第6回

— 名前 —

大連理工大学・立命館大学 国際情報ソフトウェア学部

大森 隆行

講義内容

■ 名前

➡ ■ 名前・アドレス・識別子

■ 名前付けと名前解決

- フラットな名前付け

- 構造化された名前付け

■ 名前空間の実装

名前・アドレス・識別子

- (分散システムにおける)名前：エンティティ(実体)を識別するために使われるビット列または文字列
 - エンティティ：ユーザ、端末、Webサーバ、メッセージ、プロセス等
 - エンティティは分散システム内で通信相手や処理の対象になりうる → 参照できなければならない
→ そのために名前が必要
 - 名前とエンティティを対応づけることを
名前解決(name resolution)という
- アドレス(address)：エンティティにアクセスするためのアクセスポイントの名前
 - (例) 友達に手紙を送るときに使う住所、
友達に電話をかけるときに使う電話番号

名前・アドレス・識別子

■ 識別子(identifier) :

エンティティを一意に識別可能な名前

- 一つの識別子は複数のエンティティを参照することはない
- 各エンティティはたかだか一つの識別子により参照される たかだか : at most
- ある識別子は常に同じエンティティを参照し、再利用されない
 - 名前、アドレスは再利用される可能性がある

確認問題

- 空欄に当てはまる語句を選択肢から選んで答えよ。
 - 名前：(1)を識別するために使われるビット列や文字列
 - 名前解決：(1)と(2)を対応付けること
 - (3)：エンティティを一意に識別可能な名前
 - (4)：エンティティにアクセスするためのアクセスポイントの名前
- 以下の各文は正しいか。○か×で答えよ。
 - あるエンティティに付けられた名前は、他のエンティティに再利用される可能性がある。
 - あるエンティティの識別子は、他のエンティティに再利用される可能性がある。
 - エンティティは複数の識別子を持つことがある。



講義内容

■ 名前

- 名前・アドレス・識別子

- ➡ ■ 名前付けと名前解決

 - フラットな名前付け

 - 構造化された名前付け

- 名前空間の実装

フラットな名前付け

- フラット(flat): 平らな = 構造化されていない
 - 名前自体に名前解決に役立つ情報は含まれない
 - (例) MACアドレス (media access control address)

```
Wireless LAN adapter ワイヤレス ネットワーク接続:  
接続固有の DNS サフィックス . . . . :  
説明 . . . . . : Intel(R) Dual Band Wireless-AC 3165  
→ 物理アドレス . . . . . : 70-1C-E7- [redacted]
```

通信機器に固有の番号

- 名前解決
 - ブロードキャスト
 - マルチキャスト
 - 転送ポインタ
 - ホームベースアプローチ

フラットな名前付け

■ 名前解決

■ ブロードキャスト

- (例) ARP (address resolution protocol)
同一ネットワーク内に存在するホスト
すべてに対して、指定したIPを持つホストの
MACアドレスを問い合わせる
→ 問い合わせを受信したマシンは自分の
IPアドレスと比較
→ 一致するなら自身のMACアドレスを返す

■ マルチキャスト

- ブロードキャストと比べて、受信ノードを限定できる
- あらかじめマルチキャストグループを形成
→ グループ内のマシンすべてに対して問い合わせ
(マルチキャストアドレスに対して送信)

フラットな名前付け

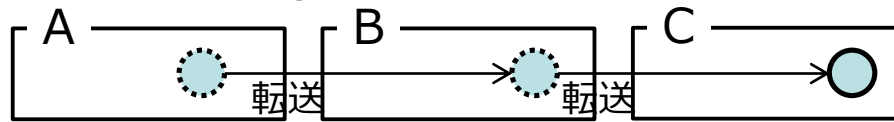
■ 名前解決

■ 転送ポインタ

- エンティティがAからBへ移動する際に、Bの位置を参照するポインタをAに残しておく

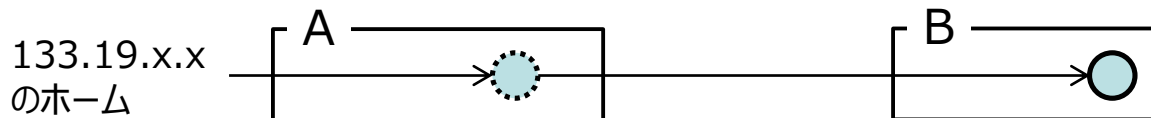
■ 欠点

- 転送ポインタのチェーンが長くなる恐れ
- チェーンの維持(できるだけ短く、堅牢に維持する必要)



■ ホームベースアプローチ

- 各エンティティの最新の現在位置を保持するホームベースを設定



■ 欠点

- ホームの存在が常に保障されていなければならない

構造化された名前付け

UNIXファイルシステム

ディレクトリテーブル

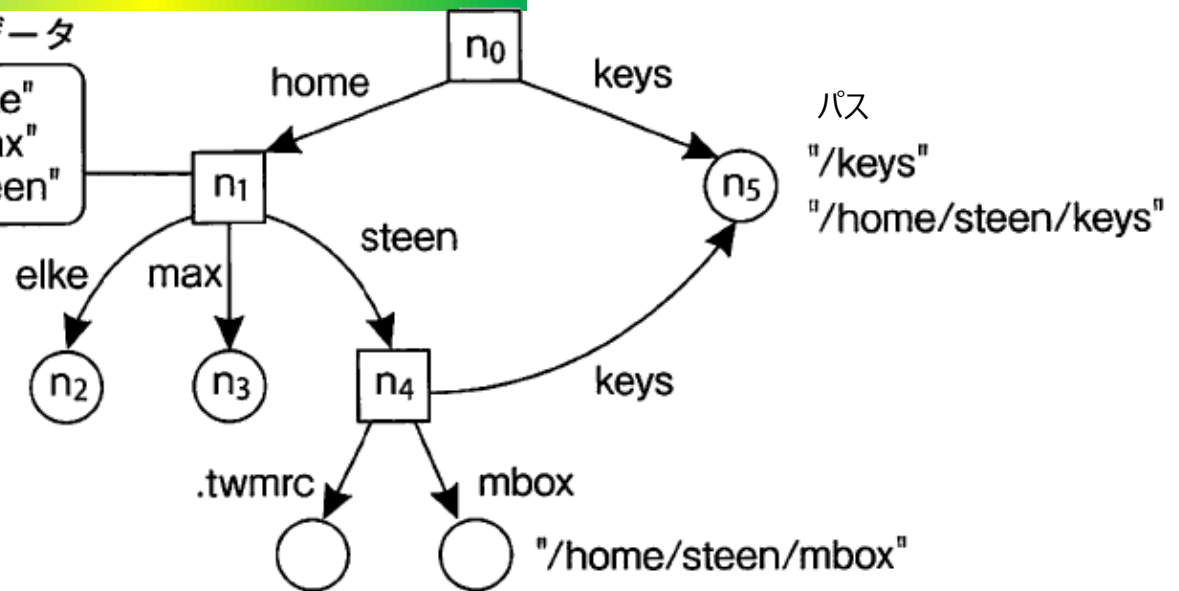
n_1 内のデータ

n_2 : "elke"
 n_3 : "max"
 n_4 : "steen"

リーフノード



ディレクトリノード



■ 2種類のノードからなるラベル付き有向グラフ

- リーフノード(葉ノード): エンティティを示す

- ディレクトリノード: 出力辺の

(辺のラベル、行き先ノードの識別子) の組を表形式で保持

- この表をディレクトリテーブル(directory table)と呼ぶ

■ 各ノードはパスにより識別できる

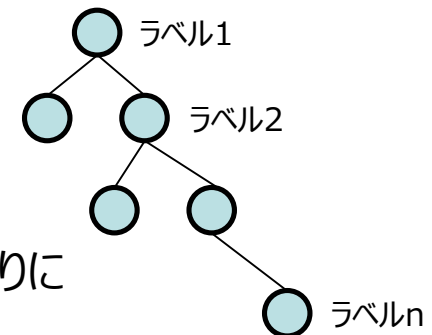
(1ノードに複数のパスが対応づけられることもある)

- 絶対パス: ルート(根)を起点とするパス

- 相対パス: ルート以外を起点とするパス

構造化された名前付け

- 名前 $N := \langle \text{ラベル1}, \text{ラベル2}, \dots, \text{ラベルn} \rangle$
 - まずラベル1が付いたノードを探す
 - ラベル1のディレクトリテーブルからラベル2を探す ...
 - 最後に、ラベルnの付いたノードの識別子を取得
 - (例) UNIXファイルシステムの場合は、inodeのインデックス番号



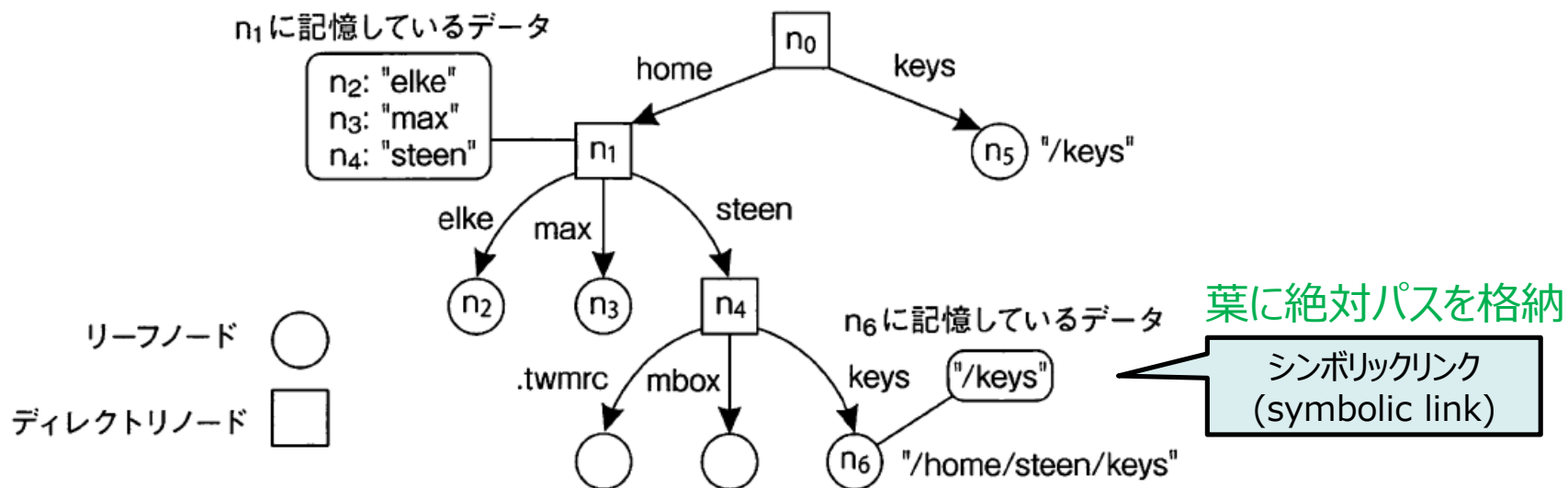
エンティティの階層通りに
名前を解決する

この例だと木構造
(有向非循環グラフ=DAG(directed acyclic graph))

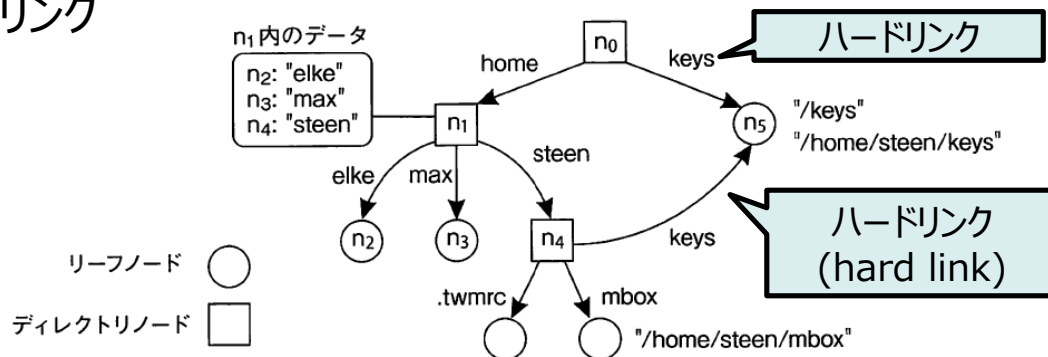
シンボリックリンクとハードリンク

■ シンボリックリンク

- エンティティには別名(alias)を付けることができる



ハードリンク



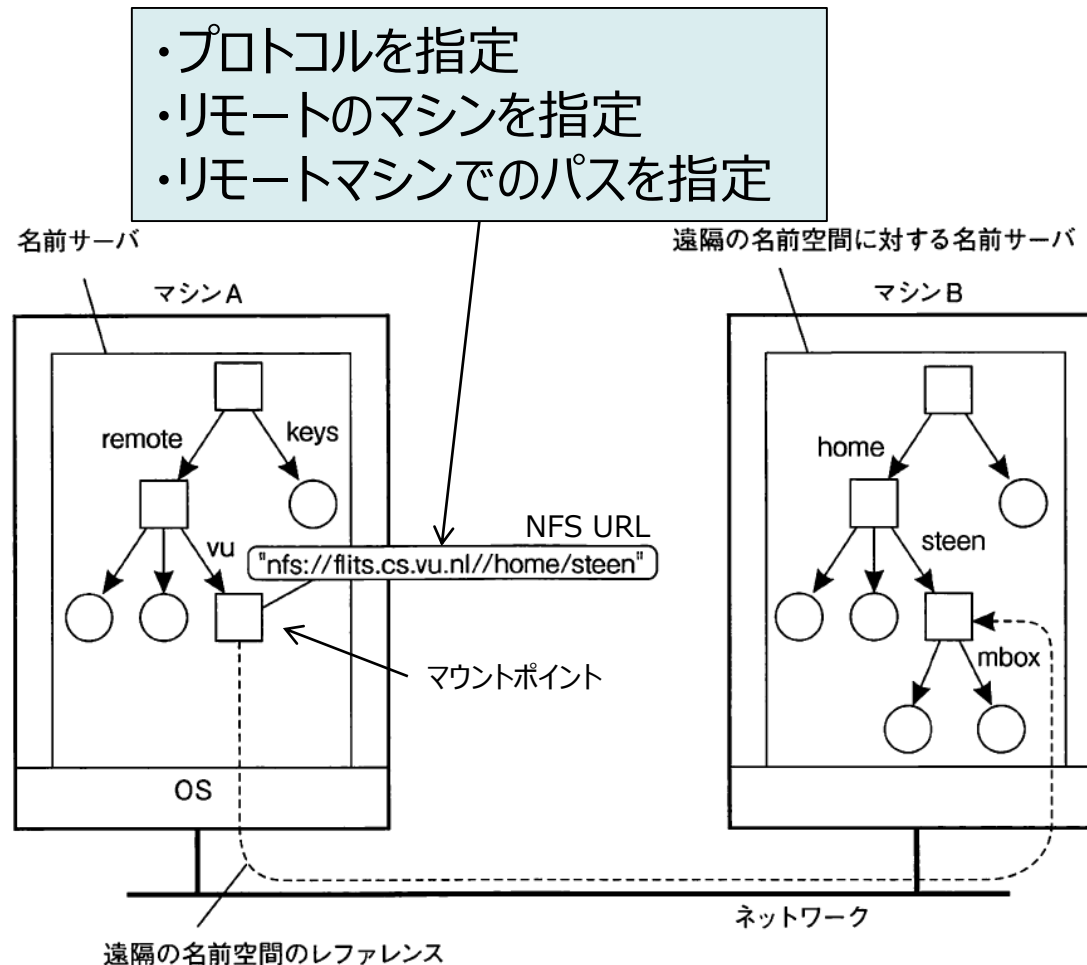
ハードリンクの場合、複数のリンクが同じ実体を指す。指しているリンクの数が0になった場合、そのデータは削除される。

マウント(mount)

- 複数の名前空間を併合することもできる
- マウント：OSからファイルシステムを
使えるようにするための手続き
 - 既存のファイル階層に、他のファイル階層の
一部を併合することも可能
 - 併合地点のことをマウントポイント(mount point)
と呼ぶ

マウント(mount)

(例) NFS (Network File System):
ネットワークの先にある他のマシンのファイルシステムをマウントできる



確認問題

- 下記は名前解決の手法に関する説明である。
どれがどの手法に関する説明かを語群から選んで答えよ。
- 同一ネットワーク内に存在するマシンすべてに
対して問い合わせる
- グループ内のマシンすべてに対して問い合わせる
- エンティティが移動するごとに、位置参照の
ためのポインタを生成
- エンティティの最新の現在位置を保持する
ホームを設定

語群： 転送ポインタ、ホームベースアプローチ、
ブロードキャスト、マルチキャスト



確認問題

- 以下の各文は正しいか。○か×で答えよ。
 - ファイル階層の最下部に新しくサブディレクトリやファイルを作成することをマウントと呼ぶ。
 - マウントにより、既存のファイル階層に、他のファイル階層の一部を併合することができる。
- NFS URLを構成する3つの要素を答えよ。



講義内容

■ 名前

- 名前・アドレス・識別子

- 名前付けと名前解決

 - フラットな名前付け

 - 構造化された名前付け

- ➡ ■ 名前空間の実装

名前空間の実装

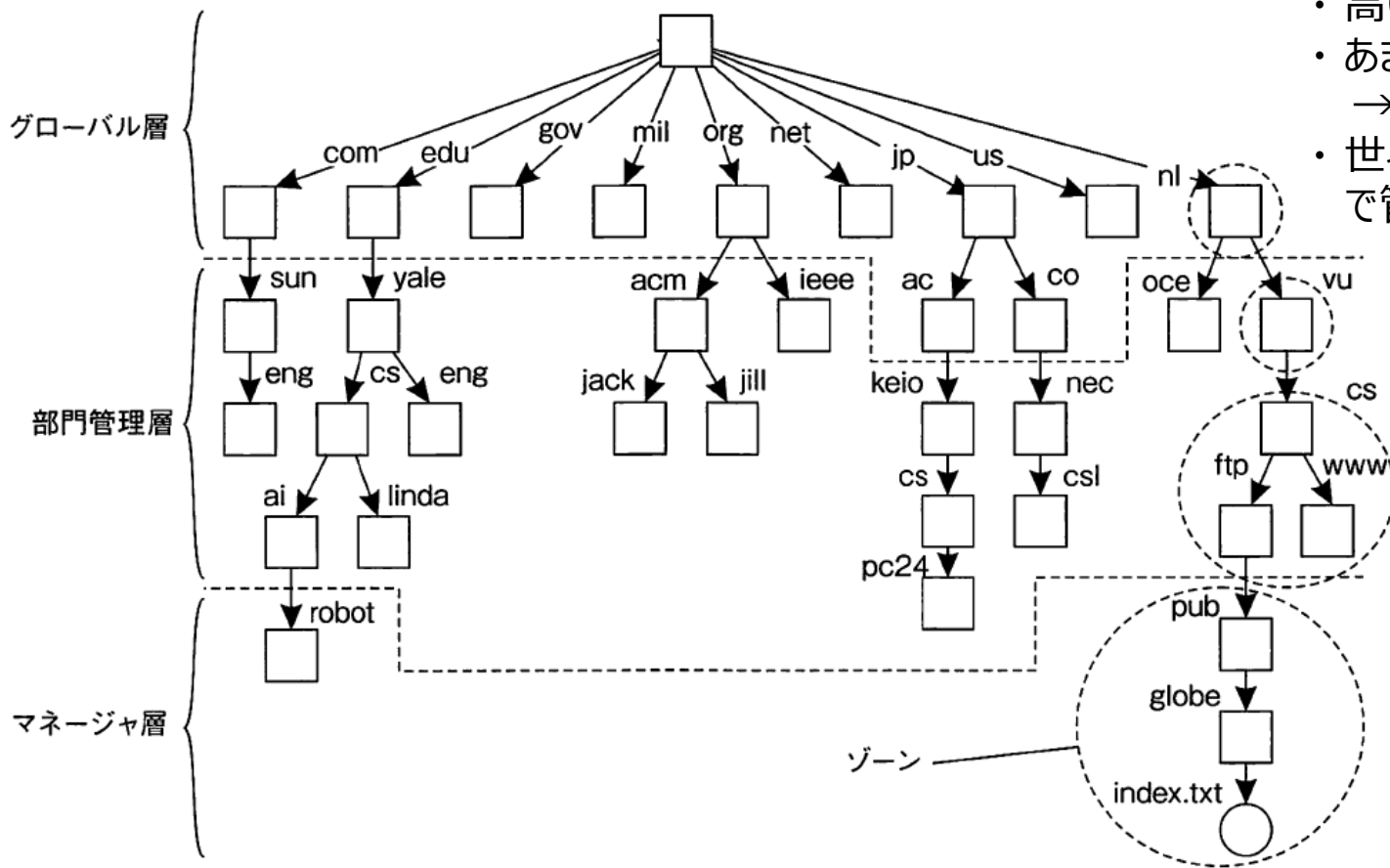
- 名前サーバ(name server)：名前付けサービスを提供
 - 分散システムがLANに限定される場合、1つの名前サーバで対応可能
 - 地理的に分散し、複数の組織にまたがる分散システムでは、複数の名前サーバで分散管理
- 名前管理分散の論理的3層
 - グローバル層：
 - ルートノードやその近傍。内容がほぼ不変
 - 部門管理層：
 - 1つの組織の中で管理されるディレクトリノード
 - マネージャ層：
 - 1つのLANに属するホストやファイル。内容変更が頻繁

名前空間の実装

nslookupコマンドの
実行結果例

■ (例) DNS (domain name system)

```
名前:      www.ritsumeai.ac.jp
Address:   133.19.170.14
```

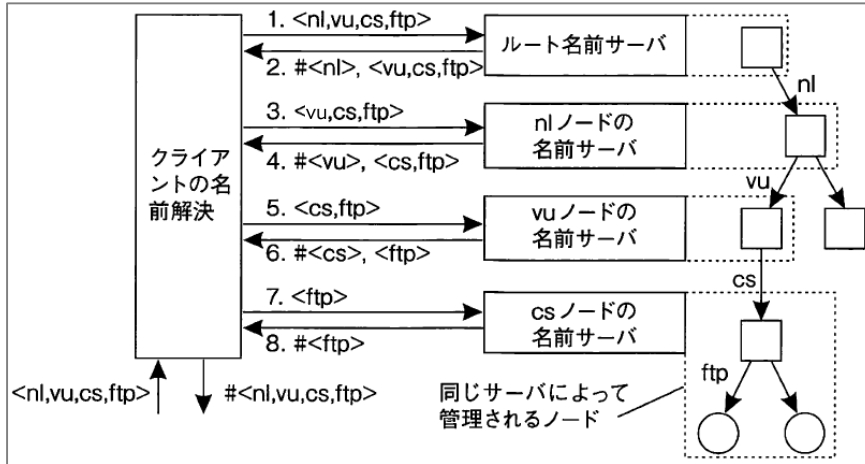


- ・ 高い可用性が求められる
- ・ あまり変化しない
→ キャッシュ(cache)が有効
- ・ 世界・国レベル(ルートゾーン)で管理

・ 組織レベルで管理

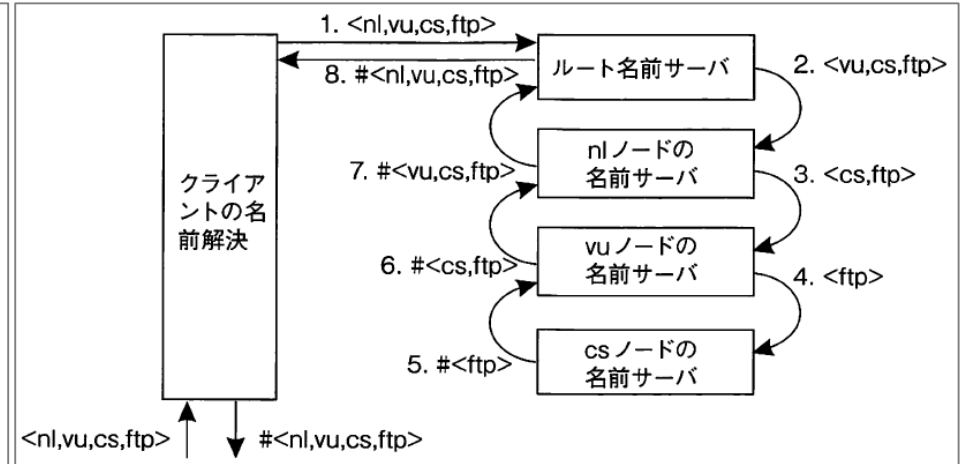
- ・ 即座に応答することが求められる
- ・ 部課レベルで管理

名前解決の仕組み



反復名前解決 (iterative name resolution)

- ・ ルートサーバから順に名前解決
- ・ クライアントと名前サーバが離れている場合は不利



再帰名前解決 (recursive name resolution)

- ・ 名前サーバが下位サーバに再帰的に問い合わせる名前解決
- ・ 高性能な名前サーバが必要
- ・ 結果のキャッシュが有効に働く

属性ベース名前付け (attribute-based naming)

- (属性, 値) の組で実体を記述
 - ディレクトリサービスとも呼ばれる
 - (例) 電子メールには宛先、差出人、件名などの属性があり、検索の絞り込みに使える
- 適切な属性とその値を設計するのは困難な場合がある
- 属性値の統一的な管理のための取り組み
 - RDF (Resource Description Framework)

RDF

■ 統一的なリソース記述のためのフレームワーク

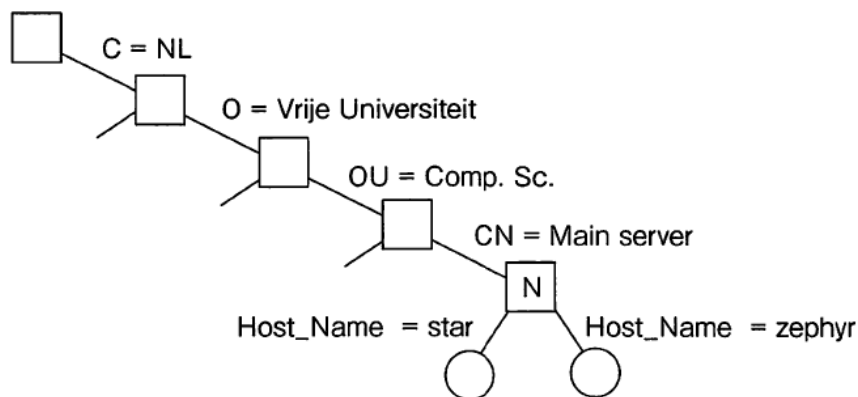
- 主語、目的語、述語の組でリソース間の関連をXML形式で記述
- 主語、述語、目的語自身がリソースになることもできる
- 述語をプロパティ、目的語をその値と呼ぶこともある
- リソースはURI(Uniform Resource Identifier)で参照
 - URI: リソースを指し示すための統一的書式で書かれた識別子
 - URL, URN (Uniform Resource Name)を含む



LDAP (Lightweight Directory Access Protocol)

■ 分散されたディレクトリサービスを提供するためのプロトコル

■ 階層化された名前付け 兼 属性ベース名前付け



属性	値
C (Country):	NL (Netherland)
O (Organization) :	Vrije University
...	

←同じ属性(Host_Name)だが、
どちらの値も探索可能

/C=NL/O=Vrije Universiteit/OU=Comp.Sc./CN=Main Server

… ノードNを指定

answer=search("&(C=NL)(O=Vrije Universiteit)(OU=*)(CN=Main Server)")

… Vrije Universiteitにおける主要サーバを取得

確認問題

- DNSを構成する名前サーバをグローバル層、部門管理層、マネージャ層の3つに分類して考えるとき、以下の各文はどれの説明に該当するかを答えよ。
 - 速やかに応答することが求められる。
 - 高い可用性が求められる。
 - 内容があまり変更されない。
 - 組織単位で管理される。
 - 部課単位で管理される。
- 反復名前解決が再帰名前解決と比較して優れている点の説明として適切なものはどれか答えよ。
 - クライアントと名前サーバが離れている場合にも有効である。
 - サーバの性能が比較的低くても良い。
 - サーバにおける問い合わせ結果のキャッシュが有効に働く。
 - クライアントはサーバに一度問い合わせるだけでよい。



確認問題

- RDFに関する説明として正しいものをすべて答えよ。
 - マークアップ言語により記述される。
 - リソースはURIで参照される。
 - 主語、述語、目的語の組でリソース間の関連を記述する。
 - 主語、述語、目的語自身がリソースになることもできる。
- LDAPの説明として誤っているものはどれか。
 - 属性と値の組み合わせを使って階層構造を表現する。
 - ディレクトリサービスに接続するために使用される。
 - 1つのノードは同じ名前の属性を2つ以上持たない。



参考文献

- 「分散システム」
水野 忠則 監修、共立出版、2015
- 「分散システム 原理とパラダイム 第2版」
アンドリュー・S・タネンバウム 他 著、
ピアソン・エデュケーション、2009
- 「セマンティック・ウェブのためのRDF/OWL入門」
神崎 正英 著、森北出版、2005