LDAP : Une présentation très rapide

Table des matières

1 Présentation de LDAP

- LDAP : **Lightweight Directory Access Protocol**.
- C'est un protocole d'accès à un annuaire.
- Il n'y a donc pas de contrainte d'implantation.
- Un annuaire propriétaire peut (et doit) fournir une interface LDAP (comme ODBC pour les S.G.B.D.R.).
- Il existe des annuaires LDAP natifs (`openLDAP`).

2 Caractéristiques de LDAP

- Accès rapide mais mises à jour plus lentes
- Structure arborescente
- Langage de recherche
- l'échange de données se fait par le format `LDIF`

```
Annuaire                Fichier                Annuaire
 LDAP A --- Requête ---> LDIF  --- Ajout ---> LDAP B
```

- Il existe des opérations de duplication et de synchronisation
- Il est donc facile de maintenir des copies

3 Utilisation de LDAP

- Annuaire d'entreprise,
- Annuaire *Active Directory* (**Windows 2000**).
- Représentation des paramètres du Système d'Information (J2EE).
- Annuaire de l'université (`http://annuaire.univ-amu.fr`).

4 Structure de LDAP

Un annuaire LDAP est un arbre (DIT pour **Directory Information Tree**). Chaque noeud corresponds à une série d'affectations d'attributs.

- On associe une ou plusieurs valeurs à chaque attribut.
- Les valeurs sont des chaînes de caractères ou des données binaires.

Une organisation `my-domain.com`

```
dn: dc=my-domain,dc=com              <-- distinguished name (l'ID)
objectclass: dcObject                <-- le type du noeud
objectclass: organization            <-- le type du noeud
o: Ma première organisation          <-- un attribut et sa valeur
dc: my-domain                        <-- idem
```

Une personne `massat.my-domain.com`

```
dn: cn=massat,dc=my-domain,dc=com    <-- distinguished name (l'ID)
objectclass: person                  <-- le type du noeud
cn: massat                           <-- un attribut et sa valeur
sn: Jean-Luc Massat                  <-- idem
description: Enseignant au DIL       <-- idem
```

Le `dn` (**distinguished name**) est l'identifiant d'une entrée LDAP. C'est un chemin dans l'arborescence de l'annuaire.

5 L'attribut particulier `objectclass`

- L'attribut `objectclass` désigne la ou les classes associées au noeud.
- Une **classe** définie les attributs obligatoires et optionnels d'un noeud.
- Les attributs et les classes sont définis dans des **schémas** LDAP.
- Dans un serveur LDAP il existe un jeu de schémas de base qui définissent des classes et des attributs.

6 Les attributs classiques

- `cn` Le *common name* ou nom commum.
- `gn` *given name* c'est à dire le prénom.
- `sn` *surname*.
- `l` Le *locality name*.
- `st` *state or province name*.

- `ou` *organisational unit*.
- `dc` *domain component*.
- `o` *organization name*.

7 Le langage de requête

Tester un attribut

```
(attribut=valeur)          égalité
(attribut~=valeur)         approximation
(attribut!=valeur)         différence
(attribut>=valeur)         supérieur
(attribut<=valeur)         inférieur
```

Avec un joker

```
(cn=user1*)                cn débute par user1
(cn=*)                     cn possède une valeur
```

Combiner des conditions

```
(|(cond1)(cond2))          condition 1 ou condition 2
(&(cond1)(cond2))          condition 1 ou condition 2
(!(cond1))                 négation de la condition 1
```

Un exemple réaliste

```
(|(sn=A*)(&(cn=user1)(dc=dil*)))
```

8 Création d'attributs et de classes

Si les classes et les attributs prédéfinis ne suffissent pas, il est possible de :

- définir de nouveaux attributs,
  - ▷ à partir de rien,
  - ▷ à partir d'autres attributs (héritage).
- définir de nouvelles classes
  - ▷ à partir de rien,
  - ▷ à partir de classes existantes (héritage).

8.1 Définition d'un attribut

Définition d'un attribut dans un schéma LDAP :

Définir `name` et ses algorithmes de comparaison

```
attributetype ( 2.5.4.41 NAME 'name'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

Définir sn et surname qui héritent de name

```
attributetype ( 2.5.4.4 NAME ( 'sn' 'surname' )
   DESC 'RFC2256: last (family) name(s) for which the entity is known by'
   SUP name )
```

c/countryName avec héritage et une seule valeur

```
attributetype ( 2.5.4.6 NAME ( 'c' 'countryName' )
   DESC 'RFC2256: ISO-3166 country 2-letter code'
   SUP name SINGLE-VALUE )
```

8.2 Définition d'une classe

Définition d'une classe dans un schéma LDAP :

```
objectclass ( 2.5.6.6 NAME 'person'
   DESC 'RFC2256: a person'
   SUP top STRUCTURAL
   MUST ( sn $ cn )
   MAY ( userPassword $ telephoneNumber $
      seeAlso $ description ) )

objectclass ( 2.5.6.2 NAME 'country'
   DESC 'RFC2256: a country'
   SUP top STRUCTURAL
   MUST c
   MAY ( searchGuide $ description ) )

objectclass ( 1.3.6.1.4.1.1466.344 NAME 'dcObject'
   DESC 'RFC2247: domain component object'
   SUP top AUXILIARY MUST dc )
```

Un exemple plus réaliste :

```
objectclass ( 2.5.6.4 NAME 'organization'
   DESC 'RFC2256: an organization'
   SUP top STRUCTURAL
   MUST o
   MAY ( userPassword $ searchGuide $ seeAlso $
      businessCategory $ x121Address $
      registeredAddress $ destinationIndicator $
      preferredDeliveryMethod $ telexNumber $
      teletexTerminalIdentifier $
      telephoneNumber $ internationaliSDNNumber $
      facsimileTelephoneNumber $ street $ postOfficeBox $
      postalCode $
      postalAddress $ physicalDeliveryOfficeName $ st $ l $
      description ) )
```