

HowTo ASIX LDAP

Lightweight Directory Access Protocol

Curs 2017 -2018

Descripció dels aprenentatges:	3
Documentació / Recursos	5
HowTo/exercicis	5
Recursos:	5
Containers:	5
Github	5
Conceptes generals de LDAP	6
Instal·lació i creació d'una BD (edt.org)	6
Consultes Idsapsearch	8
Configuració slapd.conf	9
Descripció general	9
Configuració de múltiples bd	12
Generar password de rootdn	13
Ordres client LDAP	14
Afegir i eliminar: ldapadd, ldapdelete	14
Modificar: ldapmodify, ldapmodrdn	15
Miscel·lània: ldapcompare, ldapwhoami, getent	16
Modificar el password: ldappasswd	16
Ordres LDAP de servidor: slapd	17
slaptest	17
slapcat	17
slapadd	17
slappasswd	17
slapindex	18
slapacl	19
slapauth	19
slapdn	20
slapschema	20
Organització de les pàgines man del servidor	20
Configuració de ACLs	21

Conceptes generals	21
Exemple-01:	22
Exemple-02:	22
Exemple-03:	22
Exemple-04:	23
Exemple-05 ok:	23
Exemple-05 ko(malament):	23
Exemple-06:	24
Exemple-07:	24
RDN i homePassword	25
Conclusió:	25
Synopsis ACLs	25
Ldap Documentation ACLs	25
Operation requeriments	26
Exemples de consulta de permisos	27
Configuració dinàmica del servidor	28
Instal·lació d'un servei (repàs general)	30
Configuració LDAP Client	31
Eines gràfiques: gq i phpldapadmin	33
Eina gràfica: gq	33
Eina gràfics: phpldapadmin	34
Schema: creació d'schema/objectes/atributs propis	36
Exemples de creació d'Schema	36
Exemple general (documentació)	36
Exemples futbolista	36
Cas-A: Derivat de inetorgperson	38
Cas-B: Structural standalone object	40
Cas-C: Objecte Auxiliary	42
Més enllà!	44
Apèndix: exemples usats	45
Base de dades	45
Dades DB	45

Descripció dels aprenentatges:

Bàsic

1. Instal·lar una base de dades LDAP (de cara a barraca!)
 - a. Esborrar la BD d'exemple tant la configuració com la base de dades.
 - b. Crear un fitxer de configuració slapd.conf al nostre gust per generar la base de dades edt.org (dc=edt,dc=org). Redefinir la propietat dels elements (de la configuració i de les dades) assignant ldap.ldap.
 - c. Injectar dades (populate) a baix nivell amb slapadd i amb el servidor amb ldapadd. Problema de permisos (propietari i grup amb les accions a baix nivell).
 - d. Consultar les dades a baix nivell amb slapcat. Les dades del motor de base de dades (el dimoni) i les de la base de dades dc=edt,dc=org.
2. Consulta / manipulació de les dades.
 - a. Consultes ldapsearch. Base search, scope, atributs, filtres i operadors lògics.
 - b. Afegir i eliminar dades amb ldapadd i ldapdelete.
 - c. Actualitzacions amb ldapmodify. Tractament de fitxers ldif. Modificació dels rdn amb fitxers ldif i ldapmodrdn.
 - d. Ordres client ldapwhoai, ldapcompare, ldapuri, etc.
3. Creació de múltiples bases de dades.
 - a. Crear múltiples bases de dades eliminant la configuració actual. generar les BD edt.org, m06.cat i exemple.com. Omplir amb dades (populate). Examinar la configuració i els directoris de dades.
 - b. Consulta de dades a baix nivell amb slapcat i a alt nivell amb ldapsearch.
 - c. Persistència de les dades de l'antiga base de dades.
 - d. Entendre la distinció entre directori de configuració i directoris de dades. Persistència de dades.
4. Ordres de servidor slapd.
 - a. Repàs a totes les ordres de servidor: slapadd, slapcat, slappasswd, etc.
 - b. Generar passwords amb slappasswd.
 - c. Definició i regeneració d'index. Examinar-ne els fitxers.
5. Tractament de ACLs.
 - a. Modificar els passwords dels usuaris amb ldappasswd.
 - b. Modificar passwords dels usuaris amb ldapmodify.
 - c. Definició de acls. Tractament de les acls i les seves regles de funcionament. Valor per defcete, combinació de regles globals i locals, clàusules what, by, acces.
 - d. Examinar els valors de les acls consultant la configuració de la base de dades amb slapcat. Consultar les acls de les entitats i els atributs amb ldapacl.
 - e. Modificar les acls directament modificant el motor de base de dades cn=config, amb ordres ldapmodify.
6. Configuració dinàmica del servidor.
 - a. Accés al motor de la base de dades: l'element cn=config.
 - b. Modificar dinàmicament els valors de rootdn i rootpw.

7. Miscel·lània.
 - a. Consulta dels logs amb journalctl.
 - b. Eina gràfica gq.
 - c. Estudi dels schema, objectes, atributs i regles.

Intermig

8. Disseny d'un Schema propi.
 - a. Definició d'atributs propis
 - b. Crear Objectes Sctructurals.
 - c. Crear Objectes Auxiliars.
 - d. Crear un Schema propi.
 - e. Implementar un Schema propi i fer el populate i l'explotació de dades
9. LDAP distribuït.
 - a. Ldap distribuït: arbres i subarbres en servidors diferents.
 - b. Creació de referrals bàsics.
 - c. Cració de referrals amb resolució.
10. Conexions segures amb TLS
 - a. Obtenció de certificats digitals.
 - b. Implementació de connexions LDAP segures amb TLS.

Avançat:

11. Underconstruction!!
 - a. Redundàcia: productors i consumidors.
 - b. Backups de les configuracions i les dades.
 - c. Overlays.
12. Implementació amb altres àmbits:
 - a. Apache LDAP
 - b. Samba LDAP

Documentació / Recursos

HowTo/exercicis

Documentació:

- [objectius-ldap](#)
- [activitats_asix_m06_uf1_nf1_2015-2016](#) (fitxers exemples: [dades](#) [config](#))
- [HowTo-ASIX-draft-presentacio-Docker.pdf](#)
- [M06-UF1NF1-Introduccio_a_LDAP](#)
- [HowTo-ASIX_LDAP_2016-2017](#)
- Documentació: [Mastering Openldap](#)
- [Fitxers d'exemples per crear BD LDAP](#)
- Documentació: [Howto-ASIX-2_LDAP](#)

Recursos:

- **web** de @edt ASIX M06-ASO, del mòdul. N'hi ha prou de posar a google [ASIX-M06](#).
- **github edtasixm06** amb tota l'estructura per generar imatges automatitzades docker, pas a pas els servidors (i altres) que hem anat fabricant. Posar al google [github edtasixm06](#).
- **dockerhub images**. Imatges de docker amb els exemples de servidors, clients, etc que hem fabricat al mòdul. Posar al google [dockerhub edtasixm06](#).

Containers:

Curs 2018-2019

- edtasixm06/ldapserver:18base
- edtasixm06/ldapserver:18Datacn
- edtasixm06/ldapserver:18acl
- edtasixm06/ldapserver:18schema
- edtasixm06/ldapserver:18group
- edtasixm06/ldaphost:base
- edtasixm06/portainer
- edtasixm06/documentation

Github

- <https://github.com/edtasixm06/ldapserver.git>

Conceptes generals de LDAP

Consultar el document de presentació pdf "[M06-UF1NF1-Introduccio a LDAP](#)".

Conceptes clau de configuració:

- El fitxer **slapd.conf** conté la configuració per generar la base de dades, però és un sistema deprecated.
- El directori **/etc/openldap/slapd.d** és on es desa la configuració en format d'estructura de directoris Idiff. Es transforma amb slaptest del fitxer de configuració al directori.
- El directori **/var/lib/ldap** és on es desa per defecte la informació, les dades de la base de dades. Si hi ha vàries bases de dades cal un directori diferent per a cada una.
- Es pot consultar (**slapcat**) i inserir dades massivament en offline (**slapadd**) amb el servei apagat.
- Per usar ordres LDAP client cal que el servei estigui engegat, i cal connectar al servidor **localhost** (a l'aula per defecte les ordres client consulten al servidor d'informàtica).

Concepte claus de funcionament:

- Cal assegurar-se que els permisos del directori **/etc/openldap/slapd.d** (i subarbre) i els de **/var/lib/ldap** (i subarbre) siguin propietat de l'usuari i grup **ldap.ldap**.
- Les dades es poden assignar a la base de dades amb ordres de servidor (**slapadd**) i amb ordres de client (**ldapadd**), que tenen característiques diferents.
- Sempre que es modifiqui la configuració del directori **/etc/openldap/slapd.d** o de les dades de **/var/lib/ldap** usant l'ordre **slapadd** **cal** tornar a assignar la propietat d'usuari i grup a **ldap.ldap**.
És a dir, si es realitzen accions de root sobre la configuració o les dades (amb el servei apagat) cal restaurar el propietari i grup a **ldap.ldap**.
- La càrrega de dades intensiva (inicial per exemple) es fa accedint directament al backend (amb el servei offline) usant l'ordre **slapadd**.
- Amb el servei engegat es poden afegir dades a la base de dades amb l'ordre LDAP client **ldapadd**. La comunicació és del client al servidor usant el protocol **LDAP**.
- Assegurar-se que les ordres client consulten al servidor LDAP apropiat, si no s'indica a les aules d'informàtica es consulta al servidor d'informàtica (gandhi). Es recomana usar l'opció **"-h localhost:389"** per forçar l'ordre a consultar el propi servidor.

Instal·lació i creació d'una BD (edt.org)

L'objectiu d'aquest apartat es instal·lar el servidor, eliminar la BD per defecte que incorpora i generar-ne una corresponent a **"edt.org"**, que s'identifica com a **"dc=edt,dc=org"**.

Carregar les dades en offline (servidor apagat) de la organització edt.org. Restaurar el propietari i grup a **ldap.ldap** per poder engegar el servei slapd. Un cop engegat el servei carregar les dades d'usuaris de edt.org i fer-ne consultes **ldapsearch**.

Descripció del procediment d'instal·lació:

Part-A

- Instal·lar els paquets de `openldap-clients` i `openldap-servers`
- Esborrar la base de dades d'exemple que porta incorporada (`myexample.com`) i el directori de dades.
- Afegir el fitxer `DB_CONFIG` (que tristament hem esborrat) al directori `/var/lib/ldap`. El trobareu en el paquet `openldap-servers`.
- Verificar el fitxer de dades slapd.conf i generar a partir d'aquest fitxer el directori de configuració `/etc/openldap/slapd.d`. Verificar el directori creat i observar l'estructura de dades de directoris `ldif` que s'ha generat.
nota podeu consultar el fitxer de configuració a l'apartat de configuració.
nota en usar l'ordre `slaptest` mostra warnings perquè verifica l'existència de la base de dades i les dades quan encara no existeixen. Un cop feta l'ordre es pot usar **`slaptest -u`** i si no mostra errors ni warnings és pot continuar.
- Fer un dump de la base de dades generada observant la base de dades {1}, el monitor {2} (si s'ha configurat) i el propi servei del dimoni `slapd` {0}.
- *En aquest punt s'ha generat la configuració i l'estructura per a una BD anomenada **`dc=edt,dc=org`**, pero no conté cap element, cap entrada, ni tan sols el node arrel. S'ha preparat tot, però encara no hi ha cap dada.*

Part-B

- Carregar les dades de la organització "`edt.org`". Es carrega el node arrel i tres ou (unitats d'organització), però no es carreguen dades d'usuaris (es farà posteriorment).
Aquestes dades es carregen amb el servei apagat usant `slapadd` que accedeix directament a baix nivell als fitxers del backend.
Observar el directori de dades `/var/lib/ldap`.
- Restaurar els permisos apropiats de propietari i grup a `ldap.ldap` del directori de configuració (`/etc/openldap/slapd.d`) i del directori de dades (`/var/lib/ldap`).
- Engregar el servei `slapd` i verificar que està en marxa.
- Consultar el contingut de la base de dades "`dc=edt,dc=org`" de l'organització "`edt.org`". Usar tant la comanda client `ldapsearch` com la comanda servidor `slapcat`.

Part-C

- Afegir dades a la base de dades amb l'ordre client `ldapadd`. Concretament s'afegiran usuaris a la "`ou=usuaris`".
- Verificar les dades llistant-les amb consultes client `ldasearch`, i també fer un volcat de les dades amb `slapcat`.

**** no copieu aquestes línies d'ordres amb copy+paste a la consola perquè la codificació de caràcters genera guins i espais incorrectes. Sembla que estigui bé però no ho està!**

Part-A

```
# yum -y install openldap-servers openldap-clients
```

```
# rm -rf /etc/openldap/slapd.d/*
```

```
# rm -rf /var/lib/ldap/*
```

```
# cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```



```

# slaptest -v -f slapd-edt.org.conf
# slaptest -v -f slapd-edt.org.conf -F /etc/openldap/slapd.d
# slaptest -v -f slapd-edt.org.conf -F /etc/openldap/slapd.d -u
# slaptest -v -F /etc/openldap/slapd.d -u

# slapcat
# slapcat -n1
# slapcat -n0

## Part-B
# slapdd -F /etc/openldap/slapd.d -l organitzacio_edt.org.ldif
# ls /var/lib/ldap

# chown -R ldap.ldap /etc/openldap/slapd.d
# chown -R ldap.ldap /var/lib/ldap

# systemctl start slapd
# systemctl status slapd

# ldapsearch -x -LLL -h localhost:389 -b 'dc=edt,dc=org'
# slapcat
# slapcat -n1 | less

## Part-C
# ldapadd -x -h localhost:389 -D 'cn=Manager,dc=edt,dc=org' -w secret
-f usuarios_edt.org.ldif

# ldapsearch -x -LLL -h localhost:389 -b 'dc=edt,dc=org'
# slapcat
# slapcat -n1 | lesss

```

Ara la base de dades ja conté un node arrel (edt.org), els nodes d'unitat d'organitzacio ou (maquines, clients, productes i usuaris) i un conjunt d'usuaris. Usant ordres client LDAP es poden consultar i actualitzar les dades de la BD.

Consultes ldapsearch

Atenció, cal indicar el servidor a consultar amb l'opció **-h localhost:389**.

Elements de les consultes:

- base de la consulta (en SQL seria la clausula FROM)
- indicar els camps a llistar (en SQL seria els camps del select) . Hi ha els operadors 1.1, * i +. Diferència entre atributs d'usuari i operacionals.
- establir condicions de filtrat (en SQL seria la clausula where)
- scope o àmbit de les consultes: base, one, sub, children
- operadors: and &, or |, not ! i els relacionals habituals.

Opcions generals de les consultes:

-x per indicar que el tràfic LDAP no es xifrat, és en text pla

-LLL per obtenir respostes 'planes', sense capçaleres ni comentaris, només dades de resposta.

-h host:port per indicar el host i el port al que es vol consultar. Atenció que cal indicar-ho perquè el client ldap per defecte a les aules d'informàtica està configurat per accedir al servidor d'informàtica (gandhi).

Exemple (zoom) de consulta amb filtres and i or:

' (& (| (cn=* Mas) (cn=* Pou)) (gidNumber=600)) '

```
# ldapsearch -x -h localhost:389 -b 'dc=edt,dc=org'
# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org'

# ldapsearch -x -h localhost:389 -LLL -b 'ou=productes,dc=edt,dc=org'
# ldapsearch -x -h localhost:389 -LLL -b 'ou=productes,dc=edt,dc=org' 1.1
# ldapsearch -x -h localhost:389 -LLL -b 'ou=productes,dc=edt,dc=org' *
# ldapsearch -x -h localhost:389 -LLL -b 'ou=productes,dc=edt,dc=org' +

# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' dn cn mail
# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' dn cn mail uid uidNumber

# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' 'cn=Pere Pou' *
# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' 'cn=Pere Pou' +
# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' 'cn=Pere Pou' dn cn mail
# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' 'cn=* Pou' dn cn gidNumber

# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' 'gidNumber=600' dn cn mail
uidNumber gidNumber
# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' '(&(|(cn=* Mas)(gidNumber=600))'
dn cn mail uidNumber gidNumber

# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' -s base
# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' -s one
# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' -s children
# ldapsearch -x -h localhost:389 -LLL -b 'dc=edt,dc=org' -s sub

# ldapsearch -x -h localhost:389 -LLL -b 'ou=usuaris,dc=edt,dc=org' -s base
# ldapsearch -x -h localhost:389 -LLL -b 'ou=usuaris,dc=edt,dc=org' -s one
# ldapsearch -x -h localhost:389 -LLL -b 'ou=usuaris,dc=edt,dc=org' -s children
# ldapsearch -x -h localhost:389 -LLL -b 'ou=usuaris,dc=edt,dc=org' -s sub
```

Configuració slapd.conf

Descripció general

Exemple de fitxer de configuració slapd.d que crea la base de dades corresponent a la organització "edt.org". En el protocol LDAP aquesta base de dades s'identifica com a "dc=edt,dc=org".

En aquest fitxer podem observar clarament les seccions:

- include on es carregen els schema de dades necessaris per la BD. per exemple podem observar que es carrega l'schema de *inetorgperson*.
- directives globals del servei slapd com per exemple el *bind* per indicar que usa el protocol ldapv2.
- definició de la base de dades de l'organització "edt.org".
- [opcional] definició d'índex
- [opcional] definició d'altres bases de dades
- [opcional] activació o no de la funció de monitoritzar el servei slapd.

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
include      /etc/openldap/schema/corba.schema
include      /etc/openldap/schema/core.schema
include      /etc/openldap/schema/cosine.schema
include      /etc/openldap/schema/duaconf.schema
include      /etc/openldap/schema/dyngroup.schema
include      /etc/openldap/schema/inetorgperson.schema
include      /etc/openldap/schema/java.schema
include      /etc/openldap/schema/misc.schema
include      /etc/openldap/schema/nis.schema
include      /etc/openldap/schema/openldap.schema
include      /etc/openldap/schema/ppolicy.schema
include      /etc/openldap/schema/collective.schema

# -- Global Directives
-----
# Allow LDAPv2 client connections. This is NOT the default.
allow bind_v2
pidfile      /var/run/openldap/slapd.pid
#argsfile    /var/run/openldap/slapd.args

# --Database dc=edt,dc=org -----
database bdb
suffix "dc=edt,dc=org"
rootdn "cn=Manager,dc=edt,dc=org"
rootpw secret
directory /var/lib/ldap

# Indices to maintain for this database
index objectClass          eq,pres

# ACLs for this database
access to *
    by self write
    by * read
# --end Database -----

# -- Database monitor -----
```

```
database monitor
# allow only rootdn to read the monitor
access to *
    by dn.exact="cn=Manager,dc=edt,dc=org" read
    by * none
# --end Database -----
```

La part clau del fitxer de configuració slapd.conf és la definició de les bases de dades a emmagatzemar, hi haurà una secció database per a cada una d'elles.

nota a vegades és fàcil confondre's i escriure les ACL al final del document fora del bloc de la base de dades on realment es pretenia configurar aquestes ACLs.

Els elements principals a definir són:

```
database bdb
suffix "dc=edt,dc=org"
rootdn "cn=Manager,dc=edt,dc=org"
rootpw secret
directory /var/lib/ldap
index objectClass eq,pres
```

- Es defineix una base de dades de tipus bdb (Berkeley database). Aquest es el backend on es desen les dades. LDAP no descriu com es desen les dades i es poden fer servir diversos backends diferents (postgres, el filesystem, etc).
- La directiva suffix defineix el nom de la base de dades. Si l'organització es "edt.org" el dn (Distinguished Name) és "dc=edt,dc=org".
- Cal definir un usuari administrador o root de la base de dades. Atenció, no és un usuari de dins de la base de dades es un usuari extern que es defineix aquí, al fitxer de configuració.

Es defineix en dos passos, la directiva que defineix l'usuari administrador de la base de dades s'anomena rootdn, i en aquest cas és un usuari (no real ni del LDAP) anomenat "cn=Manager,dc=edt,dc=org".

També cal assignar un password a aquest usuari, el password es defineix aquí amb la directiva rootpw i se li assigna "secret" com a password.

En aquest cas el passwd es en text pla però es poden generar passwords xifrats, consultar l'apartat pertinent.

En resum, es crea un usuari virtual (ni del sistema ni de la base de dades) amb drets totals d'administració d'aquesta base de dades.

- Les dades de la base de dades es desen al directori indicat per la directiva directory. Generalment es el directori /var/lib/ldap.
- Finalment cal indicar per quins elements es vol que el servei indexi les dades de la base de dades. Aquí es defineix un index per als objectClass.

Definició de ACLs:

```
access to *
    by self write
    by * read
```

Es poden definir múltiples ACLs per als elements de la base de dades fins i tot a nivell d'atributs. **Sempre** rootdn pot modificar (accedir i modificar) qualsevol contingut, per alguna cosa es root!.

El tractament de ACLs es similar als altres àmbits informàtics on es poden aplicar i com es habitual, pot acabar essent bastant complicat. En la forma més simple son fàcils d'entendre.

La ACL d'exemple indica que:

- qualsevol usuari pot modificar les seves pròpies dades.
- qualsevol usuari pot veure les dades dels altres.

Evidentment **no** és una ACL molt assenyada, però ens servirà per començar a remenar...

Reflexions sobre les dades i les ACLs:

- Un usuari pot modificar *totes* les seves dades? Puc passar de profe a Conseller d'Ensenyament amb un simple `ldapmodify`? Modificar la nòmina?
- Els altres usuaris quines dades poden observar? El password dels altres? L'adreça?
- Hi ha un camp que tot usuari ha de poder modificar per ell mateix, el seu propi password.

nota a vegades és fàcil confondre's i escriure les ACL al final del document fora del bloc de la base de dades on realment es pretenia configurar aquestes ACLs.

Configuració de múltiples bd

Aquest és un exemple on es configuren tres bases de dades diferents corresponents a les organitzacions "edt.org", "example.com" i "m06.cat". Cada base de dades es desa en un directori diferent dins de `/var/lib` utilitzant un nom tipus `/var/lib/ldap.edt.org`.

```
include      /etc/openldap/schema/corba.schema
include      /etc/openldap/schema/core.schema
include      /etc/openldap/schema/cosine.schema
include      /etc/openldap/schema/duaconf.schema
include      /etc/openldap/schema/dyngroup.schema
include      /etc/openldap/schema/inetorgperson.schema
include      /etc/openldap/schema/java.schema
include      /etc/openldap/schema/misc.schema
include      /etc/openldap/schema/nis.schema
include      /etc/openldap/schema/openldap.schema
include      /etc/openldap/schema/ppolicy.schema
include      /etc/openldap/schema/collective.schema
# Allow LDAPv2 client connections. This is NOT the default.
allow bind_v2
pidfile      /var/run/openldap/slapd.pid
#argsfile    /var/run/openldap/slapd.args

# --Database dc=example,dc=com -----
database bdb
```

```

suffix "dc=example,dc=com"
rootdn "cn=Manager,dc=example,dc=com"
rootpw secret
directory /var/lib/ldap.example.com
index objectClass eq,pres
# --end Database -----

# --Database dc=edt,dc=org -----
database bdb
suffix "dc=edt,dc=org"
rootdn "cn=Manager,dc=edt,dc=org"
rootpw jupiter
directory /var/lib/ldap.edt.org
index objectClass eq,pres
# --end Database -----

# --Database dc=m06,dc=cat -----
database bdb
suffix "dc=m06,dc=cat"
rootdn "cn=Manager,dc=m06,dc=cat"
rootpw jupiter
directory /var/lib/ldap.m06.cat
index objectClass eq,pres
# --end Database -----

# --Database Monitor -----
database monitor
# allow only rootdn to read the monitor
access to *
    by dn.exact="cn=Manager,dc=example,dc=com" read
    by dn.exact="cn=Manager,dc=edt,dc=org" read
    by dn.exact="cn=Manager,dc=m06,dc=cat" read
    by * none
# --end Database -----

```

Generar password de rootdn

Per generar passwords de servidor es pot usar l'ordre de servidor slappasswd que permet interactivament generar un password en el format de xifrat indicat. Per met xifrar en varis formats com per exemple MD6 i SHA1.

```

# slappasswd -h {md5}
New password:
Re-enter new password: jupiter
{MD5}J6UUjqD73a4i2QK+qaGVMQ==

# slappasswd -h {SHA}
New password:
Re-enter new password: jupiter

```

```
{SHA}ovf8ta/reYP/u2zj0afpHt8yE1A=
```

```
# slappasswd -h {CRYPT}
New password:
Re-enter new password: jupiter
{CRYPT}ZMcZ4/x5x0Klo
```

Un cop generat el password de rootdn amb el xifrat (o text pla) a usar, es copia a la directiva de configuració rootpw de la base de dades, per exemple:

```
rootdn "cn=Manager,dc=edt,dc=org"
rootpw {MD5}J6UUjqD73a4i2QK+qaGVMQ==
```

Ordres client LDAP

ldapsearch (tractada anteriorment)

ldapadd

ldapdelete

ldapmodify

ldapmodrdn

ldappasswd

ldapwhoami

ldapurl

ldapcompare

getent

Podeu consultar l'ajuda de les ordres client ldap consultant les seves pàgines man:

```
# man ldap
ldapadd      ldap.conf    ldapexop     ldapmodrdn   ldapsearch   ldapwhoami
ldapcompare  ldapdelete   ldapmodify   ldappasswd   ldapurl
```

Afegir i eliminar: ldapadd, ldapdelete

```
# afegir
# ldapadd -x -D cn=Manager,dc=edt,dc=org -w secret -f /tmp/ldap/usuaris1.ldiff

# esborrar
# ldapdelete -x -D cn=Manager,dc=edt,dc=org -w secret "cn=Pau
Pou,ou=usuaris,dc=edt,dc=org"

# oju!
# ldapdelete -x -r -D cn=Manager,dc=edt,dc=org -w secret "ou=usuaris,dc=edt,dc=org"
```

Modificar: ldapmodify, ldapmodrdn

```
# modificar
# ldapmodify -x -D cn=Manager,dc=edt,dc=org -w secret -f modificacions-ldif
```

Exemples de fitxer de modificacions tipus LDIF per a l'ordre *ldamodify*. Consultar **man (5)**

ldif: changetype: <[modify|add|delete|modrdn]>

```
dn: cn=Pau Pou,ou=usuaris,dc=edt,dc=org
changetype: modify
replace: mail
mail: modme@example.com
-
replace: homephone
homephone: 111-222-333
-
delete: description
-
add:description
description: nova descripció per a l'usuari Pau

dn: cn=Anna Pou,ou=usuaris,dc=edt,dc=org
changetype: delete

dn: cn=Anna Pou,ou=usuaris,dc=edt,dc=org
changetype: add
objectClass: posixAccount
objectClass: inetOrgPerson
cn: Anna Pou
cn: Anita Pou
sn: Pou
mail: anna@edt.org
ou: Alumnes
uid: anna
uidNumber: 5002
gidNumber: 600
homeDirectory: /tmp/home/anna
userPassword::
e1NTSEF9Qm00QjNCdS9mdUg2QmJ5OWxneGZGQXdMWXJLMFJiT3E=
description: modified description
homePhone: 93-222-333

dn: cn=Anna Pou,ou=usuaris,dc=edt,dc=org
changetype: modify
replace: homePhone
homePhone: 93-123-456

dn: cn=Anna Pou,ou=usuaris,dc=edt,dc=org
```



```
changetype: modify
delete: mail

dn: cn=Pau Pou,ou=usuaris,dc=edt,dc=org
changetype: modrdn
newrdn: cn=Pau Maria Pou
deleteoldrdn: 0
#newsuperior: ou=usuaris,dc=edt,dc=org
```

```
# modificar rdn
# ldapmodrdn -x -D cn=manager,dc=edt,dc=org -w secret 'cn=Anna
Puig,ou=usuaris,dc=edt,dc=org' 'cn=Annita Puig'
# ldapsearch -x -b dc=edt,dc=org '(cn=Annita Puig)'
```

Miscel·lània: ldapcompare, ldapwhoami, getent

```
# comparar
# ldapcompare -x "cn=Annita Puig,ou=usuaris,dc=edt,dc=org" mail:anna@edt.org
# ldapcompare -x "cn=Annita Puig,ou=usuaris,dc=edt,dc=org" mail:annita@edt.org

# identitat
# ldapwhoami -x
# ldapwhoami -x -D "cn=Manager,dc=edt,dc=org"
# ldapwhoami -x -D "cn=Manager,dchan=edt,dc=org" -w secret

# ldapwhoami -x -D cn=Manager,dc=edt,dc=org -W
Enter LDAP Password:
dn:cn=Manager,dc=edt,dc=org

# Si no s'indica l'opció "-x" intenta comunicació xifrada SASL (no usar...)
# ldapwhoami
SASL/DIGEST-MD5 authentication started
Please enter your password:
ldap_sasl_interactive_bind_s: Invalid credentials (49)
    additional info: SASL(-13): user not found: no secret in database

# ldapwhoami -x
anonymous

# verificar si l'usuari ldap forma part dels usuaris del sistema integrats als del /etc/passwd
# getent passwd | grep anna
# getent passwd
```

Modificar el password: ldappasswd

```
# modifica el passwd:q
# ldappasswd -v -x -D 'cn=Manger,dc=edt,dc=org' -w secret 'cn=Anna
Pou,ou=usuaris,dc=edt,dc=org'
```

```
# Idappasswd -v -x -D 'cn=Anna Pou,dc=edt,dc=org' -w secret 'cn=Anna Pou,ou=usuaris,dc=edt,dc=org'

# Idappasswd -v -x 'cn=Anna Pou,ou=usuaris,dc=edt,dc=org'
```

Ordres LDAP de servidor: slapd

El servidor slapd s'acompanya d'un conjunt d'ordres de baix nivell que actuen directament sobre la base de dades amb el servei apagat, es tracta de les ordres:

- slaptest
- slapadd
- slapcat
- slappasswd
- slapindex
- slapacl
- slapauth
- slapdn
- slapschema

slaptest

```
# slaptest -v -f /etc/openldap/slapd.conf #(test slapd.conf file)
# slaptest -v -F /etc/openldap/slapd.d #(test slapd.d directory)
# slaptest -v -F /etc/openldap/slapd.d.web #(test slapd.d-web anoder-directory)

# slaptest -v -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d -u
#(generate slapd.d tree and no check DB, doesn't exist)
# slaptest -v -F /etc/openldap/slapd.d -n0 -l file-slapd.ldif #(generate DB using .ldif)
```

slapcat

```
# slapcat -v -n0    #lists cn=config DB
# slapcat -n n°     #lists n° DataBase. Shuld contain entrys.
# slapcat           #lists default DB (usually the first suffix defined)
# slapcat -n1       #lists DB {1} usually the defaut
```

slapadd

```
# slapdd -F /etc/openldap/slapd.d -l organitzacio_edt.org.ldif
```

slappasswd

```
# slappasswd -h {md5}
New password:
```

```
Re-enter new password: jupiter
{MD5}J6UUjqD73a4i2QK+qaGVMQ==

# slappasswd -h {SHA}
New password:
Re-enter new password: jupiter
{SHA}ovf8ta/reYP/u2zj0afpHt8yE1A=

# slappasswd -h {CRYPT}
New password:
Re-enter new password: jupiter
{CRYPT}ZMcZ4/x5x0Klo
```

slapindex

Al directori de dades /var/lib/ldap hi ha les dades de la base de dades i també els índex que s'han generat. Els índex a realitzar es descriuen en la directiva índex de fitxer de configuració slapd.

```
database bdb
suffix "dc=edt,dc=org"
rootdn "cn=Manager,dc=edt,dc=org"
rootpw jupiter
directory /var/lib/ldap
index objectClass eq,pres
index cn,sn,mail eq,pres
access to * by self write by * read
```

Podeu consultar la documentació de la directiva *index* en la pàgina de manual **man (5) slapd-bdb**.

index {<attrlist>|default} [pres,eq,approx,sub,<special>]

Specify the indexes to maintain for the given attribute (or list of attributes). Some attributes only support a subset of indexes. If only an <attr> is given, the indices specified for default are maintained. Note that setting a default does not imply that all attributes will be indexed. Also, for best performance, an **eq** index should always be configured for the **objectClass** attribute.

A number of special index parameters may be specified. The index type **sub** can be decomposed into **subinitial**, **subany**, and **subfinal** indices. The special type **nolang** may be specified to disallow use of this index by language subtypes. The special type **nosubtypes** may be specified to disallow use of this index by named subtypes.

Note: changing index settings in slapd.conf(5) requires rebuilding indices, see slapindex(8); changing index settings dynamically by LDAPModifying "cn=config" automatically causes rebuilding of the indices online in a background task.

Si es llista el contingut del directori /var/lib/ldap es poden veure fitxers de índex. Aquests fitxers es poden regenerar (degut per exemple a que s'ha fet una càrrega massiva via slapadd o degut a que s'han corromput). Es poden regenerar tots o un índex en concret.

```
# ls /var/lib/ldap
alock cn.bdb __db.001 __db.002 __db.003 DB_CONFIG dn2id.bdb id2entry.bdb
log.00000000001 objectClass.bdb sn.bdb mail.bdb

# file /var/lib/ldap/*
/var/lib/ldap/alock:          data
/var/lib/ldap/cn.bdb:         Berkeley DB (Btree, version 9, native byte-order)
/var/lib/ldap/__db.001:       Applesoft BASIC program data
/var/lib/ldap/__db.002:       data
/var/lib/ldap/__db.003:       data
/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.00000000001: Berkeley DB (Log, version 18, native byte-order)
/var/lib/ldap/objectClass.bdb: Berkeley DB (Btree, version 9, native byte-order)
/var/lib/ldap/sn.bdb:         Berkeley DB (Btree, version 9, native byte-order)
/var/lib/ldap/mail.bdb:       Berkeley DB (Btree, version 9, native byte-order)

# slapindex -v
indexing id=000000001
indexing id=000000002

# slapindex mail
```

slapacl

extret del man slapacl

slapacl - Check access to a list of attributes.

slapacl is used to check the behavior of slapd(8) by verifying access to directory data according to the access control list directives defined in its configuration. It opens the slapd.conf(5) configuration file or the slapd-config(5) backend, reads in the access/olcAccess directives, and then parses the attr list given on the command-line; if none is given, access to the entry pseudo-attribute is tested.

slapauth

extret del man slapauth

slapauth - Check a list of string-represented IDs for LDAP authn/authz

Slapauth is used to check the behavior of the slapd in mapping identities for authentication and authorization purposes, as specified in slapd.conf(5). It opens the slapd.conf(5) configuration file or the slapd-config(5) backend, reads in the authz-policy/olcAuthzPolicy and authz-regexp/olcAuthzRegexp directives, and then parses the ID list given on the command-line.

slapdn

extret del man slapdn

slapdn - Check a list of string-represented LDAP DN's based on schema syntax

Slapdn is used to check the conformance of a DN based on the schema defined in slapd(8) and that loaded via slapd.conf(5). It opens the slapd.conf(5) configuration file or the slapd-config(5) backend, reads in the schema definitions, and then parses the DN list given on the command-line.

slapschema

extret del man slapschema

slapschema - SLAPD in-database schema checking utility

Slapschema is used to check schema compliance of the contents of a slapd(8) database. It opens the given data-base determined by the database number or suffix and checks the compliance of its contents with the corresponding schema. Errors are written to standard output or the specified file. Databases configured as subordinate of this one are also output, unless -g is specified.

Administrators may need to modify existing schema items, including adding new required attributes to object-Classes, removing existing required or allowed attributes from objectClasses, entirely removing objectClasses, or any other change that may result in making perfectly valid entries no longer compliant with the modified schema. The execution of the slapschema tool after modifying the schema can point out inconsistencies that would otherwise surface only when inconsistent entries need to be modified.

The entry records are checked in database order, not superior first order. The entry records will be checked considering all (user and operational) attributes stored in the database. Dynamically generated attributes (such as subschemaSubentry) will not be considered.

Organització de les pàgines man del servidor

Podeu consultar l'ajuda de les ordres client ldap consultant les seves pàgines man:

# man slap					
<i>slapacl</i>	<i>slapd-config</i>	<i>slapdn</i>	<i>slapd-sock</i>	<i>slapo-dyngroup</i>	<i>slapo-sock</i>
<i>slapadd</i>	<i>slapd-dnssrv</i>	<i>slapd-ndb</i>	<i>slapd-sql</i>	<i>slapo-dynlist</i>	<i>slapo-sssvlv</i>
<i>slapauth</i>	<i>slapd-hdb</i>	<i>slapd-null</i>	<i>slapindex</i>	<i>slapo-memberof</i>	<i>slapo-syncprov</i>
<i>slapcat</i>	<i>slapd-ldap</i>	<i>slapd.overlays</i>	<i>slapo-accesslog</i>	<i>slapo-pbind</i>	<i>slapo-translucent</i>
<i>slapd</i>	<i>slapd-ldbm</i>	<i>slapd-passwd</i>	<i>slapo-auditlog</i>	<i>slapo-pcache</i>	<i>slapo-unique</i>
<i>slapd.access</i>	<i>slapd-ldif</i>	<i>slapd-perl</i>	<i>slapo-chain</i>	<i>slapo-ppolicy</i>	<i>slapo-valsort</i>
<i>slapd.backends</i>	<i>slapd-mdb</i>	<i>slapd.plugin</i>	<i>slapo-collect</i>	<i>slapo-refint</i>	<i>slappasswd</i>
<i>slapd-bdb</i>	<i>slapd-meta</i>	<i>slapd-relay</i>	<i>slapo-constraint</i>	<i>slapo-retcode</i>	<i>slapschema</i>
<i>slapd.conf</i>	<i>slapd-monitor</i>	<i>slapd-shell</i>	<i>slapo-dds</i>	<i>slapo-rwm</i>	<i>slaptest</i>

Observeu que estan organitzades segons es tracti de:

- ❑ Odres de servidor.
- ❑ El dimoni slapd.
- ❑ Descripció de access, backends, overlay i plugins (slapd.xxx).
- ❑ Descripció de varis temes (slapd-xxx).

Configuració de ACLs

Es poden definir múltiples ACLs per als elements de la base de dades fins i tot a nivell d'atributs. **Sempre** rootdn pot modificar (accedir i modificar) qualsevol contingut, per alguna cosa es root!.

El tractament de ACLs es similar als altres àmbits informàtics on es poden aplicar i com es habitual, pot acabar essent bastant complicat. En la forma més simple son fàcils d'entendre.

Reflexions sobre les dades i les ACLs:

- Un usuari pot modificar *totes* les seves dades? Puc passar de profe a Conseller d'Ensenyament amb un simple `ldapmodify`? Modificar la nòmina?
- Els altres usuaris quines dades poden observar? El password dels altres? L'adreça?
- Hi ha un camp que tot usuari ha de poder modificar per ell mateix, el seu propi password.

nota a vegades és fàcil confondre's i escriure les ACL al final del document fora del bloc de la base de dades on realment es pretenia configurar aquestes ACLs.

Conceptes generals

[consultar la documentació de: `man slapd.access`, `man slapd.conf` i el **capítol 8 ACLs Administració LDAP**]

En el la configuració de slapd.d es poden especificar ACLs globals per a totes les bases de dades, es la secció de definició de directives globals.

Per a cada base de dades es poden descriure ACLs específiques per a cada base de dade. Ull a posar-les al lloc pertinent dins de cada bloc database.

Es genera una llista de regles ACL concatenant al final de les regles específiques de cada BD les regles globals (que tenen menys precedència que les específiques de la BD).

Les regles s'apliquen de manera seqüencial començant per la primera i anar descendint. Si una regla fa match al *what* ja no s'examinen més regles. Com a totes les ACL de tota la vida! Per tant l'ordre en que s'escriuen és determinant.

A més a més recordar que:

- L'usuari rootdn te drets totals per a fer-ho tot. Independentment de les ACLs.

- L'usuari *anonymous* no pot realitzar actualitzacions (encara que les ACLs ho permetessin).
- Per defecte si no s'ha indicat cap ACL es permet la lectura de tot per a tothom i només rootdn pot realitzar actualitzacions.
`access to * by * read`
- Si hi ha alguna ACL definida (sigui global, local a la BD o totes dues) s'afegeix al final implícitament una regla que denega qualsevol tipus d'accés.
`access to * by * none`
- Un cop s'està processant una entrada ACL que ha fet match (s'ha seleccionat el *what*) es comproven els operadors *by*, si no se n'aplica cap implícitament al final es denega qualsevol access.
`by * none`
- els drets són acumulatius, evidentment si es té el dret de write implica el dret de read, search, etc.

Exemple-01:

`access to * by * read`

La ACL d'exemple indica que:

- qualsevol usuari, anònim o autènticat pot veure totes les dades de la BD.
- únicament rootdn pot realitzar actualitzacions.

Exemple-02:

`access to * by * write`

La ACL d'exemple indica que:

- qualsevol usuari pot modificar qualsevol de les dades de la BD, tant les seves com les dels altres.
- tot i que la ACL permet drets de write a tothom (anònims i usuaris autènticats) cal recordar que els usuaris anònims no poden realitzar actualitzacions.
- els drets són acumulatius, evidentment si es té el dret de write implica el dret de read, search, etc. És a dir, tothom pot observar totes les dades de la base de dades (també els usuaris anònims).
- evidentment rootdn pot realitzar actualitzacions.

Exemple-03:

`access to * by self write by * read`

La ACL d'exemple indica que:

- qualsevol usuari, anònim o autènticat pot veure totes les dades de la BD.
- un usuari pot modificar totes les seves pròpies dades.

- rootdn pot realitzar tot tipus d'actualitzacions.

Exemple-04:

```
access to attrs=homePhone by * read
access to * by * write
```

La ACL d'exemple indica que:

- tots els usuaris poden modificar qualsevol dada de la BD excepte el camp homePhone que es únicament de lectura.
- si s'està processant el camp homePhone fa *match* la primera regla i s'assigna dret de read i ja no es continua avaluant les següents regles.
- si s'està avaluant un camp/element que no és homePhone no fa *match* la primera regla i si que s'aplica la segona regla (amb l'operador *).
- rootdn pot realitzar tot tipus d'actualitzacions.

Exemple-05 ok:

```
access to attrs=homePhone
    by dn.exact="cn=Anna Pou,ou=usuaris,dc=edt,dc=org" write
    by * read
access to * by * write
```

La ACL d'exemple indica que:

- tots els usuaris poden modificar qualsevol dada de la BD excepte el camp homePhone que es únicament de lectura.
- excepte la usuària "Anna pou" que sí que té el dret de modificar els homePhone.
- si s'està processant el camp homePhone fa *match* la primera regla i llavors es valida qui és l'usuari, si és "Anna Pou" pot modificar-lo, si és un altre usuari (o anònim) només pot fer read.
- rootdn pot realitzar tot tipus d'actualitzacions.

Exemple-05 ko(malament):

```
access to attrs=homePhone
    by dn.exact="cn=Anna Pou,ou=usuaris,dc=edt,dc=org" write
access to attrs=homePhone
    by * read
access to * by * write
```

La ACL d'exemple és incorrecte, perquè:

- si s'està processant el camp homePhone fa *match* amb la primera clausula what. Llavors passa a avaluar-se l'operador *by*.
- si l'usuari es "Anna Pou" s'aplica el write.
- però que passa si no és "Anna Pou"? **NO** es passa a la següent clàusula what sinó que es mira si hi ha un altre *by* (dins el mateix what) com que no hi és implícitament és com si hi ha un *by * none*.

- En realitat la ACL incloent els valors implícits és com si diu:
`access to attrs=homePhone`
`by dn.exact="cn=Anna Pou,ou=usuaris,dc=edt,dc=org" write`
`by * none`
`access to attrs=homePhone by * read`
`access to * by * write`
`access to * by * none`
- on podem observar en vermell les definicions implícites. la primera d'elles es la que fa que si no es tracta de la "Anna Pou" ningú pugui accedir al homePhone.
- fixeu-vos també que a la segona clàusula what no s'hi entrarà mai! Igual que a la última regla implícita (en vermell) que tampoc s'hi entrarà mai.

Exemple-06:

```
access to attrs=homePhone
  by dn.exact="cn=Anna Pou,ou=usuaris,dc=edt,dc=org" write
  by dn.exact="cn=Admin System,ou=usuaris,dc=edt,dc=org" write
  by * read
access to *
  by dn.exact="cn=Admin System,ou=usuaris,dc=edt,dc=org" write
  by self write
  by * read
```

La ACL d'exemple indica que:

- el homePhone només pot ser actualitzat per "Anna Pou" i per "Admin System" (de tothom!). Tots els altres usuaris i l'usuari anònim només en poden fer read.
- tots els altres camps poden ser modificats per l'usuari "Admin System" i per el propi usuari (self). Els altres usuaris poden fer read de les dades però no actualitzar dades d'altres usuaris.
- rootdn pot realitzar tot tipus d'actualitzacions.

Exemple-07:

Tot usuari es pot modificar el seu propi password i tothom pot veure totes les dades de tothom, excepte els altres passwords.

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

Podeu trobar més exercicis a [practica_ldap_acls.pdf](#)

RDN i homePassword

L'atribut que s'utilitzi com a RDN per identificar els dn dels usuaris i l'atribut homePassword tenen una consideració especial (que potser han de tenir també altres atributs). Això és degut a que són utilitzats per a fer el BIND dels usuaris.

Anem a pams, quan un usuari realitza una acció identificant-se amb el seu dn i password, el client ldap primer es connecta al servidor com a usuari anònim. Llavors intenta validar el dn amb el password. Un cop validat realitza l'acció encomanada com a tal usuari.

És a dir, primerament hi ha l'autenticació de l'usuari, que com a "anonymous" ha de poder accedir al dn i al homePassword per poder fer l'autenticació.

Si algun d'aquests atributs és none no es podrà fer l'autenticació.

Cal almenys assignar un accés de **auth** per a l'atribut homePaaswd.

Cal almenys assignar un accés de **auth/search** als altres atributs necessaris per accedir a l'element (elements del dn, en especial el RDN).

Conclusió:

- Après que el ***** ha d'anar al final o emmascara tots els altres **to**.
- Après que no es poden posar dos cops un mateix atribut perquè no més entrarà al primer.
- Un cop entrat a un **to** hi ha una llista de **by**, si no hi ha el ***** aquest serà none.
- L'atribut **userPassword** i el que s'utilitza com a RDN han de tenir permís de **auth/search** si no no es podran fer els bind d'autenticació.

Synopsis ACLs

Ldap Documentation ACLs

If no access controls are present, the default policy allows anyone and everyone to read anything but restricts updates to rootdn. (e.g., "**access to * by * read**").

When dealing with an access list, because the global access list is effectively appended to each per-database list, if the resulting list is non-empty then the access list will end with an implicit **access to * by * none** directive. If there are no access directives applicable to a backend, then a default read is used.

Be warned: the **rootdn** can always read and write EVERYTHING!

access to <what> [by <who> [<access>] [<control>]]+

Grant access (specified by <access>) to a set of entries and/or attributes (specified by <what>) by one or more requestors (specified by <who>).

The field <what> specifies the entity the access control directive applies to. It can have the forms:

dn[.<dnstyle>]=<dnpattern>
filter=<ldapfilter>

attrs=<attrlist>[val[/matchingRule][.<attrstyle>]=<attrval>]

The field **<who>** indicates whom the access rules apply to. Multiple **<who>** statements can appear in an access control statement, indicating the different access privileges to the same resource that apply to different accesses. It can have the forms:

```
*
anonymous
users
self[.<selfstyle>]
dn[.<dnstyle>[,<modifier>]]=<DN>
```

THE <ACCESS> FIELD

The optional field **<access>** ::= [[real]self]{<level>|<priv>} determines the access level or the specific access privileges the **who** field will have. Its components are defined as

```
<level> ::= none|disclose|auth|compare|search|read|{write|add|delete}|manage
<priv> ::= {=|+|-}{0|d|x|c|s|r|{w|a|z}|m}+
```

The modifier **self** allows special operations like having a certain access level or privilege only in case the operation involves the name of the user that's requesting the access.

The level access model relies on an incremental interpretation of the access privileges. The possible levels are **none**, **disclose**, **auth**, **compare**, **search**, **read**, **write**, and **manage**.

Each access level implies all the preceding ones, thus **manage** grants all access including administrative access. The **write** access is actually the combination of **add** and **delete**, which respectively restrict the **write** privilege to **add** or **delete** the specified **<what>**.

The **none** access level disallows all access including disclosure on error.

The **disclose** access level allows disclosure of information on error.

The **auth** access level means that one is allowed access to an attribute to perform authentication/authorization operations (e.g. **bind**) with no other access. This is useful to grant unauthenticated clients the least possible access level to critical resources, like passwords.

Operation requirements

Operations require different privileges on different portions of entries. The following summary applies to primary database backends such as the BDB and HDB backends. Requirements for other backends may (and often do) differ.

The **add** operation requires **add (=a)** privileges on the pseudo-attribute entry of the entry being added, and **add (=a)** privileges on the pseudo-attribute children of the entry's parent. When adding the suffix entry of a database, **add** access to children of the empty DN ("") is required. Also if Add content ACL checking has been configured on the database (see the `slapd.conf(5)` or `slapd-config(5)` manual page), **add (=a)** will be required on all of the attributes being added.

The **bind** operation, when credentials are stored in the directory, requires auth (**=x**) privileges on the attribute the credentials are stored in (usually userPassword).

The **compare** operation requires compare (**=c**) privileges on the attribute that is being compared.

The **delete** operation requires delete (**=z**) privileges on the pseudo-attribute entry of the entry being deleted, and delete (**=d**) privileges on the children pseudo-attribute of the entry's parent.

The **modify** operation requires write (**=w**) privileges on the attributes being modified. In detail, add (**=a**) is required to add new values, delete (**=z**) is required to delete existing values, and both delete and add (**=az**), or write (**=w**), are required to replace existing values.

The **modrdn** operation requires write (**=w**) privileges on the pseudo-attribute entry of the entry whose relative DN is being modified, delete (**=z**) privileges on the pseudo-attribute children of the old entry's parents, add (**=a**) privileges on the pseudo-attribute children of the new entry's parents, and add (**=a**) privileges on the attributes that are present in the new relative DN. Delete (**=z**) privileges are also required on the attributes that are present in the old relative DN if deleteoldrdn is set to 1.

The **search** operation, requires search (**=s**) privileges on the entry pseudo-attribute of the searchBase (NOTE: this was introduced with OpenLDAP 2.4). Then, for each entry, it requires search (**=s**) privileges on the attributes that are defined in the filter. The resulting entries are finally tested for read (**=r**) privileges on the pseudo-attribute entry (for read access to the entry itself) and for read (**=r**) access on each value of each attribute that is requested. Also, for each referral object used in generating continuation references, the operation requires read (**=r**) access on the pseudo-attribute entry (for read access to the referral object itself), as well as read (**=r**) access to the attribute holding the referral information (generally the ref attribute).

Exemples de consulta de permisos

L'ordre `slapacl` permet consultar els permisos assignats a una entrada de la base de dades, una entity i també els permisos assignats als atributs. També permet consultar si un usuari concret té o no determinats drets.

Observar que tota entitat consta de dos elements anomenats entity i children que determinen els permisos de la 'caixa' de l'entitat i els permisos dels 'fills' de l'entitat. Es pot pensar en un sistema equivalent al dels permisos de fitxers i de directoris.

```
# slapacl -b 'ou=usuaris,dc=edt,dc=org'  
entry: read(=rscxd)  
children: read(=rscxd)  
ou=usuaris: read(=rscxd)  
description=Container per usuaris del sistema linux: read(=rscxd)
```

```

objectClass=organizationalUnit: read(=rscxd)
...

# slapacl -b 'cn=Anna Pou,ou=usuaris,dc=edt,dc=org'
entry: read(=rscxd)
children: read(=rscxd)
objectClass=posixAccount: read(=rscxd)
objectClass=inetOrgPerson: read(=rscxd)
cn=Anna Pou: read(=rscxd)
cn=Anita Pou: read(=rscxd)
sn=Pou: read(=rscxd)
mail=anna@edt.org: read(=rscxd)
ou=Alumnes: read(=rscxd)
uid=anna: read(=rscxd)
uidNumber=5002: read(=rscxd)
gidNumber=600: read(=rscxd)
homeDirectory=/tmp/home/anna: read(=rscxd)
userPassword=****: read(=rscxd)
...

# slapacl -b 'cn=Anna Pou,ou=usuaris,dc=edt,dc=org' 'mail'
mail: read(=rscxd)

# slapacl -b 'cn=Anna Pou,ou=usuaris,dc=edt,dc=org' 'mail/read'
read access to mail: ALLOWED
# slapacl -b 'cn=Anna Pou,ou=usuaris,dc=edt,dc=org' 'mail/write'
write access to mail: DENIED
# slapacl -b 'cn=Anna Pou,ou=usuaris,dc=edt,dc=org' 'mail/add'
add access to mail: DENIED

# slapacl -D 'cn=Marta Mas,ou=usuaris,dc=edt,dc=org' -b 'cn=Marta
Mas,ou=usuaris,dc=edt,dc=org' 'userPassword'
authcDN: "cn=marta mas,ou=usuaris,dc=edt,dc=org"
userPassword: write(=wrscxd)

slapacl -D 'cn=Marta Mas,ou=usuaris,dc=edt,dc=org' -b 'cn=Marta
Mas,ou=usuaris,dc=edt,dc=org' 'userPassword/read'
authcDN: "cn=marta mas,ou=usuaris,dc=edt,dc=org"
read access to userPassword: ALLOWED

# slapacl -b 'cn=Marta Mas,ou=usuaris,dc=edt,dc=org' 'userPassword/read'
read access to userPassword: DENIED

```

Configuració dinàmica del servidor

Actualitzacions de la configuració del servidor 'en calent'

El servei LDAP i la configuració del servidor, del propi dimoni funcionen com una base de dades LDAP. Totes les opcions de configuració del servidor es poden consultar i actualitzar amb ordres ldap.

Recapitem, el servidor s'ha configurat amb les directives del fitxer de configuració slapd.conf amb el que s'ha generat una estructura de directoris slapd.d. Aquesta estructura de directoris té forma d'entrades LDIF.

A més a més en engegar el servidor i crear la base de dades de l'organització "edt.org" també s'han generat automàticament altres bases de dades, corresponents al propi servidor, al frontend, al les BD que es creen i al monitor (si s'ha activat).

La base de dades config es crea sempre (tant si s'indica en el fitxer de configuració com si no) i representa tota la configuració del servidor slapd. Es poden fer consultes i actualitzacions ldap sobre aquesta base de dades i això generarà canvis 'en calent' en la configuració del servidor slapd.

****atenció:****

Per poder accedir a la base de dades "cn=config" corresponent a la configuració del dimoni del servei slapd cal definir un rootdn i un rootpw per a l'entrada database config. Aquesta entrada és única. L'exemple mostra què cal incloure de nou en la configuració de slapd.conf. Caldrà eliminar i **regenerar** de nou la configuració, **no** les dades.

part afegida al slapd.conf

```
database config
rootdn "cn=Sysadmin,cn=config"
rootpw {SSHA}5DfZc1WXelwrP7C3fr23WLZiPZ5YHMgA
# el passwd es syskey
```

```
# systemctl stop slapd
# rm -rf /etc/openldap/slapd.d/*
# slaptest -f slapd-cn=configl-edt.org.conf -F /etc/openldap/slapd.d/
config file testing succeeded
# chmod -R ldap.ldap /etc/openldap/slapd.d/
# systemctl start slapd
```

Ara autenticats amb l'usuari "cn=Sysadmin,cn=config" (atenció a no posar el dc=edt,dc=org!) es poden realitzar consultes ldap a la configuració del servidor. Podem llistar totes les entrades LDAP igual que si es realitzés l'ordre "*slapcat -n0*". També llistar les dades per a una BD concreta (per exemple la de "dc=edt,dc=org". I fins i tot mostrar els valors concrets d'una directiva de configuració (un atribut en la BD ldap).

```
# slapcat -n0
# ldapsearch -x -h localhost -LLL -D 'cn=Sysadmin,cn=config' -w syskey -b 'cn=config'
# ldapsearch -x -h localhost -LLL -D 'cn=Sysadmin,cn=config' -w syskey -b 'cn=config' dn
```

```
# configuració de la bd{1} edt.org
# ldapsearch -x -h localhost -LLL -D 'cn=Sysadmin,cn=config' -w syskey -b
'olcDatabase={1}bdb,cn=config'

# configuració de les ACLs
# ldapsearch -x -h localhost -LLL -D 'cn=Sysadmin,cn=config' -w syskey -b
'olcDatabase={1}bdb,cn=config' olcAccess

# configuració dels índex
# ldapsearch -x -h localhost -LLL -D 'cn=Sysadmin,cn=config' -w syskey -b
'olcDatabase={1}bdb,cn=config' olcDbIndex
```

També es poden fer actualitzacions a la base de dades “cn=config” usant les mateixes eines de ldap (ldapmodify, ldapadd, etc). Cal autenticar-se com a rootdn per poder realitzar les actualitzacions. A continuació es pot veure un exemple de fitxer LDIF usat per a la modificació de la configuració del servidor:

```
dn: olcDatabase={1}bdb,cn=config
changetype: modify
delete: olcAccess
-
add: olcAccess
olcAccess: to * by * read
```

El fitxer entry.ldif anterior elimina totes les ACLs de la base de dades {1} que correspon a “dc=edt,dc=org” i a continuació estableix una única ACL amb drets de lectura per a tothom.

```
# slapcat -n0 | grep olcAccess

# ldapmodify -x -h localhost -D 'cn=Sysadmin,cn=config' -w syskey -f entry.ldif

# slapcat -n0 | grep olcAccess
# ldapsearch -x -h localhost -LLL -D 'cn=Sysadmin,cn=config' -w syskey -b
'olcDatabase={1}bdb,cn=config' olcAccess
```

Instal·lació d'un servei (repàs general)

Passos a realitzar:

- instal·lar el servei.
- observar el contingut dels paquets instal·lats.
- identificar els directoris de configuració, dades, etc.
- identificar el nom del servei i saber fer: start, stop, status, enable, disable.
- identificar el fitxer de configuració.
- identificar el fitxer del PID del servei i observar-lo en l'arbre de processos. Llistar els serveis actius (list-units) i observar que en forma part.

- Identificar el directori de logs del servei i observar el tipus de logs que genera.
- Identificar el fitxer de lock del subsystem que impedeix a altres serveis LDAP engegarse o usar recursos apropiats.

Configuració LDAP Client

Observar el contingut del fitxer de configuració client */etc/openldap/ldap.conf*.

Exemple del fitxer client */etc/openldap/ldap.conf* per defecte que s'instal·la.

```
# cat /etc/openldap/ldap.conf
#
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

#BASEdc=example,dc=com
#URI ldap://ldap.example.com ldap://ldap-master.example.com:666

#SIZELIMIT 12
#TIMELIMIT 15
#DEREF never

TLS_CACERTDIR /etc/openldap/certs

# Turning this off breaks GSSAPI used with krb5 when rdns = false
SASL_NOCANON on
```

Exemple de fitxer client */etc/openldap/ldap.conf* d'un host de les aules d'informàtica:

```
# cat /etc/openldap/ldap.conf
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

#BASEdc=example,dc=com
#URI ldap://ldap.example.com ldap://ldap-master.example.com:666

#SIZELIMIT 12
#TIMELIMIT 15
#DEREF never

TLS_CACERTDIR /etc/openldap/certs

# Turning this off breaks GSSAPI used with krb5 when rdns = false
SASL_NOCANON on

URI ldap://ldap/
BASE dc=escoladeltreball,dc=org
```

Eines gràfiques: gq i phpldapadmin

Eina gràfica: gq

Utilitzar l'eina gràfica gq per consultar la base de dades ldap. Examinar les entitats i els seus atributs. En especial identificar els atributs requerits i opcionals.

El gq **falla molt sovint**, però és realment pràctic per **examinar els objectclass i els seus atributs**, ajuda molt en la creació de nous schema (consultant la part de schema).

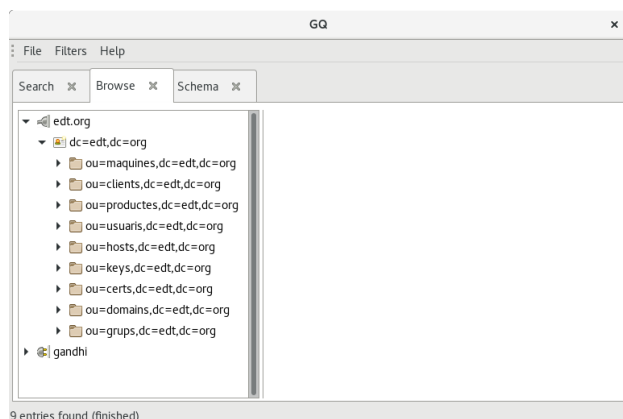
```
# dnf -y install gq
$ gq &
```

Editar el server amb el que contactar:

Menú File:

- Preferences
 - Servers
 - edit:
 - name: edt.org (o el nom identificatiu de la connexió que es desitgi)
 - hostname: 172.17.0.2 (si és un container docker) indicar el nom de host o adreça IP del servidor ldap.
- Browse:
 - seleccionar la connexió edt.org
 - passwd: usar <enter> si fem un bind anònim.
 - podem observar tot l'arbre DIT de dc=edt,dc=org
- Schema
 - podem observar tots els schema carregats
 - també els atributs, objectes, matching rules i syntaxes.

Podem definir tantes connexions a servidor com ens calguin. Podem accedir a altres servidors ldap remots indicant un nou server i configurant-ne l'accés.



Eina gràfics: phpldapadmin

Per utilitzar des d'un navegador web l'eina gràfica phpldapadmin cal instal·lar i configurar phpldapadmin i també apache (servei httpd).

Podeu consultar documentació de com configurar phpldapadmin en el “chapter 8 LDAP and the Web” apartat “phpLDAPAdmin” del llibre “Mastering OpenLDAP. Configuring, Securing and Integrating Directory Services” de l'editorial Pack Publishing.

En principi phpldapadmin està configurat per ser accedit **localment**, incorpora un servidor http i és a través del port 80 local via **localhost/phpldapadmin** que podrem accedir a l'administració web del ldap.

Si volem permetre l'accés remot a la seu web de phpldapadmin caldrà configurar el servidor web Apache per tal de permetre connexions remotes (per defecte està configurat per permetre només l'accés local al port 80)..

****nota**** Si volem per poder engegar el container amb el servidor ldapserver però atacar al port 389 del host i no de la ip local del container cal mapejar al host el port ldap:

```
$ docker run -p 389:389 -d edtasixm06/ldapserver:18basecn
```

Procediment de treball:

- instal·lar el software
- configurar el phpldapadmin perquè accedeixi al repositori de dades ldap (al nostre container ldapserver).
- configurar apache (el servidor web) perquè la web de phpldapadmin sigui accessible externament i no únicament localment.
- accedir a la pàgina web d'administració ldap.

Instal·lar el software:

```
# dnf install phpldapadmin php-xml httpd
# systemctl start httpd
# systemctl status httpd
# https -S
```

Configurar el servei phpldapadmin:

```
# vim /etc/phpldapadmin/config.php

$servers->newServer('ldap_pla');
$servers->setValue('server','name','Local LDAP Server');
$servers->setValue('server','host','172.17.0.2');
$servers->setValue('server','port',389);
$servers->setValue('server','base',array('dc=edt,dc=org'));
$servers->setValue('login','auth_type','session');
$servers->setValue('server','tls',false)
```

```
// $servers->setValue('appearance','password_hash','sha1');  
// comentada perquè generava un error  
// $servers->setValue('login','attr','uid');  
// comentada perquè generava un error
```

Configurar Apache per permetre l'accés remot

```
# vim /etc/httpd/config.d/phpldapadmin.conf
```

```
Alias /phpldapadmin /usr/share/phpldapadmin/htdocs  
Alias /ldapadmin /usr/share/phpldapadmin/htdocs  
<Directory /usr/share/phpldapadmin/htdocs>  
  <IfModule mod_authz_core.c>  
    Require all granted  
  </IfModule>  
  <IfModule !mod_authz_core.c>  
    Allow from *  
  </IfModule>  
</Directory>
```

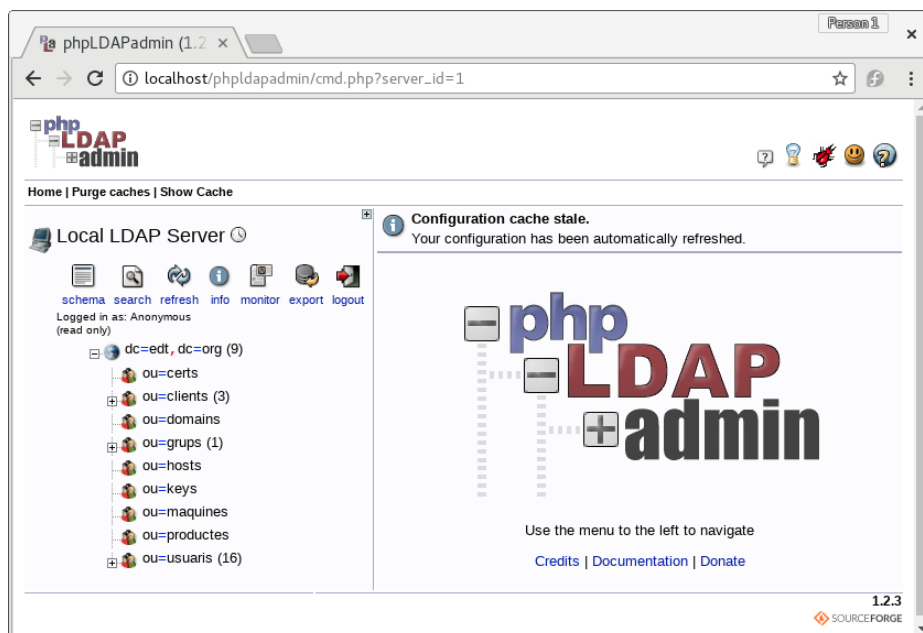
Exemple de connexió com a usuari anònim.

Per accedir a la web d'administració de phpldapadmin cal contactar al port 80 del host on s'està executant el servei web Apache que conté aquesta web.

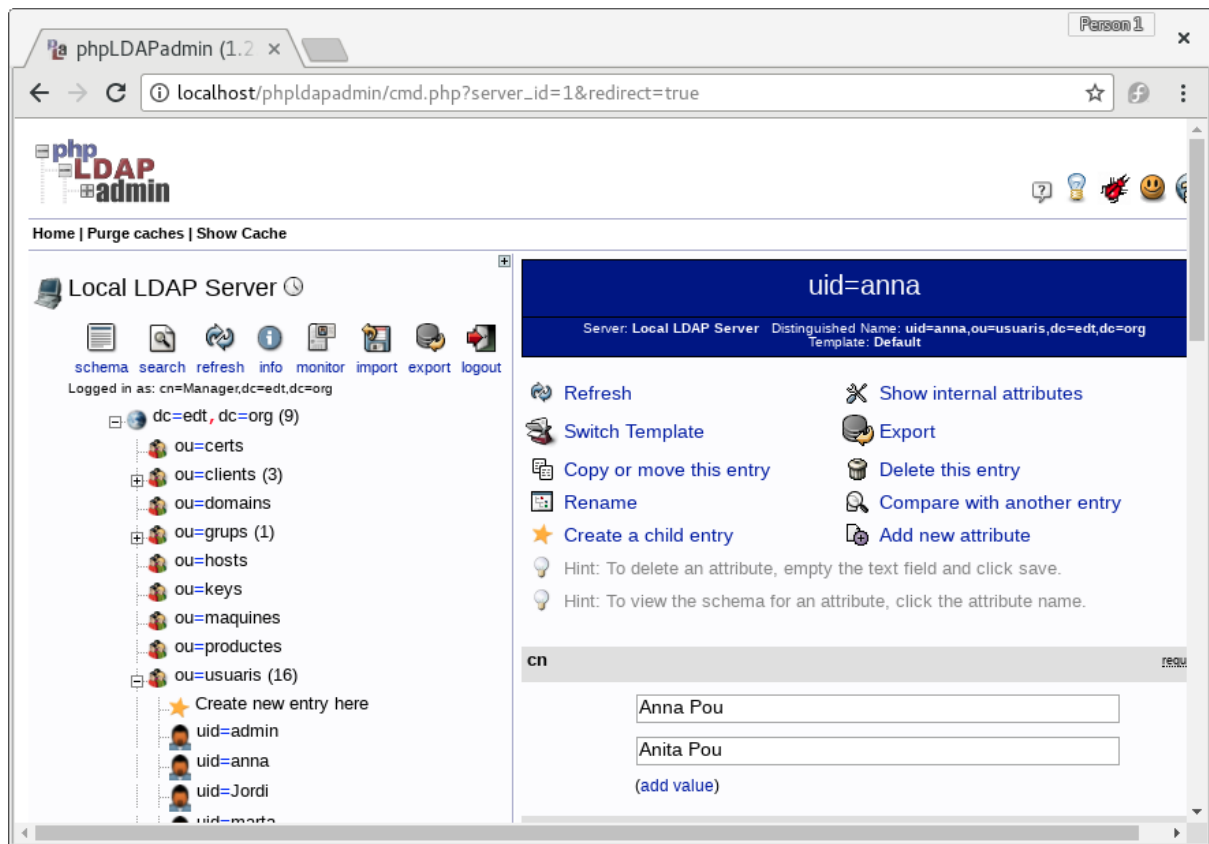
Si ho fem localment caldrà indicar la URL: **localhost/phpldapadmin**

Si ho fem remotament caldrà indicar la URL: **http://172.17.0.3/phpldapadmin** suposant que aquesta és l'adreça IP on s'executa el servei.

Exemple d'accés local:



Coma **anonymous** simplement podem consultar informació (si les ACLs ho permeten). Podem identificar-nos com un usuari en l'apartat de login. Si fem el Bind com a **rootDN** es pot administrar tota la base de dades.



Observei l'estrella amb l'opció "Create New user" i les opcions Copy, Rename, etc que apareixen.

Schema: creació d'schema/objectes/atributs propis

En aquest apartat es mostraran exemples de creació d'atributs, objectes i esquemes propis. Consultar la documentació de la guia d'administració ldap disponible localment (file:///usr/share/doc/openldap-servers/guide.html) o a la seu web de openldap (<http://www.openldap.org/doc/admin24/>). Consultar el capítol (13) "Schema specification".

També es recomana consultar la documentació de la web www.ldap.com, en especial l'apartat [Understanding LDAP schema](#).

Una altra manera de practicar i aprendre el funcionament dels atributs, objectes i schema es navegar per aquests elements a la pestanya Schema del visor gràfic gq.

Es treballaran els següents exemples:

- Implementar els exemples de la documentació de openldap, creant atributs i objectes propis. Carregar l'Schema a la base de dades i carregar dades.
- Implementar un schema propi amb dades de futbolistes. Creant l'objecte futbolista i els seus atributs. Aquest objecte serà derivat de inetOrgperson. Carregar l'Schema a la base de dades i carregar dades.
- Modificar l'exemple anterior fent de l'objecte un objecte estructural que no deriva de cap altre objecte, és a dir, derivat de TOP. Carregar l'Schema a la base de dades i carregar dades.
- Fer un tercer exemple amb futbolistes on l'objecte futbolista és de tipus auxiliar. Això implica que cada entitat futbolista de la base de dades ha d'anar associada a un objecte estructural (s'utilitzarà inetorgPerson). Carregar l'Schema a la base de dades i carregar dades.

Exemples de creació d'Schema

Exemple general (documentació)

Exemples futbolista

Llistat d'exemple de l'schema que implementa els exemples de la documentació openldap:

```

attributetype ( 1.1.2.1.1 NAME 'x-my-UniqueName'
  DESC 'unique name with my organization'
  SUP name )
#
attributetype ( 1.1.2.1.2 NAME 'x-my-UniqueOrg'
  DESC 'unique name of my organization'
  EQUALITY caseIgnoreMatch
  SUBSTR caseIgnoreSubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
  SINGLE-VALUE )
#
attributetype ( 1.1.2.1.3 NAME 'x-my-Photo'
  DESC 'a photo (application defined format)'
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.40
  SINGLE-VALUE )
#
attributetype ( 1.1.2.1.4 NAME 'x-my-PhotoURI'
  DESC 'URI and optional label referring to a photo'
  SUP labeledURI )

# Exemple objecte
objectclass ( 1.1.2.2.1 NAME 'x-my-data'
  DESC 'nou tipus objecte my-data'
  AUXILIARY

```

```

MUST description
MAY ( x-my-UniqueOrg $ x-my-Photo $ x-my-PhotoURI $ countryName $
      localityName ) )

# Exemple objecte
objectclass ( 1.1.2.2.2 NAME 'x-my-Person'
  DESC 'nou tipus objecte my-person'
  SUP inetOrgPerson
  STRUCTURAL
  MUST description
  MAY ( x-my-UniqueName $ countryName $ localityName $ streetAddress $
        stateOrProvinceName ) )

```

Llistat d'un fitxer LDIF amb dades injectables a la base de dades dc=edt,dc=org d'entitats que utilitzen l'schema definit anteriorment:

```

dn: o=depinf,ou=maquines,dc=edt,dc=org
objectclass: organization
objectclass: x-my-data
o: depinf
description: organitzacio depinf amb un objectclass
             compost de organization i x-my-data
             camp obligatoris o i description
countryName: uk
localityName: barcelona

dn: cn=Persona01,ou=maquines,dc=edt,dc=org
objectclass: x-my-person
cn: Persona01
sn: paio01
description: objecte x-my-data derivat de inetorgpersona
             els camps must son cn i sn derivats de Person
             coma camp must del derivat hi ha la description
countryName: uk
localityName: barcelona

```

Finalment observar com s'ha modificat el fitxer de configuració slapd.conf per incloure aquest schema, verificar-lo i regenerar la configuració

```

# cat slapd.conf
...
include                /var/tmp/m06/schema/mysch01.schema

# slaptest -f slapd.conf
# slatest -f slapd.conf -F /etc/openldap/slapd.d

# slapcat -n0 | grep dn:

```

Carregar i comprovar les dades de la BD:

```
# ldapadd -xvh localhost:386 -D cn=Manager,dc=edt,dc=org' -w secret -f dades-ldif
# slapcat | grep dn:
```

Cas-A: Derivat de inetorgperson

Implementar les dades de un futbolista (equip, dorsal, webpage, foto, lesinat) creant un nou objecte x-Futbolistes derivat de inetOrgPerson.

Exemple del fitxer fut.schema

```
attributetype ( 1.1.2.1.1 NAME 'x-equip'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
    SINGLE-VALUE )

attributetype ( 1.1.2.1.2 NAME 'x-dorsal'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
    SINGLE-VALUE )

attributetype ( 1.1.2.1.4 NAME 'x-webpage'
    DESC 'MOD urlabel for webpage'
    EQUALITY caseExactMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributetype ( 1.1.2.1.5 NAME 'x-photo'
    DESC 'fotosssss'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)

attributetype ( 1.1.2.1.6 NAME 'x-lesionat'
    DESC 'lesionat TRUE'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
    SINGLE-VALUE)

objectclass ( 1.1.2.2.1 NAME 'x-Futbolistes'
    DESC 'Futboleros'
    SUP inetOrgPerson
    STRUCTURAL
    MUST x-equip
    MAY (x-dorsal $ x-webpage $ x-photo $ x-lesionat) )
```

Exemple de fitxer LDIF amb dades de futbolistes:

```
dn: cn=Vladimir remar,ou=Productes,dc=edt,dc=org
```


objectclass: x-Futbolistes

cn: vladimir remar

cn: VLAD ThE IMPaLer

sn: vladi

x-equip: los pimientos

x-dorsal: 7

x-webpage: www.vladimir.remar

x-photo: ///var/tmp/myphoto.jpg

x-lesionat: FALSE

homephone: 555-212-2220

mail: vladimir.remar@gmail.com

dn: cn=renzo remar,ou=Productes,dc=edt,dc=org

objectclass: x-Futbolistes

cn: renzo remar

cn: RARS0

sn: renzo

x-equip: los pimientos

x-dorsal: 10

x-webpage: www.renzo.remar

x-photo: ///var/tmp/myphoto2.jpg

x-lesionat: TRUE

homephone: 555-212-2221

mail: renzo.remar@gmail.com

dn: cn=Jose Miguel Remar,ou=Productes,dc=edt,dc=org

objectclass: x-Futbolistes

cn: Jose miguel Remar

cn: chetos

sn: jose

x-equip: los aguacates

x-dorsal: 1

x-webpage: www.josemi.remar

x-photo: ///var/tmp/myphoto3.jpg

homephone: 555-212-2223

mail: joseMi.remar@gmail.com

Cas-B: Structural standalone object

Modificació de l'exemple anterior per generar un objecte x-Futbolistes que sigui estructural pero no derivi de cap altre objecte. Bé, de fet si deriva de TOP que és una classe abstracta de la qual deriven tots.

Ara ja no hereta els atributs i propietats de inetOrgPerson com en el cas anterior. Això obliga a afegir un camp nom de futbolista per poder-lo identificar apropiadament i usar-lo en el dn.

Observar que els camps cn, sn, homePhone i mail usats en l'exemple anterior ja no són necessaris.

Exemple de fitxer schema:

```
attributetype ( 1.1.2.1.7 NAME ( 'x-nom' 'lonom' )
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
    SINGLE-VALUE )

attributetype ( 1.1.2.1.1 NAME 'x-equip'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
    SINGLE-VALUE )

attributetype ( 1.1.2.1.2 NAME 'x-dorsal'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
    SINGLE-VALUE )

attributetype ( 1.1.2.1.4 NAME 'x-webpage'
    DESC 'MOD urlabel for webpage'
    EQUALITY caseExactMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

attributetype ( 1.1.2.1.5 NAME 'x-photo'
    DESC 'fotosssss'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)

attributetype ( 1.1.2.1.6 NAME 'x-lesionat'
    DESC 'lesionat TRUE'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
    SINGLE-VALUE)

objectclass ( 1.1.2.2.1 NAME 'x-Futbolistes'
    DESC 'Futboleros'
    SUP TOP
    STRUCTURAL
    MUST ( lonom $ x-equip )
    MAY ( x-dorsal $ x-webpage $ x-photo $ x-lesionat ) )
```

Llistat de dades LDIF:

```
dn: lonom=Vladimir remar,ou=Productes,dc=edt,dc=org
objectclass: x-Futbolistes
lonom: vladimir remar
#cn: VLaD ThE IMPaLer
#sn: vladi
x-equip: los pimientos
x-dorsal: 7
x-webpage: www.vladimir.remar
x-photo: ///var/tmp/myphoto.jpg
```

```

x-lesionat: FALSE
#homephone: 555-212-2220
#mail: vladimir.remar@gmail.com

dn: lonom=renzo remar,ou=Productes,dc=edt,dc=org
objectclass: x-Futbolistes
lonom: renzo remar
#cn: RARS0
#sn: renzo
x-equip: los pimientos
x-dorsal: 10
x-webpage: www.renzo.remar
x-photo: ///var/tmp/myphoto2.jpg
x-lesionat: TRUE
#homephone: 555-212-2221
#mail: renzo.remar@gmail.com

dn: lonom=Jose Miguel Remar,ou=Productes,dc=edt,dc=org
objectclass: x-Futbolistes
lonom: Jose miguel Remar
#cn: chetos
#sn: jose
x-equip: los aguacates
x-dorsal: 1
x-webpage: www.josemi.remar
x-photo: ///var/tmp/myphoto3.jpg
#homephone: 555-212-2223
#mail: joseMi.remar@gmail.com

```

Cas-C: Objecte Auxiliary

Modificar els exemples anteriors per generar un objecte x-Futbolistes que sigui Auxiliary i que es pugui usar en entitats que combinin les dades de l'objecte estructural inetOrgPerson i x-Futbolistes.

Ara ja torna a no caldre el nom (x-nom) o almenys no cal que sigui obligatori ja que aquesta informació és als camps cn i sn. Observar les dades de un element inserit llistat amb slapcat.

Exemple de fitxer schema:

```

attributetype ( 1.1.2.1.7 NAME ( 'x-nom' 'lonom')
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
    SINGLE-VALUE )

attributetype ( 1.1.2.1.1 NAME 'x-equip'

```

EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
SINGLE-VALUE)

attributetype (1.1.2.1.2 NAME 'x-dorsal'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE)

attributetype (1.1.2.1.4 NAME 'x-webpage'
DESC 'MOD urlabel for webpage'
EQUALITY caseExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15)

attributetype (1.1.2.1.5 NAME 'x-photo'
DESC 'fotosssss'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.40)

attributetype (1.1.2.1.6 NAME 'x-lesionat'
DESC 'lesionat TRUE'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE)

objectclass (1.1.2.2.1 NAME 'x-Futbolistes'
DESC 'Futboleros'
SUP TOP
AUXILIARY
MUST x-equip
MAY (x-nom \$ x-dorsal \$ x-webpage \$ x-photo \$ x-lesionat))

Llistat de les dades LDIF:

dn: cn=Vladimir remar,ou=Productes,dc=edt,dc=org
objectclass: x-Futbolistes
objectclass: inetOrgPerson
cn: vladimir remar
cn: VLaD ThE IMPaLer
sn: vladi
x-equip: los pimientos
x-dorsal: 7
x-webpage: www.vladimir.remar
x-photo: ///var/tmp/myphoto.jpg
x-lesionat: FALSE
homephone: 555-212-2220
mail: vladimir.remar@gmail.com

dn: cn=renzo remar,ou=Productes,dc=edt,dc=org
objectclass: x-Futbolistes
objectclass: inetOrgPerson
cn: renzo remar

cn: RARS0
sn: renzo
x-equip: los pimientos
x-dorsal: 10
x-webpage: www.renzo.remar
x-photo: ///var/tmp/myphoto2.jpg
x-lesionat: TRUE
homephone: 555-212-2221
mail: renzo.remar@gmail.com

dn: cn=Jose Miguel Remar,ou=Productes,dc=edt,dc=org
objectclass: x-Futbolistes
objectclass: inetOrgPerson
cn: Jose miguel Remar
cn: chetos
sn: jose
x-equip: los aguacates
x-dorsal: 1
x-webpage: www.josemi.remar
x-photo: ///var/tmp/myphoto3.jpg
homephone: 555-212-2223
mail: joseMi.remar@gmail.com

Més enllà!

Altres activitats a realitzar per a l'aprenentatge de LDAP són:

- ☐ Monitoritzar el tràfic d'una consulta/actualització LDAP amb wireshark.
- ☐ Utilitzar eines gràfiques com GQ per a l'administració de les dades amb LDAP.
- ☐ Backup de les dades del LDAP. Desar en tar.gz tota la informació d'una base de dades. Exportar totes les dades en format LDIF.
- ☐ Backup de la configuració ldap. Desar en un tar.gz tota la configuració del servidor LDAP. Exportar la configuració del servidor a un fitxer LDIF. Exportar la configuració d'una BD concreta en format LDIF.
- ☐ Schema (I): observar els "schema", la ubicació dels fitxers de definició schema i LDIF.
- ☐ Schema (II): ampliar un schema fent una definició pròpia de dades.
- ☐ LDAP amb autenticació SASL. Tràfic segur sobre TLS.
- ☐ Servidors de rèplica (de backup).
- ☐ Servei delegat: arbre d'entitats amb diversos servidors.

Apèndix: exemples usats

Base de dades

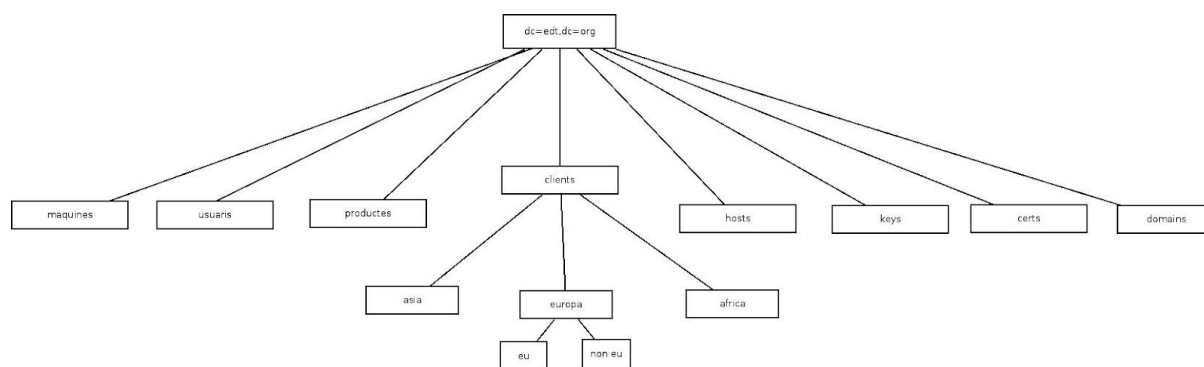
dc=edt,dc=org
rootdn=Manager
rootpwd=secret

dc=example,dc=com
rootdn=Manager
rootpw=jupiter

dc=m06,dc=cat
rootdn=Root
rootpw=class

dc=subtree,dc=edt,dc=org
rootdn=Sysadmin
rootpw=syskey

Dades DB



dc=edt,dc=org

- | | | | | | | |
|-------------|---------|-----------|---------|--------|------------|--------|
| 1) maquines | clients | productes | | | | |
| 2) usuaris | | | | | | |
| 3) hosts | keys | certs | domains | asia | africa | |
| | europa: | | | | | |
| | eu | noneu | | | | |
| 4) grups | | | | | | |
| | cup | admin | alumnes | profes | asiaafrica | europa |

5) referral de subtree

dc=example,dc=com

dc=m06,dc=cat