## 1. 目录服务

目录是一个为查询、浏览和搜索而优化的专业分布式数据库，它呈树状结构组织数据，就好象Linux/Unix系统中的文件目录一样。目录数据库和关 系数据库不同，它有优异的读性能，但写性能差，并且没有事务处理、回滚等复杂功能，不适于存储修改频繁的数据。所以目录天生是用来查询的，就好象它的名字 一样。

目录服务是由目录数据库和一套访问协议组成的系统。类似以下的信息适合储存在目录中：

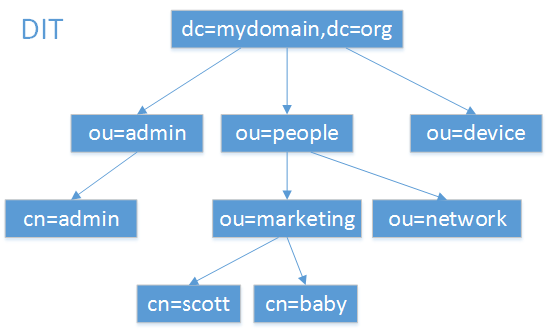
* 企业员工信息，如姓名、电话、邮箱等；
* 公用证书和安全密钥；
* 公司的物理设备信息，如服务器，它的IP地址、存放位置、厂商、购买时间等；

[LDAP](http://www.openldap.org/doc/admin24/intro.html" \t "https://blog.csdn.net/mawming/article/details/_blank)是轻量目录访问协议(Lightweight Directory Access Protocol)的缩写，LDAP是从X.500目录访问协议的基础上发展过来的，目前的版本是v3.0。与LDAP一样提供类似的目录服务软件还有[ApacheDS](http://directory.apache.org/apacheds/" \t "https://blog.csdn.net/mawming/article/details/_blank)、[Active Directory](https://technet.microsoft.com/en-us/library/dd448614.aspx" \t "https://blog.csdn.net/mawming/article/details/_blank)、[Red Hat Directory Service](http://www.redhat.com/en/technologies/cloud-computing/directory-server" \t "https://blog.csdn.net/mawming/article/details/_blank) 。

## 2. LDAP特点

* LDAP的结构用树来表示，而不是用表格。正因为这样，就不能用SQL语句了
* LDAP可以很快地得到查询结果，不过在写方面，就慢得多
* LDAP提供了静态数据的快速查询方式
* Client/server模型，Server 用于存储数据，Client提供操作目录信息树的工具
* 这些工具可以将数据库的内容以文本格式（LDAP 数据交换格式，LDIF）呈现在您的面前
* LDAP是一种开放Internet标准，LDAP协议是跨平台的Interent协议

## 3. LDAP组织数据的方式

[](http://sean-images.qiniudn.com/ldap_intro_dctree.png)

## 4. 基本概念

在浏览LDAP相关文档时经常会遇见一些概念，下面是常见概念的简单解释。

### 4.1 Entry

条目，也叫记录项，是LDAP中最基本的颗粒，就像字典中的词条，或者是数据库中的记录。通常对LDAP的添加、删除、更改、检索都是以条目为基本对象的。

dn：每一个条目都有一个唯一的标识名（distinguished Name ，DN），如上图中一个 dn：”cn=baby,ou=marketing,ou=people,dc=mydomain,dc=org” 。通过DN的层次型语法结构，可以方便地表示出条目在LDAP树中的位置，通常用于检索。

rdn：一般指dn逗号最左边的部分，如cn=baby。它与RootDN不同，RootDN通常与RootPW同时出现，特指管理LDAP中信息的最高权限用户。

Base DN：LDAP目录树的最顶部就是根，也就是所谓的“Base DN”，如”dc=mydomain,dc=org”。

### 4.2 Attribute

每个条目都可以有很多属性（Attribute），比如常见的人都有姓名、地址、电话等属性。每个属性都有名称及对应的值，属性值可以有单个、多个，比如你有多个邮箱。

属性不是随便定义的，需要符合一定的规则，而这个规则可以通过schema制定。比如，如果一个entry没有包含在 inetorgperson 这个 schema 中的objectClass: inetOrgPerson，那么就不能为它指定employeeNumber属性，因为employeeNumber是在inetOrgPerson中定义的。

LDAP为人员组织机构中常见的对象都设计了属性(比如commonName，surname)。下面有一些常用的别名：

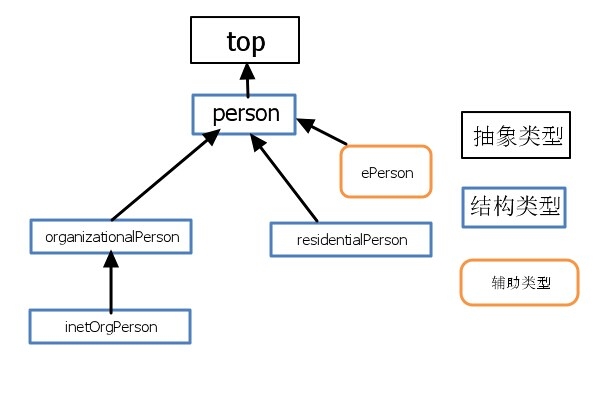
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **属性** | **别名** | **语法** | **描述** | **值(举例)** |
| commonName | cn | Directory String | 姓名 | sean |
| surname | [sn](https://www.baidu.com/s?wd=sn&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/mawming/article/details/_blank) | Directory String | 姓 | Chow |
| organizationalUnitName | ou | Directory String | 单位（部门）名称 | IT\_SECTION |
| organization | o | Directory String | 组织（公司）名称 | example |
| telephoneNumber |  | Telephone Number | 电话号码 | 110 |
| objectClass |  |  | 内置属性 | organizationalPerson |

### 4.3 ObjectClass

对象类是属性的集合，LDAP预想了很多人员组织机构中常见的对象，并将其封装成对象类。比如人员（person）含有姓（sn）、名（cn）、电 话(telephoneNumber)、密码(userPassword)等属性，单位职工(organizationalPerson)是人员 (person)的继承类，除了上述属性之外还含有职务（title）、邮政编码（postalCode）、通信地址(postalAddress)等属 性。

通过对象类可以方便的定义条目类型。每个条目可以直接继承多个对象类，这样就继承了各种属性。如果2个对象类中有相同的属性，则条目继承后只会保留 1个属性。对象类同时也规定了哪些属性是基本信息，必须含有(Must 活Required，必要属性)：哪些属性是扩展信息，可以含有（May或Optional，可选属性）。

对象类有三种类型：结构类型（Structural）、抽象类型(Abstract)和辅助类型（Auxiliary）。结构类型是最基本的类型， 它规定了对象实体的基本属性，每个条目属于且仅属于一个结构型对象类。抽象类型可以是结构类型或其他抽象类型父类，它将对象属性[中共](https://www.baidu.com/s?wd=%E4%B8%AD%E5%85%B1&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/mawming/article/details/_blank)性的部分组织在一起， 称为其他类的模板，条目不能直接集成抽象型对象类。辅助类型规定了对象实体的扩展属性。每个条目至少有一个结构性对象类。

对象类本身是可以相互继承的，所以对象类的根类是top抽象型对象类。以常用的人员类型为例，他们的继承关系：  
[](http://sean-images.qiniudn.com/ldap_objectclass.jpg)

下面是inetOrgPerson对象类的在schema中的定义，可以清楚的看到它的父类SUB和可选属性MAY、必要属性MUST(继承自organizationalPerson)，关于各属性的语法则在schema中的attributetype定义。

# inetOrgPerson

# The inetOrgPerson represents people who are associated with an

# organization in some way. It is a structural class and is derived

# from the organizationalPerson which is defined in X.521 [X521].

objectclass ( 2.16.840.1.113730.3.2.2

NAME 'inetOrgPerson'

DESC 'RFC2798: Internet Organizational Person'

SUP organizationalPerson

STRUCTURAL

MAY (

audio $ businessCategory $ carLicense $ departmentNumber $

displayName $ employeeNumber $ employeeType $ givenName $

homePhone $ homePostalAddress $ initials $ jpegPhoto $

labeledURI $ mail $ manager $ mobile $ o $ pager $

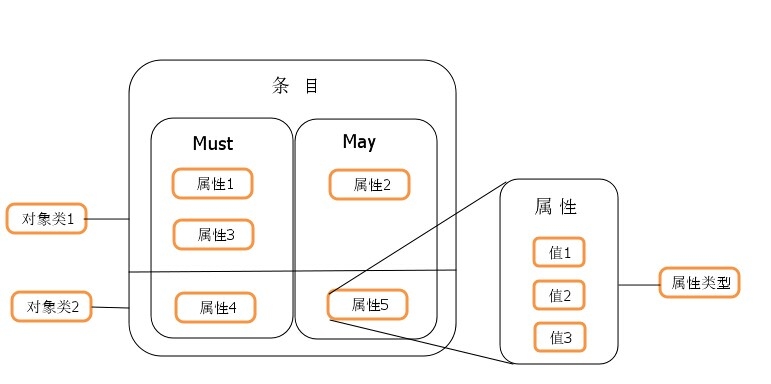
photo $ roomNumber $ secretary $ uid $ userCertificate $

x500uniqueIdentifier $ preferredLanguage $

userSMIMECertificate $ userPKCS12 )

)

### 4.4 Schema

对象类（ObjectClass）、属性类型（AttributeType）、语法（Syntax）分别约定了条目、属性、值，他们之间的关系如下 图所示。所以这些构成了模式(Schema)——对象类的集合。条目数据在导入时通常需要接受模式检查，它确保了目录中所有的条目数据结构都是一致的。  
[](http://sean-images.qiniudn.com/ldap_schema_attr_entry.jpg)

schema（一般在/etc/ldap/schema/目录）在导入时要注意前后顺序。

### 4.5 backend & database

ldap的后台进程slapd接收、响应请求，但实际存储数据、获取数据的操作是由Backends做的，而数据是存放在database中，所以你可以看到往往你可以看到backend和database指令是一样的值如 bdb 。一个 backend 可以有多个 database instance，但每个 database 的 suffix 和 rootdn 不一样。openldap 2.4版本的模块是动态加载的，所以在使用backend时需要moduleload back\_bdb指令。

bdb是一个高性能的支持事务和故障恢复的数据库后端，可以满足绝大部分需求。许多旧文档里（包括官方）说建议将bdb作为首选后端服务（primary backend），但[2.4版文档](http://www.openldap.org/doc/admin24/backends.html" \t "https://blog.csdn.net/mawming/article/details/_blank)明确说hdb才是被首先推荐使用的，这从 2.4.40 版默认安装后的配置文件里也可以看出。hdb是基于bdb的，但是它通过扩展的索引和缓存技术可以加快数据访问，修改entries会更有效率，有兴趣可以访问上的链接或[slapd.backends](http://linux.die.net/man/5/slapd.backends" \t "https://blog.csdn.net/mawming/article/details/_blank)。

另外config是特殊的backend，用来在运行时管理slapd的配置，它只能有一个实例，甚至无需显式在slapd.conf中配置。

### 4.6 TLS & SASL

分布式LDAP 是以明文的格式通过网络来发送信息的，包括client访问ldap的密码（当然一般密码已然是二进制的），SSL/TLS 的加密协议就是来保证数据传送的保密性和完整性。

SASL （Simple Authenticaion and Security Layer）简单身份验证安全框架，它能够实现openldap客户端到服务端的用户验证，也是ldapsearch、ldapmodify这些标准客户端工具默认尝试与LDAP服务端认证用户的方式（前提是已经安装好 [Cyrus SASL](http://www.openldap.org/doc/admin24/sasl.html" \t "https://blog.csdn.net/mawming/article/details/_blank)）。SASL有几大工业实现标准：Kerberos V5、DIGEST-MD5、EXTERNAL、PLAIN、LOGIN。

Kerberos V5是里面最复杂的一种，使用GSSAPI机制，必须配置完整的Kerberos V5安全系统，密码不再存放在[目录服务器](https://www.baidu.com/s?wd=%E7%9B%AE%E5%BD%95%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/mawming/article/details/_blank)中，每一个dn与Kerberos数据库的主体对应。DIGEST-MD5稍微简单一点，密码通过 saslpasswd2生成放在sasldb数据库中，或者将明文hash存到LDAP dn的userPassword中，每一个authid映射成目录服务器的dn，常和SSL配合使用。参考[将 LDAP 客户端配置为使用安全性](http://docs.oracle.com/cd/E19957-01/820-0293/6nc1tbp0h/index.html" \t "https://blog.csdn.net/mawming/article/details/_blank)

EXTERNAL一般用于初始化添加schema时使用，如ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/core.ldif。

### 4.7 LDIF

LDIF（LDAP Data Interchange Format，数据交换格式）是LDAP数据库信息的一种文本格式，用于数据的导入导出，每行都是“属性: 值”对，见 [openldap ldif格式示例](http://seanlook.com/2015/01/22/openldap_ldif_example/" \t "https://blog.csdn.net/mawming/article/details/_blank)

OpenLDAP(2.4.3x)服务器安装配置方法见[这里](http://seanlook.com/2015/01/21/openldap-install-guide-ssl/" \t "https://blog.csdn.net/mawming/article/details/_blank)。

**参考**

* [LDAP基础概念](http://407711169.blog.51cto.com/6616996/1439623" \t "https://blog.csdn.net/mawming/article/details/_blank)
* [LDAP-HOWTO](http://www.tldp.org/HOWTO/LDAP-HOWTO/ldapbackends.html" \t "https://blog.csdn.net/mawming/article/details/_blank)
* [openldap doc admin24](http://www.openldap.org/doc/admin24/" \t "https://blog.csdn.net/mawming/article/details/_blank)

原文链接地址：[http://seanlook.com/2015/01/15/openldap\_introduction/](http://seanlook.com/2015/01/15/openldap_introduction/" \t "https://blog.csdn.net/mawming/article/details/_blank)