# LDAP
# Conceitos e Aplicações

**Antonio Carlos Feitosa Costa (antonio@cbpf.br)**

## Prefácio

Com o avanço tecnológico das redes de computadores, torna-se cada vez mais comum um ambiente computacional heterogêneo. Com este panorama vigente surge a necessidade de identificar usuários e recursos de rede usando um padrão em comum para todos os sistemas operacionais. O LDAP surge como um protocolo “curinga”, por assim dizer, que permite a autenticação de recursos e usuários duma rede usando a mesma base de informações.

O que é LDAP? Como posso usá-lo para intregrar as informações de usuários dos serviços da minha rede? Essas e outras perguntas, serão respondidas neste nota técnica.

## Conceitos

Vamos abordar alguns conceitos que servem de base para o entendimento do LDAP.

### *Serviço de Diretório*

É um serviço que usa um banco de dados com as seguintes características específicas:

- Organizado de forma hierárquica
- Otimizado para a leitura (backend storage)
- Respostas rápidas a um alto volume de consultas
- Armazenamento de informações estáticas de objetos

Exemplo:

**DNS** ( Domain Name Service)

### *X.500*

É um modelo de serviço de diretório desenvolvido pela *International Telecomunications Union* (ITU), seguindo o modelo OSI, com o objetivo de definir a ligação entre serviços de diretórios locais, formado assim, um serviço de diretório global.

Protocolo de acesso:

**DAP** (Directory Access Protocol).
**DSP** (Directory System Protocol)
**DISP** (Directory Information Shadowing Protocol)
**DOP** (Directory Operational Bindings Management Protocol)

Fonte: http://en.wikipedia.org/wiki/X.500

# LDAP

## *Descrição*

É um protocolo leve de acesso a diretórios desenvolvido pela University of Michigan, desenvolvido a partir do X.500. O X.500 tinha a desvantagem de ser altamente complexo e de custo alto. O LDAP trabalha na camada aplicação do TCP/IP permitindo que toda rede que use o TCP/IP, também utilize o LDAP, conforme a tabela abaixo:

| X.500 | TCP/IP | LDAP |
|---|---|---|
| Aplicação | Aplicação | Aplicação (Porta **389**) |
| Apresentação | | |
| Sessão | | |
| Transporte | Transporte | TCP |
| Rede | Rede | IP |
| Enlace | Enlace | |
| Física | Física | |

## *Especificações*

O LDAP possui um conjunto de especificações responsáveis pelas operações que ele realiza, abaixo temos algumas:

LDAP: Technical Specification Road Map [**RFC4510**] (draft)

LDAP: The Protocol [**RFC4511**]

LDAP: Directory Information Models **[RFC4512**]

LDAP: Authentication Methods and Security Mechanisms [**RFC4513**]

LDAP: String Representation of Distinguished Names [**RFC4514**]

LDAP: String Representation of Search Filters [**RFC4515**]

LDAP: Uniform Resource Locator [**RFC4516**]

LDAP: Syntaxes and Matching Rules [**RFC4517**]

LDAP: Internationalized String Preparation [**RFC4518**]

LDAP: Schema for User Applications [**RFC4519**]

### *Servidores*

Existem vários serviços de dirétório baseados em LDAP disponível no mercado, aqui temos os mais divulgados:

Microsoft Active Directory
OpenLDAP
Novell eDirectory
Apache Directory

Temos também uns menos cohecidos:

IBM Directory Server
Oracle Internet Directory
Netscape Directory Server
Sun ONE Directory Server

### *OpenLDAP*

É uma implementação open source do protocolo Lightweight Directory Access Protocol.

A suíte contém:

1. slapd – Servidor LDAP stand-alone (daemon);
2. slurpd – Servidor de Replicação stand-alone (daemon);
3. bibliotecas (api's, database backends, etc...);
4. utilitários, ferramentas e exemplos.

## Armazenamento

Como os dados são armazenados em diretórios ???

Esta é uma boa pergunta que será respondida neste tópico. O armazenamento dos dados em diretório é feito pelo backend storage, que é o banco de dados que armanzena as informações propriamente ditas. Mas que modela estas informações no formato de diretório é o serviço de diretório em si que utiliza de alguns padrões para isso. Vamos observar alguns padrões que o LDAP utiliza para trafegar e identificar as informações de catálogo.

### *Schema*

É um conjunto de especificações de object class e atributos.

Contém informações tais como:

- Tipo de dado que será guardado num determinado atributo;
- Cada object class e atributo possui um número de controle(OID), registrado no IANA.

Nota: Um schema pode ser criado para suprir eventuais necessidades.

### *Exemplo de schema*

```
/etc/ldap/schema/core.schema
# OpenLDAP Core schema
# $OpenLDAP: pkg/ldap/servers/slapd/schema/core.schema,v 1.79.2.7 2006/02/13
17:28:43
kurt Exp $
## This work is part of OpenLDAP Software <http://www.openldap.org/>.
#
# Standard attribute types from RFC 2256
#
# system schema
#attributetype ( 2.5.4.0 NAME 'objectClass'
# DESC 'RFC2256: object classes of the entity'
# EQUALITY objectIdentifierMatch
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
# system schema
#attributetype ( 2.5.4.1 NAME ( 'aliasedObjectName' 'aliasedEntryName' )
# DESC 'RFC2256: name of aliased object'
# EQUALITY distinguishedNameMatch
# SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 SINGLE-VALUE )
attributetype ( 2.5.4.2 NAME 'knowledgeInformation'
DESC 'RFC2256: knowledge information'
EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

### *Object Class*

É um conjunto de definições de atributos e seus respectivos tipos de valores.

Uma object class define que atributos são requeridos e quais podem ser utilizados quando ela for declarada.

A inclusão de uma ObjectClass em uma entrada determina quais os atributos deverão ser utilizados e quais poderão ser utilizados(MUST e MAY);

Ex: A objectclass person exige os atributos cn e sn, mas pode ser utilizado o atributo telephoneNumber.

Possuem derivações, uma ObjectClass pode herdar atributos requeridos de outra;

Ex: A objectclass inetorgperson tem como raiz a objectclass person, por isso os atributos cn e sn são obrigatórios.

Especificação:

RFC 4529 – Requesting Attributes by Object Class in the Lightweight Directory Access Protocol.

Exemplo de ObjectClass

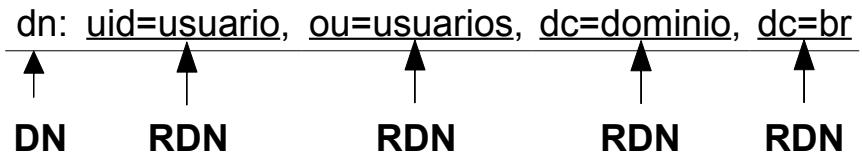
```
#/root/dominio.ldif
dn: cn=admin,dc=dominio,dc=br
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e2NyeXB0fWQ0MmJ1dU00RXJxZDI=
#eof

#/etc/ldap/schema/core.schema
...
objectclass ( 0.9.2342.19200300.100.4.19 NAME 'simpleSecurityObject'
DESC 'RFC1274: simple security object'
SUP top AUXILIARY
MUST userPassword )
...
objectclass ( 2.5.6.8 NAME 'organizationalRole'
DESC 'RFC2256: an organizational role'
SUP top STRUCTURAL
MUST cn
MAY ( x121Address $ registeredAddress $ destinationIndicator $
preferredDeliveryMethod $ telexNumber $ teletexTerminalIdentifier $
telephoneNumber $ internationaliSDNNumber $ facsimileTelephoneNumber $
seeAlso $ roleOccupant $ preferredDeliveryMethod $ street $
postOfficeBox $ postalCode $ postalAddress $
```

```
physicalDeliveryOfficeName $ ou $ st $ l $ description ) )
```

### *Atributo*

É um elemento de um dado que caracteriza e descreve uma entidade. Um conjunto de atributos forma uma entrada que é uma unidade básica de informação armazenada num diretório.

O que distingüe uma entrada da outra é o DN (Distinguished Name), que identifica uma entrada de forma não ambígua num Diretório. Os DN's são compostos por RDN (Relative Distinguished Name) e cada RDN corresponde a um ramo do diretório, desde a raiz até a entrada a qual o DN faz referência.




Especificação:

RFC 2253 – Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names.

RFC 2252 – Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions

### *Exemplo de Entrada*

```
dn: cn=admin,dc=dominio,dc=br
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e2NyeXB0fWQ0MmJ1dU00RXJxZDI=
structuralObjectClass: organizationalRole
entryUUID: 41d4ea46-f1ba-102a-8cf2-2f3fd59223d5
creatorsName:
modifiersName:
createTimestamp: 20061016233156Z
modifyTimestamp: 20061016233156Z
entryCSN: 20061016233156Z#000001#00#000000
```

### *LDIF*

LDIF (Ldap Interchange format) é o formato usado para armazenar os dados contidos na base LDAP num arquivo texto. Proporcionando os seguintes usos:

- Incluir entrada(s)
- Alterar entrada(s)
- Importar/exportar entrada(s) no LDAP
- Backup e restauração
- Replicação da(s) base(s)

Especificação:

RFC 2849 – The LDAP Data Interchange Format (LDIF) - Technical Specification.

Nota: Os valores das entradas devem obedecer ao schema do Diretório.

### *Exemplo de LDIF*

```
dn: uid=jsilva, ou=Usuarios, dc=dominio, dc=br
objectClass: inetOrgPerson
objectClass: organizationalPerson
initials: JS
givenName: Silva
sn: Joao
telephoneNumber: +55(21)5555-5555
ou: Informatica
displayName: Joao da Silva
mail: jsilva@empresa.com.br
uid: jsilva
cn: Joao da Silva
title: Programador
```

# Aplicações

Vamos agora manter nosso foco nas aplicações do LDAP em ambiente de rede. Para efeito de estudo, vamos adotar a distribuição **Debian GNU/Linux** e o **OpenLDAP** como servidor de diretórios.

### *OpenLDAP*

Vamos considerar alguns apectos básicos de implementação do LDAP usando o OpenLDAP.

O arquivo de configuração do serviço OpenLDAP é o slapd.conf. Este contém as seguintes especificações:

- Os esquemas a serem usados
- Configurações diversas (log, suporte v2, etc)
- O database backend a ser usado (modulo, caminho, etc)
- O DN raiz da base
- O DN do administrador da base e sua senha
- As políticas de acesso

Nota: Para maiores informações sobre como instalar e configurar o OpenLDAP, veja a Nota Técnica **NT001/07**.

***Lista de schemas*** (/etc/ldap/schema)

collective.schema Collective attributes (experimental)

corba.schema Corba Object

core.schema OpenLDAP "core"

cosine.schema COSINE Pilot

dyngroup.schema Dynamic Group (experimental)

inetorgperson.schema InetOrgPerson

java.schema Java Object

misc.schema Miscellaneous Schema (experimental)

nis.schema Network Information Service (experimental)

openldap.schema OpenLDAP Project (FYI)

ppolicy.schema Password Policy Schema (work in progress)

### *Backends de armazenamento*

/usr/lib/ldap/back_*.so

bdb Berkeley DB transactional backend

config Slapd configuration backend

dnssrv DNS SRV backend

hdb Hierarchical variant of bdb backend

ldap Lightweight Directory Access Protocol (Proxy) backend

ldbm Lightweight DBM backend

ldif Lightweight Data Interchange Format backend

meta Meta Directory backend

monitor Monitor backend

passwd Provides read-only access to passwd(5)

perl Perl Programmable backend

shell Shell (extern program) backend

sql SQL Programmable backend

### *Slapd.conf – Schemas*

```
# This is the main slapd configuration file. See slapd.conf(5) for more
# info on the configuration options.
########################################################################
# Global Directives:
# Features to permit
#allow bind_v2
# Schema and objectClass definitions
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/inetorgperson.schema
```

## *Slapd.conf – Configurações Diversas*

```
# Schema check allows for forcing entries to
# match schemas for their objectClasses's
schemacheck on
# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile /var/run/slapd/slapd.pid
# List of arguments that were passed to the server
argsfile /var/run/slapd/slapd.args
# Read slapd.conf(5) for possible values
loglevel 0
# Where the dynamically loaded modules are stored
modulepath /usr/lib/ldap
moduleload back_bdb
# The maximum number of entries that is returned for a search operation
sizelimit 500
# The tool-threads parameter sets the actual amount of cpu's that is used
# for indexing.
tool-threads 1
########################################################################
# Specific Backend Directives for bdb:
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
backend bdb
checkpoint 512 30
########################################################################
# Specific Backend Directives for 'other':
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
#backend <other>
```

## *Slapd.conf – Database Backend*

```
########################################################################
# Specific Backend Directives for bdb:
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
backend bdb
checkpoint 512 30
########################################################################
# Specific Directives for database #1, of type bdb:
# Database specific directives apply to this databasse until another
# 'database' directive occurs
database bdb
```

## *Slapd.conf – Configurações da Base*

```
/etc/ldap/slapd.conf - (debian)
# The base of your directory in database #1
suffix "dc=dominio,dc=br"
# rootdn directive for specifying a superuser on the database. This is needed
# for syncrepl.
# rootdn "cn=admin,dc=dominio,dc=br"
# Where the database file are physically stored for database #1
directory "/var/lib/ldap"
# For the Debian package we use 2MB as default but be sure to update this
# value if you have plenty of RAM
dbconfig set_cachesize 0 2097152 0
# Sven Hartge reported that he had to set this value incredibly high
# to get slapd running at all. See http://bugs.debian.org/303057
# for more information.
# Number of objects that can be locked at the same time.
dbconfig set_lk_max_objects 1500
# Number of locks (both requested and granted)
dbconfig set_lk_max_locks 1500
# Number of lockers
dbconfig set_lk_max_lockers 1500
# Indexing options for database #1
index objectClass eq
# Save the time that the entry gets modified, for database #1
lastmod on
# Where to store the replica logs for database #1
# replogfile /var/lib/ldap/replog
```

## *Slapd.conf – ACL*

```
# The userPassword by default can be changed
# by the entry owning it if they are authenticated.
# Others should not be able to see it, except the
# admin entry below
# These access lines apply to database #1 only
access to attrs=userPassword,shadowLastChange
by dn="cn=admin,dc=dominio,dc=br" write
by anonymous auth
by self write
by * none
# Ensure read access to the base for things like
# supportedSASLMechanisms. Without this you may
# have problems with SASL not knowing what
# mechanisms are available and the like.
# Note that this is covered by the 'access to *'
# ACL below too but if you change that as people
# are wont to do you'll still need this if you
# want SASL (and possible other things) to work
# happily.
access to dn.base="" by * read
# The admin dn has full write access, everyone else
# can read everything.
access to *
by dn="cn=admin,dc=dominio,dc=br" write
by * read
```

### *OpenLDAP – ACLs*

ACL (Access Control List) é quem define quais as entradas e/ou atributos podem ser acessados e Quais os clientes que podem ou não terem acesso.

Exemplo:

```
access to attr=userPassword
by self write
by anonymous auth
by dn.base="cn=admin,dc=unirio,dc=br" write
by * none
access to *
by * read
```

### *OpenLDAP – Comandos (Servidor)*

| | |
|---|---|
| **slapacl** | Checa o acesso a lista de atributos |
| **slapauth** | Checa uma lista de string de ID's para autenticação |
| **slapd** | Daemon do Servidor LDAP |
| **slapindex** | Cria os índices |
| **slaptest** | Testa se o arquivo de configuração está com erro |
| **slapadd** | Adiciona entradas na base |
| **slapcat** | Dump da base inteira |
| **slapdn** | Checa uma lista de strings de DN baseado no schema |
| **slappasswd** | Utilitário de geração de senhas |

Parâmetros:

**-c** Continua mesmo se houver erro

**-d** Debug Level

**-f** Config File

**-l** LDIF File

**-v** Verboso

### *OpenLDAP – Comandos (Cliente)*

| | |
|---|---|
| **ldapsearch** | Faz pesquisa na base ldap |
| **ldapadd** | Adiciona entradas |
| **ldapdelete** | Deleta entradas |
| **ldapcompare** | Compara entradas |
| **ldapmodify** | Modifica valores dos atributos |
| **ldapmodrdn** | Renomeia uma entrada |
| **ldappasswd** | Muda a senha de uma entrada |
| **ldapwhoami** | Diz quem é você na base ldap |

Parâmetros:

**-b** BaseDN

**-D** BindDN

**-f** Arquivo ldif

**-h** Servidor LDAP

**-W** Pede senha para o usuario

**-x** Autenticação simples (sem ssl)

## *OpenLDAP – Replicação*

Este recurso do OpenLDAP é usado para distribuir uma ou mais bases para vários servidores espalhados pela rede, com o objetivo de diminuir a sobrecarga de acesso num único servidor.

Configuração de Replicação (Debian GNU/Linux):

1. No slapd.conf do master:

```
replica uri=ldap://ldap02.dominio.br:389 \
binddn="cn=Admin,o=dominio,c=br" \
bindmethod=simple credentials=secret
replogfile /var/lib/ldap/replog
```

2. No slapd.conf do slave:

```
updatedn "cn=Admin,o=dominio,c=br"
updateref "ldap://ldap01.dominio.br"
```

3. Pare o servidor master e slave:

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

4. Copie a base do Master para o Slave.

```
#slapcat > base.ldif (master)
#scp base.ldif root@slave: (master)
#slapadd -l base.ldif (slave)
#chown -R openldap:openldap /var/lib/ldap (slave)
```

5. Rode novamente o slapd nos 2 servidores:

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

## *LDAP – Quando usar?*

- A informação precisa estar disponível em várias plataformas ?
- Os dados serão acessados através de vários computadores e/ou aplicações?
- Os registros armazenados sofrem poucas mudanças (poucas alterações durante o dia)?
- A estrutura dos dados pode ser representada em banco de dados do tipo “flat”, ou seja, toda a informação de um item pode ser colocado em um único registro?

Nota: Qualquer informação que é do tipo fichário é ideal para ser usado

## *LDAP – Exemplos de Aplicações*

- Autenticação de Usuários(Samba, NIS,NFS,AFS, pGina,VOIP);
- Catálogo de Endereços (Phone Book, Address Book,VOIP);
- Configurações de pacotes de software (DNS, DHCP, OpenLDAP);
- Autenticação unificada de serviços (SMTP, Webmail, IMAP, POP, HTTPd, PROXY, jabber);
- Single Sign-On (+Kerberos);
- Certificados Digitais, Chave públicas e de segurança;
- Informações de infra-estrutura (NIS maps, email aliases, mail host);
- API's (C, C++, Java, PHP, Python, FreePascal, Ruby, Perl, etc...).

# Bibliografia

Malère, Luiz. LDAP Linux HOWTO. 2007. URL: http://en.tldp.org/HOWTO/LDAP-HOWTO.html (acessado em 18/03/2008)

Johner, Heinz; Brown,Larry; Hinner,Franz-Stefan; Reis,Wolfgang; Westman,Johan. Understanding LDAP. 1998. URL: http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf (acessado em 18/03/2008)