LDAP（轻量目录访问协议）

LDAP实现提供被称为目录服务的信息服务，可以看做是一张特殊的数据库系统。可以有效的解决众多网络服务的用户账户问题，规定了统一的身份信息数据库、身份认证机制和接口，实现了资源和信息的同一管理，保证了数据的一致性和完整性。LDAP存储需要从不同的地点读取，但是不需要经常更新的数据比较有效，例如：

1. 公司员工的电话号码簿和组织结构图
2. 客户的联系信息
3. 计算机管理需要的信息，包括NIS映射、email假名，等等
4. 软件包的配置信息
5. 公用证书和安全密匙

大多数的LDAP服务器都为读密集型的操作进行专门的优化，因此，从LDAP服务器中读取数据会比从专门的OLTP优化的关系型数据库中读取数据快一个数量级。

# 1、LDAP模型

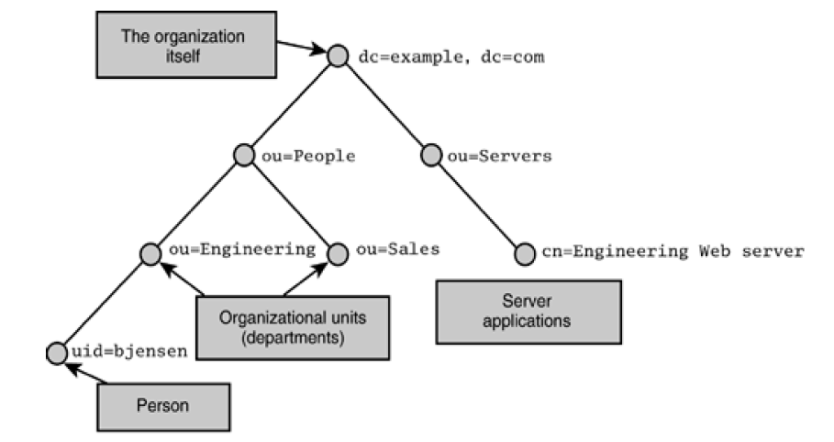
LDAP定义了四个模型，包括信息模型，命名模型，功能模型和安全模型，下面分别进行介绍。

## 1.1 信息模型

在LDAP中，信息以树状方式组织，基本数据单元是条目（相当于关系型数据中表的记录），而每个条目由属性构成，属性由类型（Type）和一个或多个值（Value）组成，相当于关系型数据库中的字段（Field，由字段名和数据类型组成）。

1）Entry

包含的信息描述了现实世界中的一个真实的对象，在目录系统中可以理解为一个节点。在目录中添加一个Entry时，该Entry必须属于一个或者多个对象类（Oject Class），Entry的类型由属性Object Class规定。每个Entry都有一个唯一的DN(distinguished name)来标识Entry在directory中的位置。



用Java的方式Entry相当于一个Instances，而Ojbect class自然就是Class。

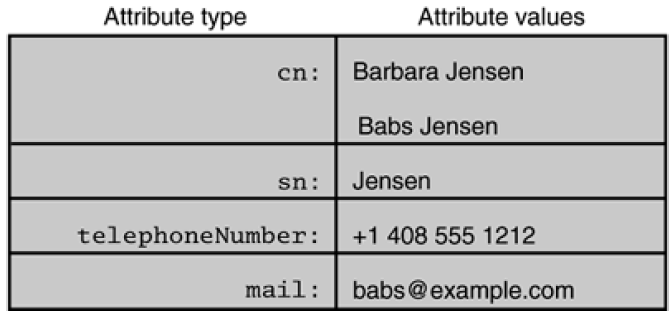
根节点DN的命名有多种方法，其中之一就是域名命名法，例如sohu.com根阶节点的DN应该是DN:dc=sohu,dc=com，People节点的DN:ou=People,dc=example,dc=com，RDN是目录树中节点的相对标识，例如People节点的RDN:ou=people。

LDAP目录服务器中存储的信息被组织成树型结构，在存储信息之前要定义模型，所有实现LDAP协议的目录服务器，都定义了很多模式。LDAP模式由以下一些元素组成：

* 1. Attribute types 属性类型，也就是属性名称。我们已经介绍过，每个属性名称其实也代表着一种属性类型。表示该属性可以存储什么样的信息。
  2. Attribute syntaxes 属性语法，该元素表示每个属性名称所存储的信息如何组织。
  3. 匹配规则，每一个属性都有匹配规则，用于数据的比较。
  4. object classes，对象类.上面已经介绍过，每个Entry都必须至少属于一个object class。规定了该Entry可以存储那些属性。

2）Attribute

每个Entry都是由许多Attribute组成，每个属性描述的是对象的一个特征，每个属性由一个类型和一个或多个值Value组成。



每个属性类型有所对应的语法和匹配规则；对象类和属性类型的定义均可以使用继承的概念。每个条目创建时，必须定义所属的对象类，必须提供对象类中的必选属性类型的属性值，在LDAP中一个属性类型可以对应多个值。

下面是常用的Attribute的含义：

|  |  |
| --- | --- |
| 名称 | 描述 |
| c | 国家 |
| cn | common name,一个对象的名字，如果指人，使用全名 |
| dc | domain component，常用来指一个域名的一部分 |
| givenName | 一个人的名字，不能用来指姓 |
| I | 地名，如一个城市或者其他地理区域的名字 |
| mail | 电子信箱地址 |
| o | 组织的名字 |
| ou | 一个组织单元的名字 |
| sn | Surname，一个人的姓 |
| TeleponeNumber | 电话号码 |
| uid | Userid，某个用户的登录名，与Linux系统中用户的uid不同 |

3）Object class

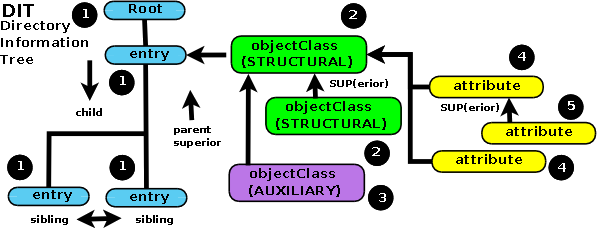
在LDAP中，一个条目必须包含一个Oject class属性，且需要赋予至少一个值。每个值将用作一条LDAP条目进行数据存储的模板，模板中包含了一个条目必须被赋值的属性和可选的属性。

Object分为三类，结构型（Structural），如Person和orginzationUnit，辅助型（Auxiliary），如extensibeObject，抽象型（Abstract）：如top，抽象型的ObjectClass不能直接使用。

下面部分常用的ObjectClass，定义在/etc/openldap/schema/core.schema文件中

|  |  |  |
| --- | --- | --- |
| 名称 | 描述 | 必要属性 |
| account |  | userid |
| alias |  |  |
| dcobject | 辅助型 |  |
| domain |  |  |
| ipHost |  |  |
| organization |  | o |
| organizationalRole |  | cn |
| organizationalUnit |  | ou |
| person |  | cn,sn |
| organizationPerson |  | cn,sn |
| inetOrgPerson |  |  |
| resientialPerson |  |  |
| posixAccount | Linux用户 | cn,gidNumber,homeDirectory  ,uid,uidNumber |
| posixGroup | Linux用户组 | cn,gidNumber |
| top | 抽象型，顶级ObjectClass |  |
| extensibeObject |  |  |

Entry的关系如下图所示：



Entry必须仅包含一个Structural类型的OjectClass，其他两种类型可包括0或者多个。Entries可以有子、父及兄弟entries，在图上描述。

## 1.2 LDAP命名模型

LDAP命名模型定义了如何在目录系统中组织数据以及如何从目录系统中查找数据，命名模型将Entry按类似倒立的树形结构进行规划，类似于UNIX系统的文件系统，下图是LDAP典型的命名模型：



LDAP目录结构中的root entry是一个特殊的Entry，包含了目录服务器的配置信息。目录中任何一个节点都可以包含信息，也都可以有子节点

在LDAP中可以定义一个别名Entry，指向另外一个Entry，如下图：



别名Entry的Object class必须是alias，其属性aliaseObjectName的值必须是Entry所指向的Entry的DN.

## 1.3 LDAP功能模型

描述LDAP协议可以采用的相关操作，来访问存储在目录树中的数据，可以将操作分成三组：

* 1. 更新操作 包括添加，删除，重命名，修改Entry
  2. Interrogation Operation 用于数据的查询
  3. 认证和控制 (bind unbind abandon)

## 1.4 LDAP安全模型

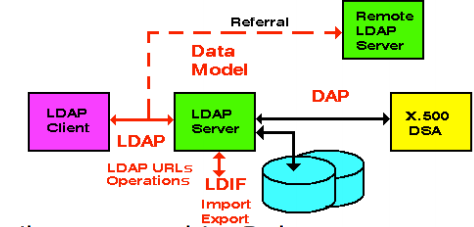
提供一个框架，包含目录中的信息不被非法访问，LDAP的安全模型主要是通过身份认证、安全通道和访问控制（ACL）来实现。

LDAP是一个面向连接的协议，在能够对LDAP目录进行任何操作之前，LDAP客户端端必须获得一个到LDAP服务端的一个连接，在这个过程中需要对LDAP客户端的身份进行验证（可以理解为用户绑定）。

在用户通过验证之后，为用户分配附加的权限，比如一些用户只能查看特定的Entry，而不能修改。一些用户可以查看并且修改所有的Entry等。

# 2、LDAP系统架构

LDAP是运行于TCP/IP之上的应用层协议。LDAP的目录服务功能建立在Client/Server模型之上，所有的目录信息数据存储在LDAP服务器中。一个或者多个LDAP服务器组成LDAP目录树，每个LDAP服务器由目录服务模块、复制服务模块和管理模块组成，如下图：



目录服务模块主要由前端部分负责通常的客户机与服务器之间的网络通讯，完成协议解析和分析。后端部分负责目录数据库的管理。

LDAP其工作流程如下：

1）客户机根据自身需求向LDAP服务器发送查询和操作请求

2）服务器负责对目录树中条目进行必要的操作

3）服务器向客户机返回一个应答，这个应答可能包含查询结果，或包含操作出错信息，或者一个引用。

4）客户机收到引用时，向更适宜的LDAP服务器发送请求

# 3、OpenLDAP安装和使用

## 3.1 OpenLDAP的安装

1）安装OpenLDAP

yum install -y openldap openldap-servers openldap-clients

2）配置LDAP

cp /usr/share/openldap-servers/slapd.conf.obsolete /etc/openldap/slapd.conf

cp /usr/share/openldap-servers/DB\_CONFIG.example /var/lib/ldap/DB\_CONFIG

通过vim修改slapd.conf，步骤如下：

设置目录树的后缀：suffix "dc=my-domain,dc=com"

将其改为：suffix "dc=example,dc=com"

该语句设置LDAP管理员的DN

找到语句：rootdn "cn=Manager,dc=my-domain,dc=com"

将其改为：rootdn "cn=Manager,dc=example,dc=com"

设置LDAP管理员的口令：

找到语句：rootpw secret

将其改为：rootpw 123123 //推荐用密文

3）修改文件及目录权限

chown ldap.ldap /etc/openldap/\*

chown ldap.ldap /var/lib/ldap/\*

4） LDAP启动

5）由于/etc/openldap/slapd.d目录中的数据是初始数据

删除这些数据

 rm -rf /etc/openldap/slapd.d/\*

重新生成配置数据

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

6）导入数据

# cat example.ldif

输入以下内容：

dn:dc=example,dc=com

objectclass:dcObject

objectclass:organization

o:Example,Inc.

dc:example

dn:cn=Manager,dc=example,dc=com

objectclass:organizationalRole

cn:Manager

ldapadd -x -D "cn=Manager,dc=example,dc=com" -W -f example.ldif

提示输入密码，至此LDAP配置完成

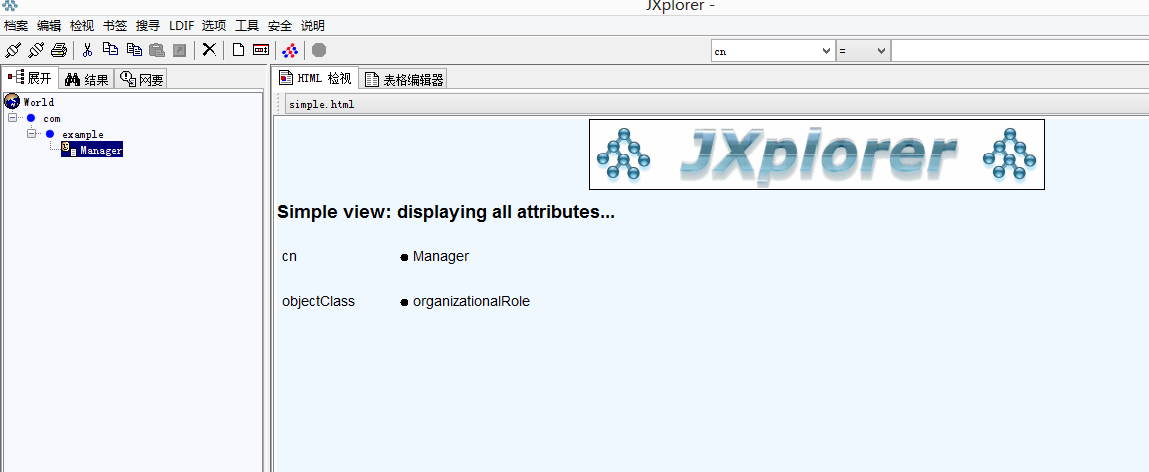
6）使用JXplore登录及查看LDAP数据

File -> connection

输入主机和端口



登录界面，如下图所示：



7）目录介绍

/usr/share/openldap-servers: 该目录有配置模板、DB\_CONFIG等模板

/etc/openldap/：该目录有Openldap的配置信息,schema等信息，其中

/var/lib/ldap: 放置了OpenLdap数据库及数据系统的配置

## 3.2 OpenLDAP的管理

### 3.2.1 添加数据

方式1）客户端管理工具ldapadd命令

该命令将一个LDIF文件的条目添加到目录，需要创建一个LDIF文件，然后再进行添加操作。LDIF的格式如下：

dn: Entry Names

attributeType: value

......

dn行类似于关系数据库中一条记录的关键字，不能与其他dn重复，一个LDIF文件可以包含多个条目，每个条目之间用空行隔开，例如：

dn: dc=dlw, dc=com

objectclass: top

objectclass: dcobject4: objectclass: organization

dc: dlw

o: dlw,Inc.

第1行dn定义该条目的标识；第2-4行定义该条目的object class，可以定义多个属性，条目的属性根据objectclass的不同而不同，有的objectclass有必须设置的属性。top没有必须定义的属性，dbobject必须定义属性dc，用来表示一个域名的部分，而organization必须定义属性o，用来表示一个组织的名称。根据objectClass的要求，第5,6行分别定义属性dc和属性o的值。根据下图创建LDIF文件example.com.ldif：



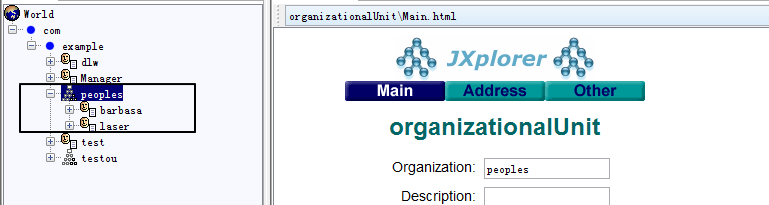
文件内容如下：



添加命令如下：

ldapadd -x -D "cn=Manager,dc=example,dc=com" -W -f dlw.com.ldif

运行结果如下图：



也可以通过程序代码向目录数据库中添加数据，下面会更具体的介绍。

### 3.2.2 查询Entry

添加到目录中的条目被保存在目录数据库，在Linux命令界面下，可使用客户端工具ldapsearch命令来进行查询，该命令格式如下：

ldapsearch  选项  过滤  属性值

常见的选项有以下几个：

-x :进行简单认证

-D：用来绑定服务器的dn

-w:绑定dn的密码

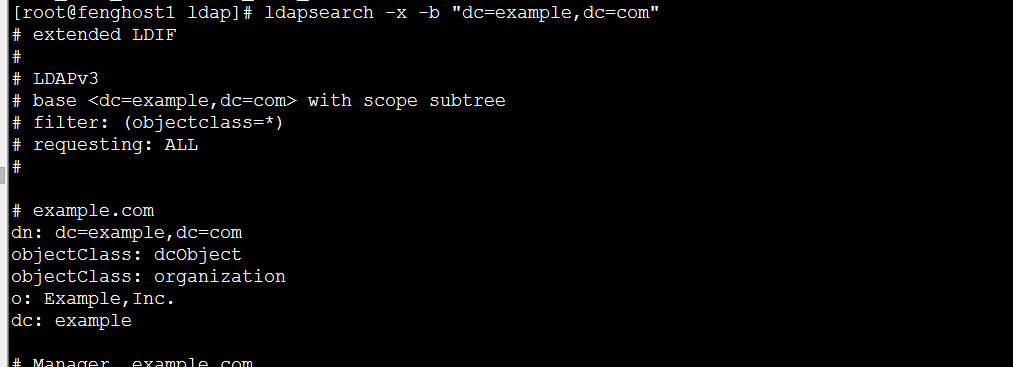
-b:指定要查询的根节点

-H:指定要查询的服务器

例如，下面的命令：

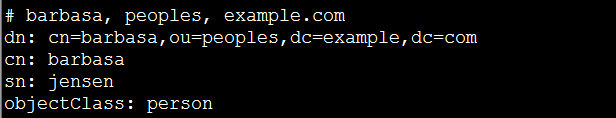
ldapsearch -x -b "dc=example,dc=com"

执行结果如下图（部分）：



而如果使用以下命令，查询显示sn中以字符jen开发的条目，将得到如下图的查询结果，只找到一个条目：

ldapsearch -x -b "dc=example,dc=com" 'sn=jen\*'



### 3.2.3 修改Entry

使用OpenLDAP客户端工具ldapmodify命令可对目录数据库中的条目进行修改，该命令的格式如下：

 ldapmodify  选项

选项与ldapsearch类似，不再介绍

1）交互式修改

修改前面创建的Entry,'cn=barbasa,ou=peoples,dc=example,dc=com'，将sn的属性修改为"Test Modify"，并添加一个description属性，设置其值为"add Attributed"

命令如下：

ldapmodify -x -D "cn=Manager,dc=example,dc=com" -W 123123

执行命令后，终端等候用户输入需要修改的条目dn，输入以下内容：

dn: cn=barbasa, ou=peoples, dc=example, dc=com

changetype: modify

replace: sn

sn: Test Modify

以上内容，第一行是查询需要修改的条目，第二行设置修改模式，第三行设置要替换的属性sn，第四行给sn重新设置一个值，替换该属性原有的值。

2）通过文件进行修改

创建修改文件，在其中输入以下内容：

dn: cn=barbasa,ou=people,dc=example,dc=com

changetype: modify

replace: sn

sn: Test User

执行命令：

ldapmodify -x -D "cn=Manager,dc=example,dc=com" -W -f modify

### 3.2.4 删除Entry

使用ldapdelete命令将其删除

ldapdelete  选项  删除条目

例如，删除目录数据库中的"cn=barbasa,ou=peoples,dc=example,dc=com"条目，具体命令如下：

ldapdelete -x -D "cn=Manager,dc=example,dc=com" -W \ "cn=barbasa,ou=peoples,dc=example,dc=com"

通过ldapsearch，

ldapsearch -x -b "dc=example,dc=com" 'sn=jen\*'

结果为空

### 3.2.5 数据导出

通过ldapadd命令可向目录数据库中添加数据，在某些情况下，需要进行反向操作，即将目录数据库中的数据导出。

使用ldapsearch命令对目录数据库进行搜索，可以通过重定向将搜索结果保存到一个文件中，达到导出数据的命令

导出数据更常用slapcat命令，命令格式如下：

slapcat -l export.ldif

注意：first database不运行使用slapcat命令，一般LDAP服务器多台，可设置一台为主服务器，其他的为从服务器。

LDAP错误总结：

http://blog.csdn.net/zdwzzu2006/article/details/8550910