TP N°1 LDAP

Master 2 I2L

Année 2009/2010

D'après la version de Jean-Christophe Soulié (2007-2008)

D. Duvivier
LIL – Université du Littoral Côte d'Opale
duvivier@lil.univ-littoral.fr

1 Installation

Comme pour tout logiciel, il est possible d'installer OpenLDAP par le biais de paquets binaires fournis par une distribution, ou bien en compilant les sources.

Les outils clients sont souvent dissociés des outils serveur et fournis dans des paquets séparés. Nous allons donc installer deux paquets. Ceci n'est pas forcément le cas dans un environnement de production où les clients agiront à distance depuis une autre machine.

Sur une distribution Debian, les paquets à installer sont les suivants : `slapd` et `ldap-utils`.

```
# aptitude install slapd ldap-utils
```

ou

```
# apt-get install slapd ldap-utils
```

Une fenêtre apparaît alors et vous demande le mot de passe associé à l'annuaire que vous mettez en place. **Indiquez « toto » comme pour les tests « habituels » en TP**.

Il existe de nombreuses documentations OpenLDAP sur Internet, notamment :

"OpenLDAP Administrator's Guide" http://www.OpenLDAP.org/doc/admin/ ; il existe d'autres documentations/formats accessibles via l'URL http://www.openldap.org/doc.

2 Configuration du serveur

L'intégralité de la configuration du serveur OpenLDAP (le démon `slapd`) s'effectue en modifiant le fichier `slapd.conf`, situé dans le répertoire `/etc/ldap`. Vous pouvez également jeter un oeil au fichier `/etc/default/slapd`.

Dans le fichier `/etc/default/slapd`, si vous avez correctement renseigné le fichier de configuration de notre DNS vu en TP SRS/Réseaux (`/etc/bind/db.m2i2l.org`), vous pouvez modifier la configuration pour obtenir une ligne du genre « `SLAPD_SERVICES="`ldap://ldap.m2i2l.org:389/ `..."` ».

Après cette modification, il est possible de taper des URL débutant par « ldap://ldap.m2i2l.org » dans votre navigateur. Attention, il faut que le navigateur soit configuré pour accepter les URL commençant par « ldap:// » (même Konqueror le fait :-)).

Voici un exemple de configuration `/etc/ldap/slapd.conf`, ainsi que l'explication des directives (man 5 slapd.conf fournit des informations complémentaires) :

```
#######################################################################
# Directives globales
# Inclusion des schemas
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema

# Ou sera stocke le PID du demon
pidfile         /var/run/slapd/slapd.pid
# Liste des arguments passes au demarrage du serveur
argsfile        /var/run/slapd/slapd.args
# Niveau de log
loglevel        none
# Emplacement des modules
# Chargement du module BDB (Berkeley DB)
modulepath      /usr/lib/ldap
moduleload      back_hdb

# Limitation de la taille des informations retournees
```

```
# (nombre maxi d'entrees retournees par operation)
sizelimit 500

# %CPU consacré a l'indexation (en nombre de threads)
tool-threads 1

#######################################################################
# Declaration des options pour le premier type de backend utilise :
# bdb
# Toutes les options s'y appliquent jusqu'a la prochaine directive
# backend (peut etre bdb, hdb)
backend         hdb
#######################################################################
# Par exemple :
#backend <autre>
#######################################################################
# Declaration des options de la premiere "base", c'est a dire de la
# premiere (et unique ici) arborescence geree par notre annuaire
# Toutes les options s'y appliquent jusqu'a la prochaine directive
# database
database        hdb
# La racine de notre arborescence
# ATTENTION : surtout ne pas utiliser "dc=univ-littoral,dc=fr"
# car ceci est utilisé par le serveur LDAP de l'université
# Remplacez par "dc=m2i2l<N°IP>,dc=fr" où <N°IP> est le dernier
# champ de votre adresse IP (pour éviter les redondances)
# Par exemple :
suffix          "dc=m2i2l250,dc=fr"

# Le compte administrateur de notre arborescence et son mot de passe
rootdn          "cn=admin,dc=m2i2l250,dc=fr"
# Il est possible d'écrire le mot de passe “en clair” :
# rootpw        "toto"
# mais il est préférable de le crypter à partir de la commande
# slappasswd :
rootpw          "{CRYPT}J9NQ/M7KbU9Ak"

# Ou sont stockes les fichiers BDBs de notre arborescence
directory       "/var/lib/ldap"

# Taille du cache maxi est fixée à Mo = 2097152 octets par défaut
# N'oubliez pas d'ajuster cette valeur selon la RAM dispo
# et la taille du/des bases gérées par votre serveur !
# Ici je la place à 16Mo =
dbconfig set_cachesize 0 16777216 0

# Numbre de « verrous » (cf man)
dbconfig set_lk_max_objects 1500
dbconfig set_lk_max_locks 1500
dbconfig set_lk_max_lockers 1500

# Options d'index (pour la base 1, i.e. La base courante)
index           objectClass eq
# Sauvegarde de l'heure a laquelle est modifiee une entree
lastmod         on

# Fréquence des « Checkpoint » de la base 1 (ici BerkeleyDB)
# Effectue une synchro de la base sur disque (flush/sync)
# et ajoute une entrée dans le journal (si autorisé/demandé)
# cf « man slapd-bdb » pour les paramètres
checkpoint      512 30

# ACLs de notre premiere arborescence (« base 1 ») :
# Une personne non authentifiee peut s'authentifier
# Une personne authentifiee peut modifier son propre mot de passe
```

```
# Les autres n'ont pas accès à l'attribut mot de passe
access to attrs=userPassword,shadowLastChange
          by dn=”cn=admin,dc=m2i2l250,dc=fr"
          by anonymous auth
          by self write
          by * none
# Pour certaines fonctionnalités (SASL) il faut ajouter cette ligne
access to dn.base="" by * read
# Tout le monde peut lire l'annuaire
access to *
          by dn=”cn=admin,dc=m2i2l250,dc=fr" write
          by * read

# Ci-dessus, en ajoutant - dans chaque bloc “access to”
# la ligne suivante, nous donnons des accès complets à “admin” :
#       by dn="cn=admin,dc=m2i2l250,dc=fr" write

#######################################################################
# Autre arborescence (base 2)
#database <autre>
#suffix "dc=debian,dc=org"
#[...]
```

Copiez la configuration par défaut sous un autre nom par sécurité :

```
cp /etc/ldap/slapd.conf /etc/ldap/slapd.conf.orig
```

Utilisez votre éditeur préféré (emacs ?), puis sauvegardez votre configuration. Testez-là ensuite pour voir si aucune erreur n'a été commise (en général, les erreurs sont situées au niveau des dc=...) :

```
# slaptest -f /etc/ldap/slapd.conf
config file testing succeeded
```

Enfin, (re)-démarrez le serveur LDAP :

```
# /etc/init.d/slapd restart
```

Vérifiez que le serveur est bien démarré à l'aide de la commande :

```
tail /var/log/syslog
```

Comparez les résultats des commandes suivantes :

```
pidof slapd
cat /var/run/slapd/slapd.pid
```

Au passage, vérifiez les arguments effectivement passés au démarrage du serveur :

```
cat /var/run/slapd/slapd.args
```

Le fichier de configuration est subdivisé en trois sections importantes :

- La section globale (début du fichier) ;
- La section concernant les options de `backends` (débute par « `backend` ») ;
- La section concernant les déclarations et les options des arborescences gérées (débute par « `database` »).

Nous allons évoquer les directives les plus importantes ; n'hésitez pas à vous reporter au manuel du fichier de configuration pour plus d'informations :

```
# man 5 slapd.conf
```

2.1 L'inclusion des schémas

L'inclusion des schémas est effectuée par la directive « `include` ». Comme nous l'avons vu, les schémas LDAP permettent de définir les types de données contenus dans l'annuaire.

C'est grâce à ces inclusions au sein du fichier de configuration que l'on porte à la connaissance du serveur ces nouveaux types de données. Une fois les schémas chargés, il sera possible d'ajouter des entrées y faisant référence dans notre annuaire.

Plusieurs schémas sont fournis par défaut, je vous invite à regarder dans le répertoire `/etc/ldap/schema` pour les découvrir.

La directive include inclut en fait un fichier de configuration (de manière générale). Il est donc possible de disposer de plusieurs fichiers de configuration spécifiques et de les regrouper de cette manière.

2.2 Les niveaux de log

Il peut être important de savoir ce que fait exactement l'annuaire, ce, à des fins de débuggage par exemple. Pour ceci, nous avons la possibilité de changer le niveau de log dans le fichier `slapd.conf`.

Les niveaux de log disponibles sont les suivants (issus de « `man slapd.conf` ») :

Valeur	Fonction correspondante
1	Appels de fonctions
2	Gestion des paquets
4	Trace détaillée
8	Gestion des connexions
16	Affichage des paquets envoyés et reçus
32	Gestion des filtres de recherche
64	Gestion du fichier de configuration
128	Gestion des ACLs
256	Affichage des connexions, opérations et résultats
512	Affichage des entrées retournées
1024	Affichage des communications avec les backends
2048	Parsing des entrées
16384	LDAPSync replication
32768	Équivaut à « none », seul les messages de haute priorité sont loggués

Ces niveaux sont cumulables, c'est-à-dire qu'un niveau 48 équivaut aux niveaux 16 et 32. Un niveau 0 équivaut à un log désactivé.

Les logs sont gérés par `syslog`, ce qui signifie que vous pourrez consulter les informations logguées dans le fichier `/var/log/syslog`.

2.3 Les backends

Différents backends sont disponibles. Les plus couramment utilisés sont HDB/BDB (Berkeley DB : http://www.sleepycat.com --> http://www.oracle.com/database/berkeley-db/index.html) et LDBM. Ces deux backends sont des bases de données stockées dans des fichiers. HDB/BDB est recommandé car réputé plus robuste que LDBM.

Consultez les pages de man de `slapd-bdb` et `slapd-ldbm` pour plus d'informations. Vous y apprendrez notamment que « HDB » est une variante « hiérarchique » de BDB qui facilite la gestion des arborescences.

D'autres backends existent et permettent par exemple de stocker les informations dans de véritables SGBD, mais nous n'allons pas présenter ces fonctionnalités ici.

2.4 Les databases

Une section de database représente la déclaration d'une arborescence. Ceci implique plusieurs paramètres, dont une racine, un compte administrateur, …

Voici les paramètres importants dans cette section :

2.4.1 La racine

Elle est spécifiée par la directive « `suffix` ». La racine correspond souvent au FQDN (Fully Qualified Domain Name) de la machine associé aux attributs « `dc` ».

```
suffix          "dc=univ-littoral,dc=fr"
```

ou (pour les TP uniquement où nous n'utilisons volontairement pas le FQDN) :

```
suffix          "dc=m2i2l250,dc=fr"
```

2.4.2 L'accès administrateur

Il est possible de déclarer un compte qui ne sera sujet à aucune limitation. Il s'agit en quelques sortes d'un compte « `root` ». Ce compte peut correspondre ou non à une entrée dans l'annuaire. Il sera purement virtuel si aucun DN n'est effectivement stocké dans l'annuaire.

Ce compte particulier n'est pas soumis aux restrictions imposées par les ACLs (voir ci-dessous). Il est déclaré par la directive « `rootdn` ». Son mot de passe est spécifié par la directive « `rootpw` ».

Attention, ne confondez pas « `rootdn` » et DN racine, qui correspond à la base de notre annuaire ! Ici le « `rootdn` » est bien le `dn` d'un utilisateur ayant les droits root…

Le mot de passe du `rootdn` peut être soit en clair (« toto » dans notre exemple), soit crypté par la commande `slappasswd` :

```
# slappasswd
New password: <Mon_beau_mot_de_passe>
{SSHA}EOTSTuwotzohKCmXZ7Pqmm4YLA2p0
```

ou (pour les TP uniquement) :

```
# slappasswd -s "toto" -h {CRYPT}
{CRYPT}J9NQ/M7KbU9Ak
```

La valeur affichée est alors à copier-coller dans la valeur de la directive « `rootpw` » :

```
rootpw "{CRYPT}J9NQ/M7KbU9Ak"
```

Ceci permet de ne pas stocker en clair le mot de passe dans le fichier de configuration !

2.4.3 Les index

Les index sont un moyen d'accélérer les recherches au sein de l'annuaire. Dans le fichier de configuration, il convient de préciser quels attributs seront le plus fréquemment utilisés pour les recherches et doivent donc être indexés. Ceci se fait par la directive « index » :

```
index objectClass eq
```

Ici, nous activons la gestion des index sur les `objectClass`, ce qui semble un minimum !

Chaque index est destiné à faciliter un type de recherche. Le type d'index à créer est ici « `eq` », ce qui signifie qu'il sera efficace pour une recherche faisant intervenir une égalité stricte de chaînes.

Voici une liste des types d'index disponibles et leur type de recherche associé :

Index	Type de recherche (filtre), exemple
eq	`'uid=martymac'`, égalité stricte, pas d'utilisation de « wildcard » * (cf. `sub`)
sub	`'uid=marty*'`, utilisation d'un wildcard
subinitial	optimisation de `sub` pour `'uid=marty*'`, wildcard à la fin
subfinal	optimisation de `sub` pour `'uid=*mac'`, wildcard au début
subany	optimisation de `sub` pour `'uid=*rtym*'`, wilcard au début ou à la fin
approx	`'uid~=martymac'`, recherche par approximation phonétique
pres	`'objectclass=posixAccount'`, recherche de présence

Le type de recherche effectué est déduit du filtre passé au client qui effectue cette recherche.

Un ou plusieurs types de recherches peuvent être spécifiés pour un ou plusieurs attributs à la fois. Dans ce cas, la virgule sépare les différents éléments. Exemple :

```
index uid,gecos,description eq,subinitial
index uidNumber,gidNumber eq
```

Les index doivent être générés par l'administrateur pour être fonctionnels (commande « `slapindex` »), nous aborderons ce point par la suite.

2.4.4 Les listes d'accès (ACLs)

Les ACLs permettent de définir finement les droits d'accès à l'annuaire. La syntaxe générale des ACLs est la suivante :

```
access to <quoi> [ by <qui> <acces> [ <contrôle> ] ]+
```

Voici, par exemple, deux ACLs :

```
access to attrs=userPassword
          by anonymous auth
          by self write
          by * none
access to *
          by * read
```

La première concerne l'attribut `userPassword` :

- On autorise l'accès aux personnes non authentifiées uniquement pour une authentification (`by anonymous auth`) ;
- On autorise une personne authentifiée à modifier son propre mot de passe (`by self write`) ;
- Enfin, on refuse l'accès à cet attribut aux autres personnes.

La seconde ACL concerne toutes les informations contenues dans l'annuaire (`*`) :

- On autorise tout le monde à les lire

Les ACLs sont évaluées dans leur ordre d'apparition dans le fichier de configuration. OpenLDAP arrête leur évaluation lorsqu'il a trouvé une ACL faisant intervenir la cible recherchée. **Les directives les plus générales doivent donc être situées après les directives s'appliquant à une cible particulière**. C'est le cas ici avec l'ACL ciblant l'attribut `userPassword`, située avant celle ciblant toute information (`*`).

Nous n'allons pas étudier ici plus en détail les ACLs, je vous invite à consulter la page de man de « `slapd.access` » pour plus de détails.

3 Administration du serveur

3.1 Introduction

Notre serveur est désormais configuré. Voyons comment nous pouvons l'administrer.

Attention !!!

Les commandes que nous utilisons ici n'utilisent pas le protocole LDAP mais accèdent directement à la base de données sous-jacente (HDB/BDB dans notre cas). Il est donc impératif de toujours couper le serveur LDAP avant d'utiliser une commande slap(…), afin d'éviter un accès concurrent depuis le serveur lui-même, ce qui pourrait corrompre la base de données.

Coupez donc le serveur LDAP avant de continuer :

```
# /etc/init.d/slapd stop
```

3.2 Slapindex

Nous avons configuré notre serveur pour qu'il utilise des index ; la première chose à effectuer avant d'utiliser notre serveur est donc de les générer. Il faut en effet initialiser les index pour qu'OpenLDAP puisse ensuite les utiliser et les maintenir.

L'opération de génération n'est à effectuer qu'une seule fois et se fait via la commande `slapindex`.

```
# slapindex
```

3.3 Slapcat

`slapcat` est une commande très utile au quotidien. Elle effectue un « dump » de la base LDAP au format LDIF. Il est conseillé de l'utiliser régulièrement pour effectuer des sauvegardes de notre annuaire.

Par défaut, `slapcat` affiche les informations sur la sortie standard, il faut donc la rediriger vers un fichier pour obtenir notre sauvegarde :

```
# cd /etc/ldap
# slapcat > sauvegarde.ldif
# cat sauvegarde.ldif
```

3.4 Slapadd

`slapadd` est l'inverse de `slapcat`. Cette commande permet de peupler notre annuaire en utilisant un fichier LDIF. Elle est typiquement utilisée pour restaurer une sauvegarde effectuée avec `slapcat` :

```
# slapadd < sauvegarde.ldif
```

3.5 Arrêt et démarrage du serveur

L'arrêt et le démarrage du serveur LDAP se font par le biais du script `/etc/init.d/slapd` :

```
/etc/init.d/slapd [start|stop|restart|force-reload]
```

Dans un premier temps, on va plutôt le lancer dans une console pour voir ce qui se passe :

```
# cd /etc/ldap
# slapd -d 5 -g openldap -u openldap -f /etc/ldap/slapd.conf
```

- Le paramètre « `d` » permet au serveur d'afficher un certain nombre de données lorsqu'il reçoit des requêtes. Les valeurs classiques que vous pouvez utilisez au début sont 4 et 5. La valeur « 5 » vous donnera bien évidemment plus d'informations.
- Le paramètre « `f` » permet de spécifier le nom du fichier contenant la configuration du serveur.

3.6 Vérifications...

Vérifiez si tout est OK dans le fichier /etc/services :

```
# grep -i ldap /etc/services
ldap      389/tcp         # Lightweight Directory Access Protocol
ldap      389/udp
ldaps     636/tcp         # LDAP over SSL
ldaps     636/udp
```

Pour voir la liste des fichiers ouverts par le serveur, tapez l'une des commandes suivantes lorsque le serveur (`slapd`) tourne :

```
lsof -c slapd
lsof -i :389
nmap localhost -p389,636
```

4 Utilisation des outils clients

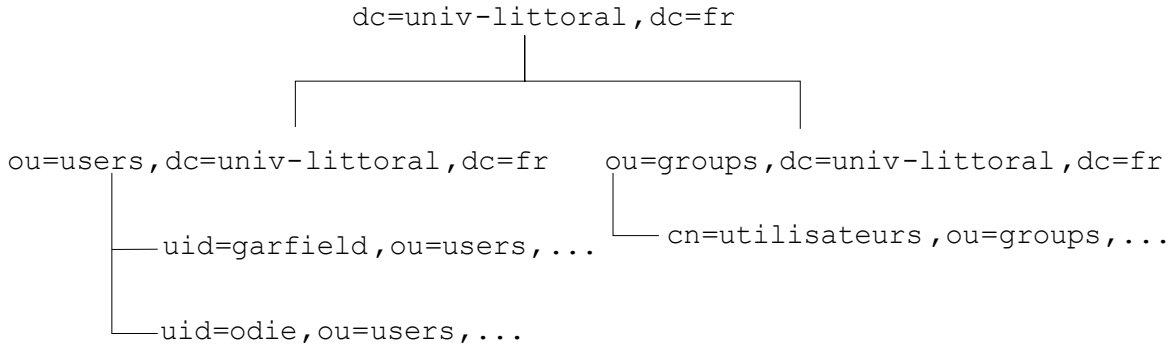
4.1 Introduction

Nous avons étudié les outils d'administration du serveur : les outils slap(…) ; nous étudions maintenant les outils clients.

A la différence des outils slap(…), les outils ldap(…) utilisent le protocole LDAP, ils peuvent donc être mis en œuvre depuis n'importe quelle machine disposant d'un accès réseau au serveur LDAP. Ils utilisent bien évidemment le format LDIF pour échanger des informations avec le serveur.

4.2 Création du schéma et ajout d'enregistrements

Dans un premier nous allons créer un schéma simple comme nous l'avons vu dans le cours :



ATTENTION : contrairement à l'exemple ci-dessus, repris du cours LDAP, surtout ne pas utiliser "dc=univ-littoral,dc=fr" car ceci est utilisé par le serveur LDAP de l'université. Remplacez par "dc=m2i2l<N°IP>,dc=fr" où <N°IP> est le dernier champ de votre adresse IP (pour éviter les redondances). Par exemple : dc=m2i2l250,dc=fr

L'initialisation de l'annuaire n'est qu'un ajout massif de plusieurs entrées. Cet ajout massif peut se faire par le biais de `slapadd` si vous possédez déjà un dump de l'annuaire et si vous vous situez sur le serveur.

A distance, c'est l'outil `ldapadd` qui permet d'effectuer cette opération. Il suffit de fournir à `ldapadd` un fichier LDIF contenant plusieurs entrées qui seront ajoutées dans le même ordre que celui dans lequel elles apparaissent dans le fichier. Voici un exemple de fichier « `schema.ldif` », si mon adresse IP se termine par "250" :

```
dn: dc=m2i2l250,dc=fr
objectClass: dcObject
objectClass: organization
dc: m2i2l250
o: m2i2l250
description: "Universite du Littoral"

dn: ou=users,dc=m2i2l250,dc=fr
objectClass: top
objectClass: organizationalUnit
ou: users

dn: ou=groups,dc=m2i2l250,dc=fr
objectClass: top
objectClass: organizationalUnit
ou: groups
```

```
dn: cn=utilisateurs,ou=groups,dc=m2i2l250,dc=fr
objectClass: posixGroup
cn: utilisateurs
gidNumber: 2000
```

REMARQUE : ATTENTION AUX ACCENTS !

Pour ajouter une entrée dans l'annuaire, utilisez la commande `ldapadd`. Sa syntaxe est la suivante :

```
ldapadd -W -D <binddn> -x -H ldap://<serveur> -f <fichier.ldif>
```

L'option « `-W` » active la demande de mot de passe pour s'authentifier en tant que `<binddn>`. L'option « `-x` » permet de ne pas utiliser SASL pour l'authentification. Enfin, le fichier LDIF source doit contenir une (ou plusieurs) entrée(s) à insérer et l'intégralité de ses (leurs) attributs.

Donc, on ajoute le tout à la base LDAP (en supposant que le fichier schema.ldif soit dans le répertoire ~/ldap, rappel : le mot de passe est « toto ») :

```
# cd ~/ldap
# ldapadd -W -D "cn=admin,dc=m2i2l250,dc=fr" -x -H ldap://localhost -f schema.ldif
Enter LDAP Password:

adding new entry "dc=m2i2l250,dc=fr"
adding new entry "ou=users,dc=m2i2l250,dc=fr"
adding new entry "ou=groups,dc=m2i2l250,dc=fr"
adding new entry "cn=utilisateurs,ou=groups,dc=m2i2l250,dc=fr"
```

Remarque : si vous avez bien configuré votre DNS lors du TP SRS/Réseaux, il est possible de remplacer la commande précédente par une commande du genre de celle ci (à adapter à VOTRE configuration) :

```
# ldapadd -W -D "cn=admin,dc=m2i2l,dc=org" -x -H ldap://ldap.m2i2l.org -f schema.ldif
```

Vérifions si les modifications ont été appliquées à la base :

```
# cd /etc/ldap
# /etc/init.d/slapd stop
# slapcat > sauvegarde2.ldif
# diff -bBd sauvegarde.ldif sauvegarde2.ldif
# /etc/init.d/slapd start
```

Au fait, où est ma base ? Ceci est défini par le champ « directory » du fichier de configuration /etc/ldap/slapd.conf --> dans /var/lib/ldap/ :

```
# file /var/lib/ldap/*

/var/lib/ldap/alock:           data
/var/lib/ldap/__db.001:        data
/var/lib/ldap/__db.002:        X11 SNF font data, LSB first
/var/lib/ldap/__db.003:        X11 SNF font data, LSB first
/var/lib/ldap/__db.004:        X11 SNF font data, LSB first
/var/lib/ldap/__db.005:        X11 SNF font data, LSB first
/var/lib/ldap/DB_CONFIG:       ASCII text
/var/lib/ldap/dn2id.bdb:       Berkeley DB (Btree, version 9, native byte-order)
/var/lib/ldap/id2entry.bdb:    Berkeley DB (Btree, version 9, native byte-order)
/var/lib/ldap/log.0000000001:  Berkeley DB (Log, version 8, native byte-order)
/var/lib/ldap/objectClass.bdb: Berkeley DB (Btree, version 9, native byte-order)
```

Une fois la base créée, on entre des utilisateurs grâce au fichier « `employes.ldif` » :

```
dn: uid=garfield,ou=users, dc=m2i2l250,dc=fr
objectClass: account
objectClass: posixAccount
cn: garfield
uid: garfield
uidNumber: 10001
gidNumber: 2000
homeDirectory: /home/garfield
userPassword:: e0NSWVBUfWRhSDJadHI4dElnZFE=
loginShell: /bin/sh
gecos: garfield
description: garfield

dn: uid=odie,ou=users, dc=m2i2l250,dc=fr
objectClass: account
objectClass: posixAccount
cn: odie
uid: odie
uidNumber: 10002
gidNumber: 2000
homeDirectory: /home/odie
userPassword:: e0NSWVBUfTQzZXltaHBSUzBqQVk=
loginShell: /bin/sh
gecos: odie
description: odie
```

Puis on ajoute le tout avec les commandes suivantes (en supposant que le fichier employes.ldif soit dans le répertoire ~/ldap, rappel : le mot de passe est « toto ») :

```
# cd ~/ldap
# ldapadd -W -D "cn=admin,dc=m2i2l250,dc=fr" -x -H ldap://localhost -f employes.ldif
Enter LDAP Password:
adding new entry "uid=garfield,ou=users, dc=m2i2l250,dc=fr"
adding new entry "uid=odie,ou=users, dc=m2i2l250,dc=fr"
```

« Normalement » ça doit fonctionner !

En cas de problème, n'oubliez pas que vous pouvez tester la conformité d'un DN donné en ligne de commande à l'aide de la commande slapdn :

```
slapdn -f /etc/ldap/slapd.conf -v 'dc=m2i2l250,dc=fr'
```

ou plus simplement :

```
slapdn -v 'dc=m2i2l250,dc=fr'
```

4.3 Rechercher une entrée : ldapsearch

La commande ldapsearch permet d'effectuer une recherche au sein de l'annuaire. Voici sa syntaxe :

```
ldapsearch -x -H ldap://<serveur> -b <base> [-s portée] [filtre] [attributs]
```

Elle reprend bien évidemment les concepts que nous avons abordés jusqu'ici :

- La base de la recherche ;
- La portée de la recherche (`base`, `one` ou `sub`) ; `sub` est la portée par défaut ;
- Le filtre ;
- Le ou les attributs que l'on souhaite afficher - l'entrée entière est affichée par défaut.

Il est possible de s'authentifier, si nécessaire, avec l'option `-D` (et l'option `-W`), mais notre annuaire est ici configuré pour permettre l'accès en lecture à tout le monde.

Exemples :

On recherche tous les `uid` commençant par « garf » à partir de la racine de l'annuaire :

```
ldapsearch -x -H ldap://localhost -b "dc=m2i2l250,dc=fr" '(uid=garf*)'
```

On recherche toutes les entrées ayant un `gidNumber` égal à 2000 :

```
ldapsearch -x -H ldap://localhost -b "dc=m2i2l250,dc=fr" "(gidNumber=2000)"
```

Cette commande nous retourne les 2 utilisateurs, mais aussi le groupe car il possède lui aussi l'attribut gidNumber. Améliorons notre requête pour ne retourner que les deux comptes utilisateurs :

```
ldapsearch -x -H ldap://localhost -b "dc=m2i2l250,dc=fr"
"(&(gidNumber=2000)(objectClass=posixAccount))"
```

Affichons enfin uniquement leur répertoire home (et pas la totalité de l'entrée comme c'est le cas par défaut) :

```
ldapsearch -x -H ldap://localhost -b "dc=m2i2l250,dc=fr"
"(&(gidNumber=2000)(objectClass=posixAccount))" homeDirectory
```

Le résultat de cette dernière requête est le suivant :

```
# ldapsearch -x -H ldap://localhost -b "dc=m2i2l250,dc=fr"
"(&(gidNumber=2000)(objectClass=posixAccount))" homeDirectory
# extended LDIF
#
# LDAPv3
# base <dc=m2i2l250,dc=fr
> with scope subtree
# filter: (&(gidNumber=2000)(objectClass=posixAccount))
# requesting: homeDirectory
#

# garfield, users, m2i2l250.fr
dn: uid=garfield,ou=users,dc=m2i2l250,dc=fr
homeDirectory: /home/garfield

# odie, users, m2i2l250.fr
dn: uid=odie,ou=users,dc=m2i2l250,dc=fr
homeDirectory: /home/odie

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

4.4 Supprimer une entrée : ldapdelete

Avant de supprimer des éléments, une petite sauvegarde s'impose :

```
# cd /etc/ldap
# /etc/init.d/slapd stop
# slapcat > sauvegarde3.ldif
# /etc/init.d/slapd start
```

La suppression d'une entrée se fait par la commande `ldapdelete`. Voici sa syntaxe :

```
ldapdelete -W -D <binddn> -x -H ldap://<serveur> <dn>
```

Puisqu'un effacement correspond à une écriture, il faudra, la plupart du temps, s'authentifier (à la différence de `ldapsearch`).

Il est possible d'effacer récursivement une branche complète en utilisant l'option « `-r` » sur le nœud de la branche. Attention, commande potentiellement dangereuse !

Il est possible de réinitialiser un annuaire par la méthode « **SBSR** » (**S**auvage et **B**rutale, mais **S**imple et **R**apide) ! En effet, il est possible de simplement supprimer les fichiers de la base de données de l'annuaire et de le redémarrer. Ces fichiers sont situés dans le répertoire `/var/lib/ldap` (cf. directive « `directory` » du fichier de configuration `/etc/ldap/slapd.conf`). Attention si vous possédez plusieurs bases à ne pas toutes les effacer… En cas de problème, il peut être nécessaire de filtrer « à la main » le dump de la base obtenu (via la commande `slapcat > sauvegarde3.ldif` par exemple). Par exemple, il s'agit de copier `sauvegarde3.ldif` sous un autre nom (par exemple `sauvegarde_complete.ldif`) puis d'enlever les entrées « inutiles ». Voici les modifications effectuées sur ma config (à adapter à votre config) :

```
# diff -bBd sauvegarde3.ldif sauvegarde_complete.ldif
1,28d0
< dn: dc=adsl,dc=proxad,dc=net
< objectClass: top
< objectClass: dcObject
< objectClass: organization
< o: adsl.proxad.net
< dc: adsl
< structuralObjectClass: organization
< entryUUID: 39e12ffe-9e13-102d-9d81-11752ba5be3e
< creatorsName:
< createTimestamp: 20090305205203Z
< entryCSN: 20090305205203.960368Z#000000#000#000000
< modifiersName:
< modifyTimestamp: 20090305205203Z
<
< dn: cn=admin,dc=adsl,dc=proxad,dc=net
< objectClass: simpleSecurityObject
< objectClass: organizationalRole
< cn: admin
< description: LDAP administrator
< userPassword:: e2NyeXB0fVFtay4vYm1zQmFScTY=
< structuralObjectClass: organizationalRole
< entryUUID: 39e717d4-9e13-102d-9d82-11752ba5be3e
< creatorsName:
< createTimestamp: 20090305205203Z
< entryCSN: 20090305205203.999261Z#000000#000#000000
< modifiersName:
< modifyTimestamp: 20090305205203Z
```

Lors de l'effacement (en supprimant les fichiers de la base de données de l'annuaire situés dans /var/lib/ldap) et de la regénération complète de la base via une sauvegarde (via `slapadd < sauvegarde3.ldif` par exemple), faites attention au groupe et propriétaire des fichiers regénérés dans /var/lib/ldap. Il doivent être initialisés à « openldap openldap » (sinon chown openldap:openldap /var/lib/ldap/* est votre ami). Si tel n'est pas le cas slapd ne peut pas verrouiller la base et le démon n'est pas lancé (vérifiez à l'aide d'une commande du genre tail /var/log/syslog).

Exemples :

Suppression de l'utilisateur odie :

```
ldapdelete -x -H ldap://localhost -W -D "cn=admin,dc=m2i2l250,dc=fr"
"uid=odie,ou=users,dc=m2i2l250,dc=fr"
```

Suppression de la branche users :

```
ldapdelete -x -H ldap://localhost -W -D "cn=admin,dc=m2i2l250,dc=fr" -r
"ou=users,dc=m2i2l250,dc=fr"
```

Si vous voulez tester plusieurs fois à partir de ce point, une petite sauvegarde s'impose :

```
# cd /etc/ldap
# /etc/init.d/slapd stop
# slapcat > sauvegarde4.ldif
# /etc/init.d/slapd start
```

Je vous laisse recréer cette branche (sans utiliser `slapadd < sauvegarde3.ldif`) ainsi que ses deux utilisateurs pour pouvoir poursuivre…

4.5 Modifier une entrée : ldapmodify

La commande `ldapmodify` est un peu le « couteau suisse » des annuaires LDAP ! Elle va permettre d'effectuer toutes sortes d'opérations, y compris l'ajout et la suppression d'entrées.

Sa syntaxe est la suivante :

```
ldapmodify -W -D <binddn> -x -H ldap://<serveur> -f <fichier.ldif>
```

`ldapmodify` peut se substituer à `ldapadd` et `ldapdelete`, mais vous allez voir que son utilisation n'est pas des plus simples ! En effet, tout passe par le fichier `ldif` indiqué en entrée et qui va décrire l'opération à effectuer…

Voici une liste (non exhaustive) d'opérations possibles :

- Ajouter d'une entrée ;
- Supprimer d'une entrée ;
- Ajouter un attribut ;
- Supprimer un attribut ;
- Modifier un attribut.

N'hésitez pas à consulter les pages de man de « `slapd` » et « `ldif` » pour une liste exhaustive des opérations que l'on peut effectuer (voire également `man -k slapd` et `man -k ldap`).

Ajouter une entrée

Ajoutons un utilisateur « `john` ».

Fichier LDIF (`ajoutjohn.ldif`) :

```
dn: uid=john,ou=users,dc=m2i2l250,dc=fr
changetype: add
objectClass: account
objectClass: posixAccount
cn: john
uid: john
uidNumber: 10003
gidNumber: 2000
homeDirectory: /home/john
userPassword:: e0NSWVBUfTg0QmNhL1BhL2tIUC4=
loginShell: /bin/sh
gecos: john
description: john
```

On remarque la présence d'un attribut « `changetype` » en plus de chacun des attributs de l'entrée que nous souhaitons ajouter.

Application de la modification :

```
# cd ~/ldap
# ldapmodify -W -D "cn=admin,dc=m2i2l250,dc=fr" -x -H ldap://localhost -f ajoutjohn.ldif
Enter LDAP Password:


adding new entry "uid=john,ou=users,dc=m2i2l250,dc=fr"
```

Supprimer une entrée

Le principe est le même que pour l'ajout d'une entrée. Cette fois, la valeur de « `changetype` » n'est plus « `add` » mais « `delete` ».

Fichier LDIF (`supprimejohn.ldif`) :

```
dn: uid=john,ou=users, dc=m2i2l250,dc=fr
changetype: delete
```

Application de la modification : cf. point précédent, il s'agit de la même commande !

Ajouter un attribut

L'ajout d'un attribut s'effectue par le `changetype` « `modify` » et par un nouvel attribut « `add` » qui précise quel attribut ajouter. Ici, nous allons ajouter une seconde description pour l'utilisateur « `garfield` ». L'attribut « description » devient donc multi-valué.

```
dn: uid=garfield,ou=users, dc=m2i2l250,dc=fr
changetype: modify
add: description
description: gros chat paresseux
```

Après la modification (via une commande du genre ldapmodify -W -D "cn=admin,dc=m2i2l250,dc=fr" -x -H ldap://localhost -f modifgarfield.ldif), l'utilisateur « `garfield` » possède les attributs suivants :

```
ldapsearch -x -H ldap://localhost -b "ou=users,dc=m2i2l250,dc=fr" "(uid=garfield)"
```

```
#[...]
# garfield, users, m2i2l250.fr
dn: uid=garfield,ou=users,dc=m2i2l250,dc=fr
objectClass: account
objectClass: posixAccount
cn: garfield
uid: garfield
uidNumber: 10001
gidNumber: 2000
homeDirectory: /home/garfield
loginShell: /bin/sh
gecos: garfield
description: garfield
description: gros chat paresseux
#[...]
```

Supprimer un attribut

La suppression d'attribut s'effectue via le `changetype` « `modify` » et l'utilisation d'un nouvel attribut : « `delete` ».

Supprimons la description, pas vraiment gentille pour Garfield, que nous venons d'ajouter :

```
dn: uid=garfield,ou=users,dc=m2i2l250,dc=fr
changetype: modify
delete: description
description: gros chat paresseux
```

Notez qu'il est possible de ne pas spécifier la valeur de la description à supprimer. Dans ce cas, toutes les descriptions seront supprimées.

Modifier un attribut

Pour modifier un attribut, le « `changetype` » à employer est, ici encore, « `modify` ». L'attribut supplémentaire à ajouter est l'attribut « `replace` » qui va préciser quel attribut remplacer. Enfin, nous spécifions la nouvelle valeur de l'attribut.

```
dn: uid=garfield,ou=users,dc=m2i2l250,dc=fr
changetype: modify
replace: description
description: ami fidele de john
```

Remarquez que la modification ci-dessus remplace toutes les descriptions par celle qui a été spécifiée. Il n'est pas possible de modifier uniquement l'une des valeurs d'un attribut multivalué.

Il faudra ruser pour effectuer cette dernière opération et le faire en deux temps : d'abord supprimer l'attribut désiré, ensuite ajouter le nouvel attribut. Imaginons que nous soyons dans le cas où garfield ait deux attributs :

```
ldapsearch -x -H ldap://localhost -b "ou=users,dc=m2i2l250,dc=fr" "(uid=garfield)"
```

```
#[...]
# garfield, users, m2i2l250.fr
dn: uid=garfield,ou=users,dc=m2i2l250,dc=fr
objectClass: account
objectClass: posixAccount
cn: garfield
uid: garfield
uidNumber: 10001
gidNumber: 2000
homeDirectory: /home/garfield
loginShell: /bin/sh
gecos: garfield
description: ami fidele de john
description: chat gourmand
#[...]
```

Si nous souhaitons remplacer uniquement « chat gourmand » par « chat paresseux », nous pouvons utiliser le fichier ci-dessous :

```
dn: uid=garfield,ou=users,dc=m2i2l250,dc=fr
changetype: modify
delete: description
description: chat gourmand

dn: uid=garfield,ou=users,dc=m2i2l250,dc=fr
changetype: modify
add: description
description: chat paresseux
```

Ce qui peut s'écrire de manière plus concise par l'utilisation du tiret « - » qui permet de chaîner les actions pour un même DN :

```
dn: uid=garfield,ou=users,dc=m2i2l250,dc=fr
changetype: modify
delete: description
description: chat gourmand
-
add: description
description: chat paresseux
```

4.6 Renommer une entrée : ldapmodrdn

L'outil `ldapmodrdn` permet de modifier le RDN (uniquement) d'une entrée. Il s'utilise de cette manière :

```
ldapadd -W -D <binddn> -x -H ldap://<serveur> <dn> <nouveau_rdn>
```

Exemple :

Pour renommer « garfield » en « pookie », nous pourrions saisir cette commande :

```
ldapmodrdn -W -D "cn=admin,dc=m2i2l250,dc=fr" -x
-H ldap://localhost "uid=garfield,ou=users,dc=m2i2l250,dc=fr"
"uid=pookie"
```

L'attribut « `uid: pookie` » sera ajouté automatiquement à l'entrée car il compose le nouveau RDN.

L'ancienne valeur de l'`uid` (« `uid: garfield` ») sera conservée.

4.7 Configuration des outils clients

Pour nous simplifier la tâche par la suite, sachez qu'il existe un fichier de configuration pour les outils clients ! Ce fichier contient les options que les commandes clientes doivent utiliser par défaut : l'adresse du serveur cible, le binddn, etc… Jusqu'ici, ces options étaient passées à chaque fois en lignes de commandes ; renseigner ce fichier de configuration évitera cette tâche répétitive et fastidieuse.

Il existe deux fichiers de configuration : `/etc/ldap/ldap.conf` et `~/.ldaprc`. Le premier est disponible pour tous les utilisateurs ; le second peut être défini par l'utilisateur et permet de surcharger les options spécifiées par le premier.

Voici un exemple de fichier `/etc/ldap/ldap.conf` :

```
# Configuration des outils clients (voir « man ldap.conf »)
# Racine
BASE dc=m2i2l250,dc=fr
# Nom et port de l'annuaire
URI ldap://localhost:389
```

Ainsi, une interrogation de l'annuaire devient simplement :

```
# ldapsearch -x
```

4.8 Jouons un peu avec les URLs...

Si votre DNS est bien configuré alors vous pourrez taper des URL indiquées ci-dessous en les adaptant à votre configuration, sinon il suffit de remplacer « ldap.m2i2l.org » par « localhost ». Comme expliqué ci-dessus, je suppose que votre navigateur accepte les URL débutant par « ldap://... ».

Pour chercher garfield :

ldap://ldap.m2i2l.org:389/dc=m2i2l,dc=org??sub?(uid=garf*)

Pour le même type de recherche que précédemment :

ldap://ldap.m2i2l.org:389/dc=m2i2l,dc=org??sub?(&(gidNumber=2000)(objectClass=posixAccount))