LDAPの基礎知識

ＤＩＴとエントリ

LDAPとは、ユーザ情報やコンピュータ、あるいはアプリケーションソフトなど、ネットワークを構成するいろいろなものを管理する特殊なデータベースのことを指します。あるいは、その情報にアクセスする手順のことを指す場合もあります。ファイルシステムのディレクトリと同様に、階層構造でデータを格納しています。

データの階層構造全体のことをＤＩＴ（Directory Information Tree）といいます。

ＤＩＴを構成する１つ１つのデータのまとまりをエントリ（Entry）といいます。エントリは、リレーショナルデータベース（ＲＤＢ）のレコードに相当するもので、属性と属性値のペアとしてデータを格納しています。

ファイルシステムでのディレクトリとの違い：

	ファイルシステム	LDAP
名前（識別子）	ディレクトリ名、ファイル名	DN（識別名）：属性と属性値のペアのうち、同じ階層内で一意に識別できるものを１つ選び（これをRDNという）、それに上位階層のRDNを続けたもの（これをDNという）をもって識別する。DNには上位階層のRDNがすべて含まれるので、DNが分かれば、DIT内のエントリ格納位置がすぐに分かる。
データの入れ物	ディレクトリ＝入れ物 ファイル＝中身	エントリは入れ物でもあり、中身でもある。DNによってディレクトリの階層を表し、その他の属性によってデータの内容を表す。図の「o=sales」などにも、その組織自身の情報を格納できる。

エントリの特徴：

- １つの属性に複数の属性値を格納できる。（SingleValue指定の属性はのぞく）
- 言語タグをつけて多国語の値を格納できる。日本語の場合は、lang-jaで使えるようになる。
- 属性ごとにあらかじめデータ型（文法）が決められていて、合致するデータのみ格納できる。

オブジェクトクラス

LDAPの属性にはたくさんの種類があるため、いくつかの属性をお互いに関連した１つのまとまりとして扱える仕組みがあります。それぞれのまとまりをオブジェクトクラスと言います。オブジェクトクラスでは、どのような属性がそこに含まれるのかが定義されています。またそれに加えて、それぞれの属性が必須（＝エントリに含まれなければならない）なのかオプション（＝エントリに含まれても含まれなくても良い）なのかも決められています。

LDAPのエントリでは、属性を単純に羅列するのではなく、まずそのエントリがどのようなオブジェクトクラスを持つのかを決め、それに沿った属性を必要に応じて格納していきます。

LDAPデータの扱い方

DITからエントリのデータを取り出したり、あるいは内容を変更したりするには、LDAPサーバへそれぞれの要求を送ります。主にどのような操作が可能かを以下にまとめます。

操作	説明
バインド	LDAPを操作するとき、最初に行う。
追加	新規エントリを追加する。
更新	エントリの属性値を追加・更新・削除する。
削除	エントリを削除する。
検索	条件に一致するエントリを検索する。
アンバインド	LDAPの操作を終了するときに行う。

バインドとは：
LDAPを操作するときは、アクセスする人が誰なのか、LDAPに対して事前に認証をしておく必要があります。この認証のことをバインドといいます。
事前に認証しておくことで、アクセスしてきた人ごとにアクセス権限を変えるなどの制御（アクセス制御）ができるようになっています。通常の認証では、「ユーザ名」と「パスワード」を使って行いますが、LDAPの場合は以下のようになります。

「ユーザ名」の代わりに「バインドDN」を指定します。バインドDNとはユーザの代わりになるようなLDAPのエントリのことで、あらかじめ作成しておきます。ここに「userPassword」属性を設定しておき、認証時のパスワードにはこれを指定します。指定したバインドDNが存在し、そのuserPassword属性値が合致した場合に認証ＯＫとなります。

認証が完了すると、以降はそのエントリがもつアクセス権限の範囲でLDAPにアクセスできるようになります。たとえば管理者のエントリでバインドした場合は、ほとんどのデータの読み書きが可能となり、また一般ユーザのエントリでバインドすれば、自分自身のエントリについては読み書き可能で他は読み込みのみ、などとなります。 その他、匿名バインドと言って、ユーザ名やパスワードを特に指定せずにバインドする方法もあります。この場合はたいてい、アクセス権がかなり限定されます。

具体的にどのようなアクセス権限を持たせるかは、別途定義することができます。

検索について：
検索条件としては以下のものあります。

- 属性値が指定した条件のものと一致するかどうか。
- 属性値の一部が指定した条件のものと一致するかどうか。
- 指定した属性が存在するかどうか。

これらをANDやORで組み合わせて、細かい検索条件を指定できます。このようにして指定した検索条件のことを検索フィルタといいます。
また、どの位置を起点として検索するか（検索ベース）、どの範囲で検索するか（検索スコープ）も指定します。検索ベースは、起点とするエントリのDNを、検索スコープは以下の３種類のうちから１つを選択します。

スコープ	説明
ベース(base)	検索ベースのエントリのみを検索対象とする。（つまりベースに指定した１つのエントリが条件に合致するかどうかを判定するだけ）
1レベル(one level)	ベースエントリの直下のエントリだけを検索対象とする。（ベースエントリは含まれない）
サブツリー(subtree)	ベースエントリとその配下のツリー全体を検索対象にする。

検索の目的に応じてスコープを指定します。

LDAPサーバとのやり取り

LDAPのデータを操作するにはLDAPサーバとのやり取りが必要です。サーバとのやり取りは、そのサーバが動いているマシン上で行う必要は無く、ネットワークを経由してLDAPサーバと通信することもできます。また、LDAPサーバとやり取りする方法（インタフェース）もいろいろと種類があります。

インタフェース	説明
LDAPクライアントコマンド	コマンドラインから入力する。ldapadd,ldapsearch等。管理者が手動で操作するのに向いている。
PHPのLDAP関数	phpの関数からLDAPサーバを呼び出す。ホームページからLDAPにアクセスするのに向いている。
perlのLDAP関数	perlの関数からLDAPサーバを呼び出す。
Java,C、C#のLDAP関連クラスや関数	JavaやC言語のクラスや関数からLDAPサーバを呼び出す。LDAPクライアントコマンドはこれらを使ってプログラミングされている。
その他	PAM（UNIXの共通認証用モジュール）、SQLなどのデータベースサーバ（？）

LDIFファイル

LDAPのエントリはLDIFというテキスト形式でも扱えます。以下は、あるエントリのLDIFです。

```
dn: cn=admin,ou=users,o=kogaku,dc=kyoto-u,dc=ac,dc=jp
objectClass: applicationProcess
objectClass: shadowAccount
cn: admin
uid: admin
userPassword: {SMD5}jgiChd/iWn+VFOJ4YHBOEJSHtAU=
description: administrator o=kogaku
description;lang-ja: o=kogaku 配下の管理者
```

「属性名:属性値」というフォーマットで記述します。エントリの最初には必ずdn属性(DN)を置き

ます。LDIFはデータをまとめて処理する場合に便利です。LDIFで記述したファイルを指定して、エントリを一括追加したり削除したりできます。なお、例の最下行のように「;lang-ja」という「言語タグ」をつけると、全角文字のデータを扱うことができます。LDAPの仕様として、言語タグをつけた属性値はBASE64でエンコードする必要がありますが、openLDAPの仕様として、UTF8でLDIFファイルが記述されていれば、文字列をそのまま記述してもOKとなっています。

LDAPの設計

LDAPサーバの設計ポイントは主に以下の３点となります。

[格納する情報の種類・属性]
個人に関する情報・組織の情報等、どのような種類のデータを持たせるか。（RDBの「テーブル」の種類に相当）また、各情報を構成する個々の属性としてどのようなものを格納するか。

[ツリーの構造]
LDAPはディレクトリ構造（ツリー）でデータ（エントリ）を格納するが、その構造は自由に定義できる。どのような構造とするか。

[アクセス制御]
適切なアクセス制御を設定してデータを保護する必要がある。どのようなアクセス制御の設定とするか。特に書き込みやパスワードの読み取りに関しては、アクセス制御をきちんと検討する必要がある。