2008年12月3日（水） 13時30分～16時30分

LPICレベル3技術解説セミナー
「301 Core Exam」受験のための勉強法
OpenLDAP編

オープンソース・ソリューション・テクノロジ株式会社

代表取締役 チーフアーキテクト 小田切耕司

OSSTech

お問い合わせ info@osstech.co.jp

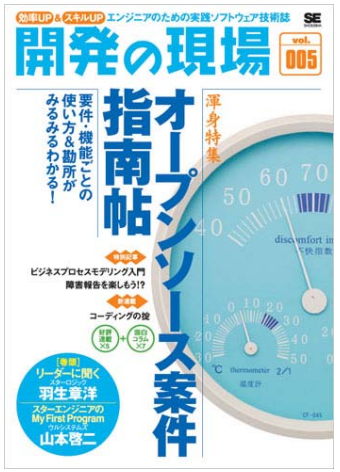
目　次

1. 講師紹介、OSSTech社紹介
2. LDAP概念と設計入門
3. LDAP構築／設定入門
4. LPIC例題解説
5. やってはいけないOpenLDAPサーバ構築
6. LPIC勉強法:OpenLDAP編まとめ

講師紹介
オープンソース・ソリューション・テクノロジ
会社紹介

講師著作紹介

- @IT やってはいけないSambaサーバ構築：2008年版
- 日経コミュニケーション2007年11/15号から3回連載
  Windows管理者に送るSamba活用の道しるべ
- 技術評論社 Software Design 2006年7月号
  - ネットワーク運用／管理 五輪書（ごりんのしょ）
  - 「壱：地の巻」Sambaファイルサーバ
  - http://www.gihyo.co.jp/magazines/SD/contents/200607
- 2006年5月 翔泳社 開発の現場 vol.005
  - オープンソース案件指南帖
  - 総論編：オープンソースの基礎知識
  - http://www.shoeisha.com/mag/kaihatsu/
- 2006年5月 技術評論社 LDAP Super Expert
  - 巻頭企画
  - [新規／移行]LDAPディレクトリサービス導入計画
  - http://www.gihyo.co.jp/magazines/ldap-se
- 2006年5月 IDG月刊Windows Server World 2006年3月、4月号
  - 3月号：Shall we Samba?【お手軽導入編】
  - 4月号：Shall We Samba?【超本格運用編】
- 2005年10月 日経BP社 セキュアなSambaサーバの作り方
  - http://itpro.nikkeibp.co.jp/linux/extra/mook/mook12/index.shtml



オープンソース・ソリューション・テクノロジ株式会社

- 2006年9月に設立
- **OSに依存しないOSSのソリューションを中心に提供**
  - Linuxだけでなく、SolarisやFreeBSDへも対応！
- **Samba、LDAPなどによる認証統合ソリューションを提供**
  - 製品パッケージ提供
  - 製品サポート提供
  - 技術コンサルティング提供

http://www.osstech.co.jp

会社概要

会 社 名	オープンソース・ソリューション・テクノロジ株式会社	所属団体等	・Linuxコンソーシアム 理事 ・LPI-Japanビジネス・パートナー
英語表記	Open Source Solution Technology Corporation		
社名略称	OSSTech（オーエスエステック）または OSSテクノロジ		
業務内容	・ソフトウエアの企画、開発、販売およびサポート ・システムの導入に関するコンサルティング ・ソフトウエアに関する教育、研修、支援	主要取引先およびパートナー様	・デル(株) ・(株)野村総合研究所 ・サン・マイクロシステムズ(株) ・キヤノンITソリューションズ(株) ・(株)バッファロー ・(株)大塚商会 ・日本電信電話(株) ・日本電気(株) ・伊藤忠テクノソリューションズ(株) ・新日鉄ソリューションズ(株) ・(株)日立システムアンドサービス ・ミラクル・リナックス株式会社
役 員	代表取締役　　小田切 耕司 技術取締役　　武田 保真		
オフィス	〒141-0022 東京都品川区東五反田1-10-7 アイオス五反田ビル Tel & FAX : 03-5422-9373		
Webページ	http://www.osstech.co.jp/		
設立	2006年9月		
資本金	1080万円		

LPIC-3 Core「301 Core Exam」: 出題範囲

- 主題 301: 概念、アーキテクチャおよび設計
- 主題 302: インストールおよび開発
- 主題 303: 設定
- 主題 304: 使用法(運用について)
- 主題 305: 統合と移行
- 主題 306: キャパシティプランニング

Part 1.
LDAP概念と設計入門

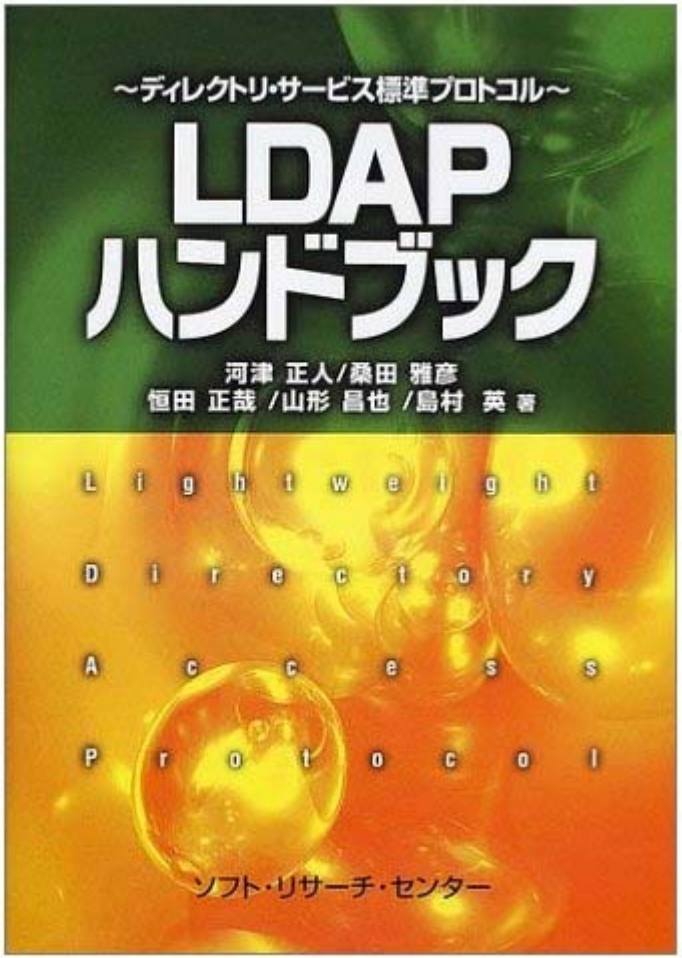
OSSTech

主題 301: 概念　LDAPとは?

- ディレクトリサービスを利用するための規約の1つ(RFCで定義)
  - ディレクトリサービスとは、キーを基に関連情報を取り出す仕組み
  - ユーザ管理、電話帳、リソース管理などに利用
  - 高機能だが運用負荷や開発コストが高かったITU- T 勧告のX.500 ディレクトリ・サービスを「90 %の機能を10 %のコストで実現する」ために設計
- 商用LDAP製品も多数存在
  - Sun Java Directory Server, Red Hat Directory Server, Novell eDirectoryなど
  - MS Active DirectoryもLDAP準拠(認証はKerberos)
- オープンソースソフト
  - OpenLDAP
    - Linux ディストリビューションに同梱されるオープンソースのLDAP
  - Red Hat / Fedora Directory Server
    - かつてのNetscape Directory ServerをOSSにしたもの(RHは有償、Fedoraは無償)
  - Apache Directory Server
    - Apacheプロジェクトが進めるJavaで書かれたDS

LDAP概念を勉強のための参考書

- LDAPハンドブック
  - ディレクトリ・サービス標準プロトコル
  - 出版社: ソフトリサーチセンター (2002/03)
  - 発売日： 2002/03



LDAPとRDBMSの違い

- LDAPはネットワークプロトコル、SQLは言語

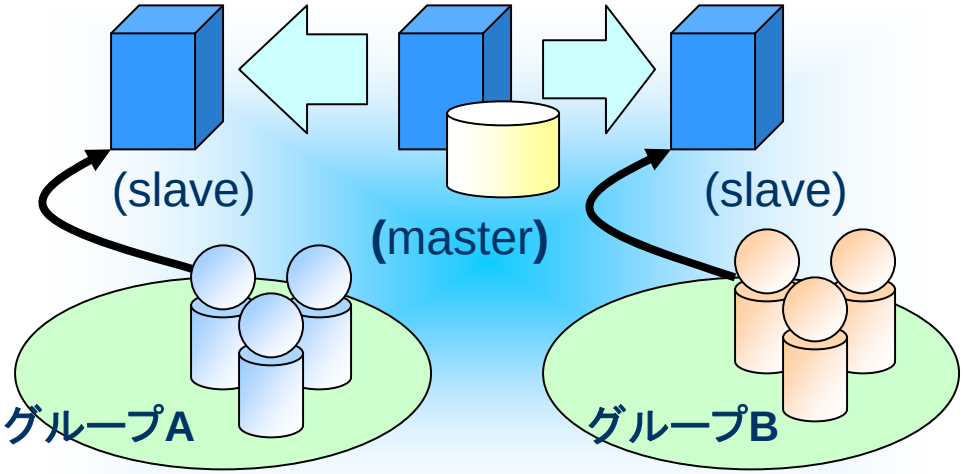
	LDAP	RDBMS
用途	検索性能重視、頻繁な更新には向かない	検索だけでなく頻繁な更新も重視
構造	木構造（行や列といった概念はない）	表構造（行や列が存在）
スキーマ	既存の登録済みスキーマ(ObjectClass)を利用するのが一般的	ユーザが業務に合わせて個別に設計し、利用する
更新	トランザクションの概念はない （トランザクション機能を持った製品もある） 大量更新には向かないので1時間に数件といった更新頻度のものに利用する	トランザクションの概念あり 1秒間に何十、何百もの更新に耐えられる設計となっている
分散	ツリーの枝単位で分散配置が可能	キーの範囲で分散配置が可能
操作	LDAP（ネットワークプロトコル）で操作 プロトコルは単純	SQL（プログラム言語）で操作 複雑な操作が可能
検索手法	木の枝葉をたどるイメージ	表の行を走査するイメージ

LDAP概念に関する勘違い

- RDBMSは永続的なユーザ情報を蓄えるために使う、LDAPは管理情報を集約するために使う
（社員DBはRDMS、全社認証システムはLDAP）
- LDAPは検索重視となっているが、RDBより必ずしも早いわけではない
- LDAPはスケールアウト型負荷分散がやりやすいから
- 更新がすぐに反映されるとは限らない
  - ユーザ追加やパスワード変更がすぐにされないことがある(だからWindowsはパスワードをキャッシュする)
- マルチマスターの利用は要注意
  - トランザクションやロックの概念が弱い
  - uid,gidの自動割り振りをLDAPでやると危険

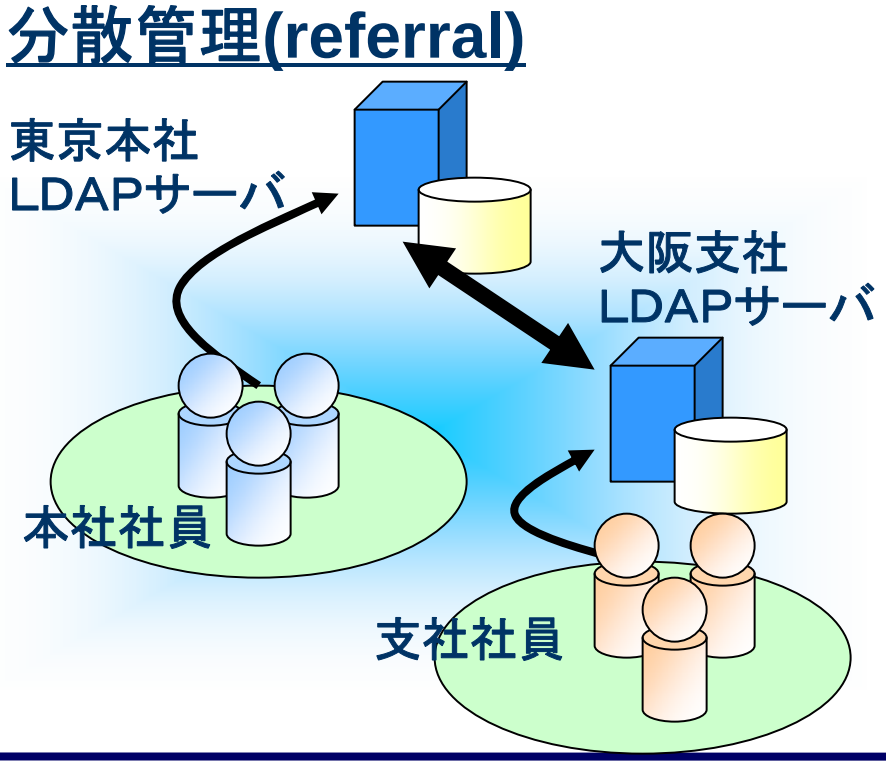
負荷分散方法1:レプリケーション

- 同じ内容のサーバを複数用意する
  - サーバを増やすだけでスケールアウトする
  - 負荷分散装置やldap.confで負荷を分散
  - 1つのサーバが持つデータ量は同じなので規模が大きくなると更新性能が低下
  - Syncreplではサブツリーだけを複製することも可能



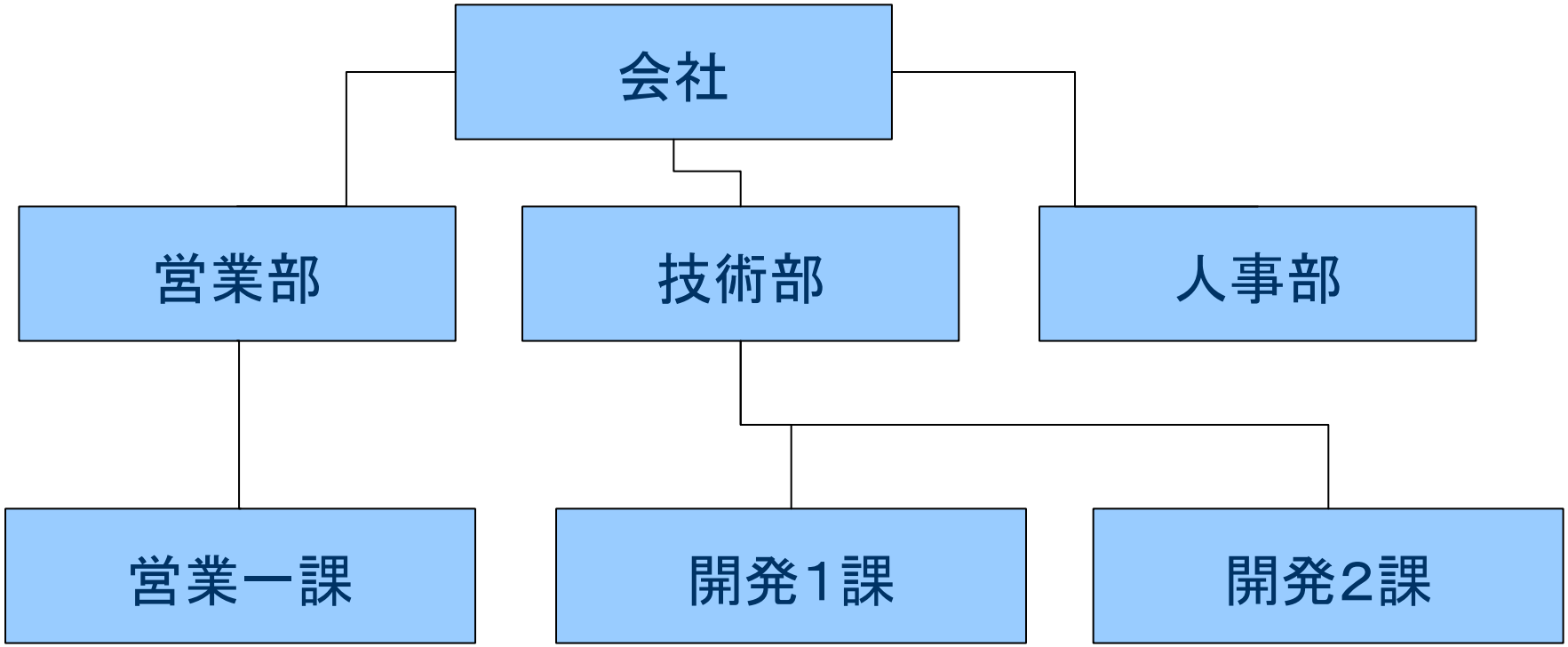
負荷分散方法2:リファラル

- サブツリー単位でサーバを分散する
  - ldap.confでbaseツリーを変える(負荷分散というよりも管理分散)
  - 1サーバがもつデータ量が減るので更新性能も上がる
  - referralが返ったら別なサーバを見に行くのはプログラム側の責任



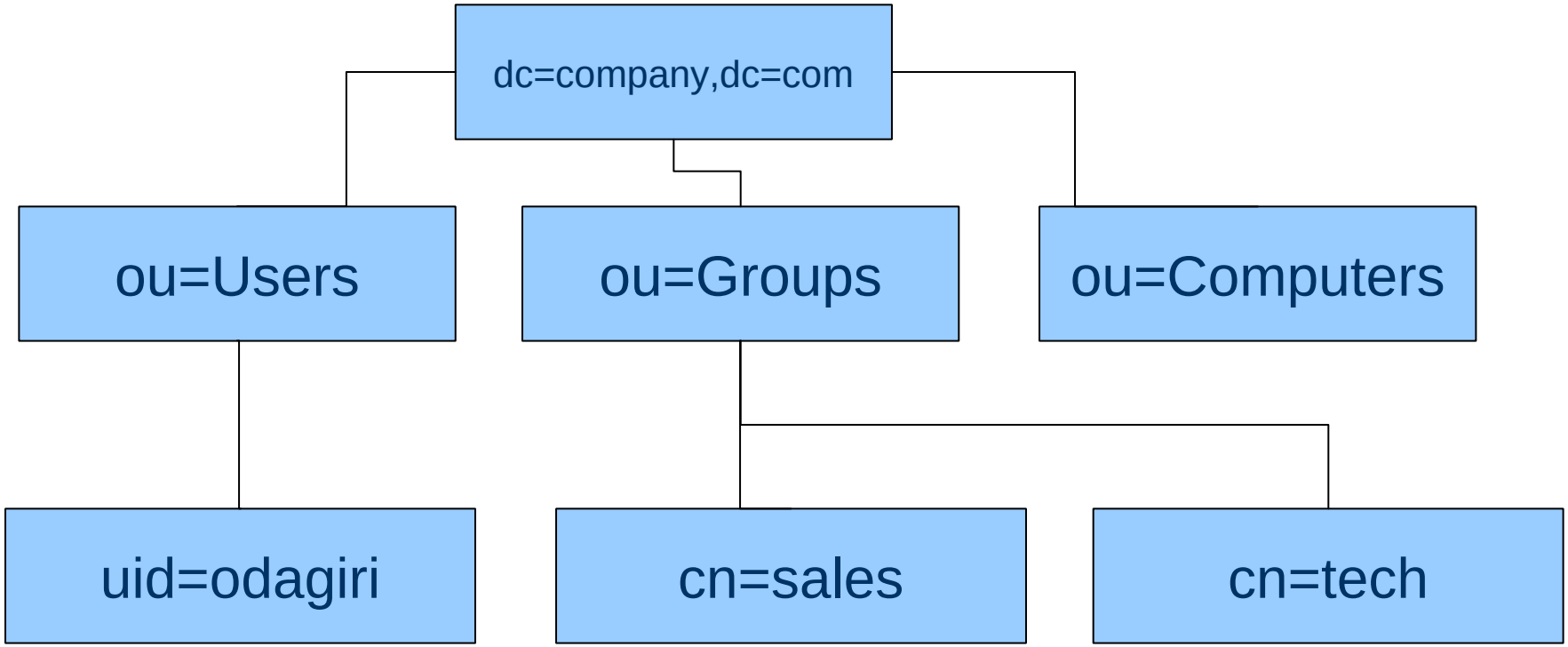
DIT(Directory information Tree)の概念

- 概念として組織構造をあげる書籍が多いが．．．



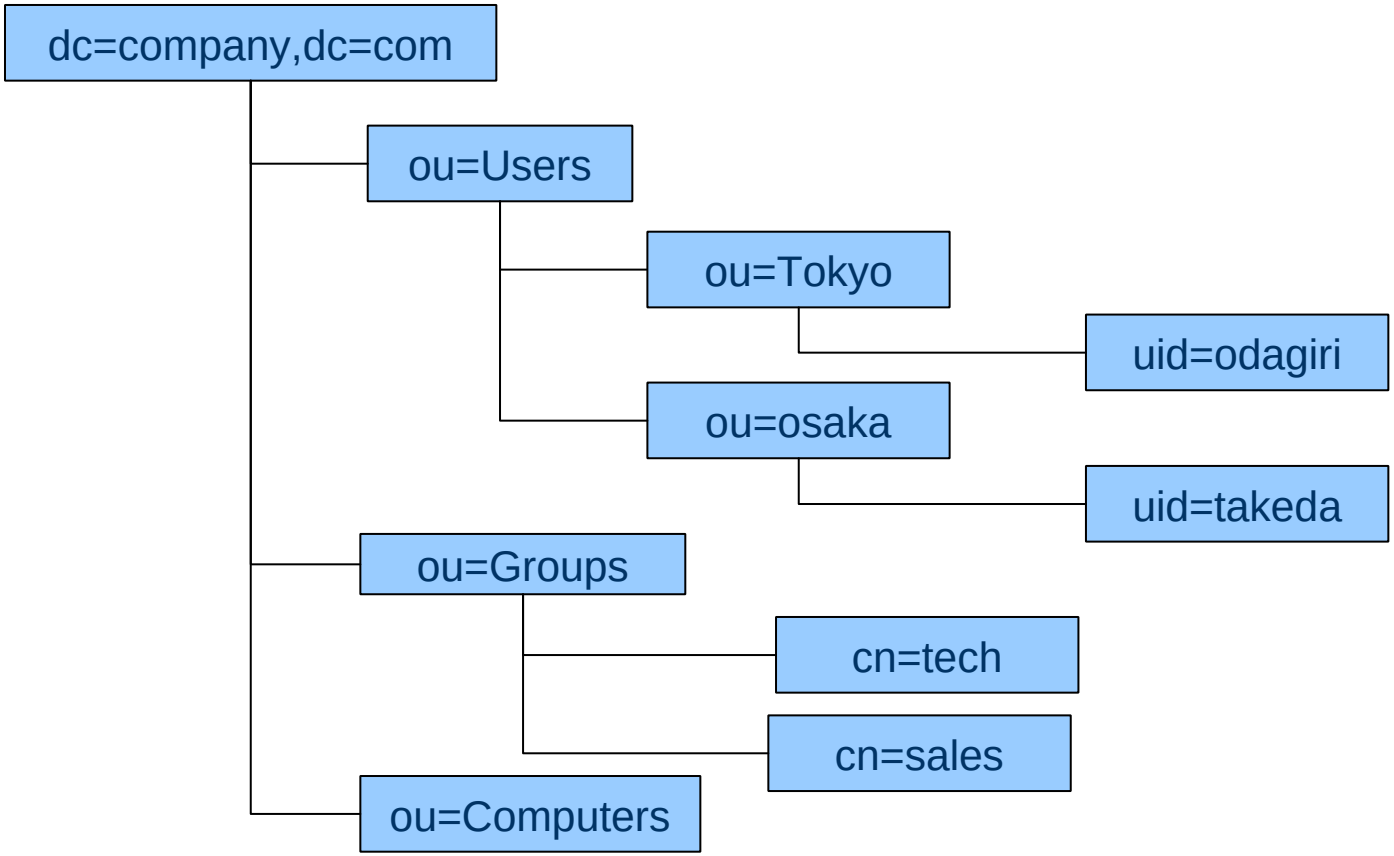
DIT(Directory information Tree)の概念

- 実構造としては管理単位で分ける



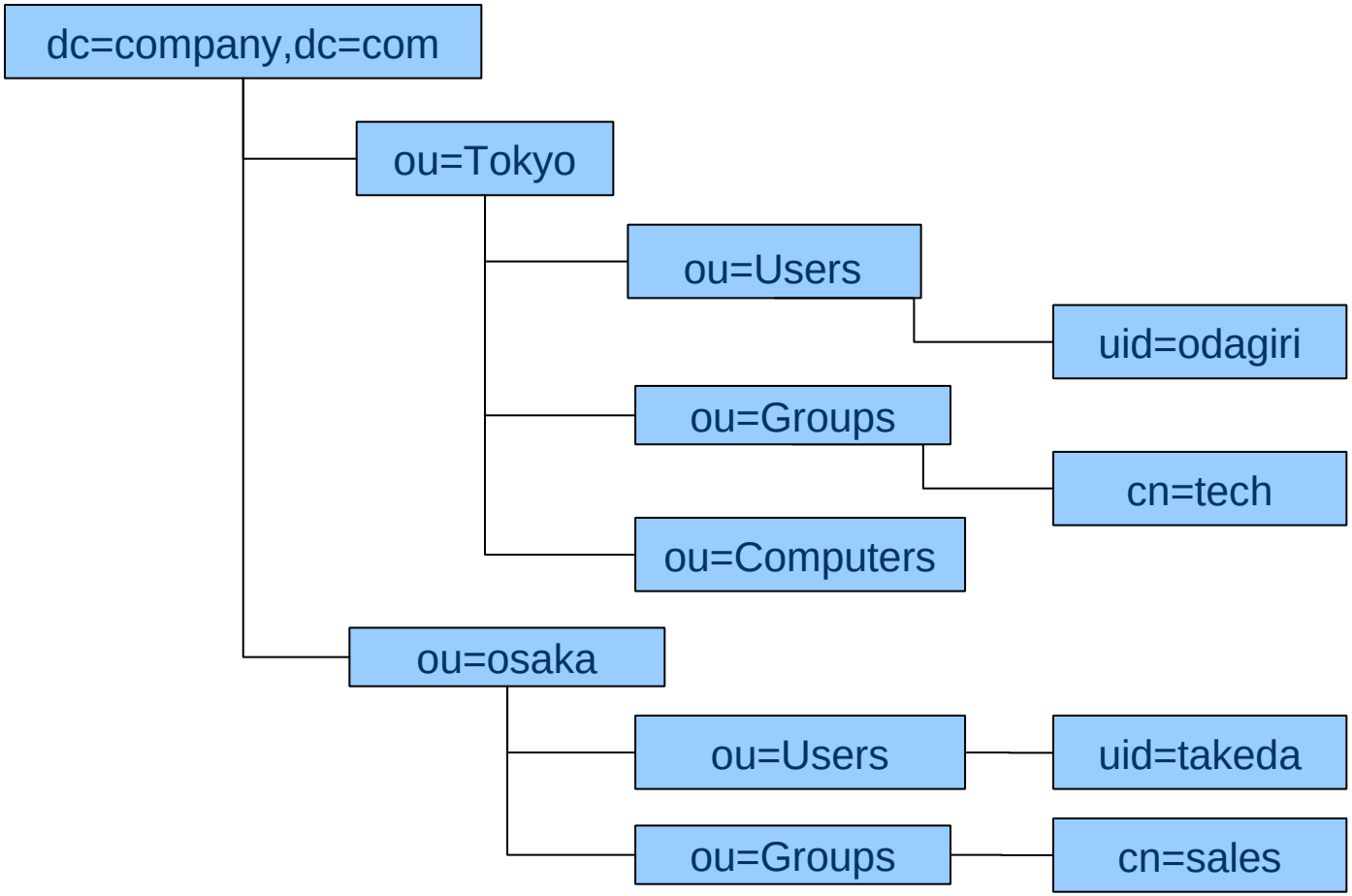
DIT(Directory information Tree)の設計

- 組織構造にマッピングしないこと、管理対象で分ける



DIT(Directory information Tree)の設計

- 組織構造にマッピングしないこと、管理対象で分ける



LDAPで何ができるか？

- Linuxユーザの統合管理
  (Mail,FTP,Telnet,Proxy,sshなど)
- Samba/Windowsユーザの統合管理
- Webサーバ(Apache)のアクセス制御
- 電話帳、メールアドレス帳
- PKI(公開キー)の保管場所として

- LDAPのスキーマはむやみに拡張しない
  本当に必要か精査する

OpenLDAPが標準で提供するスキーマ(1)

- 標準提供のスキーマを見ればLDAP何ができるかわかる
- core.schema
  - LDAPの核となるスキーマ、以下のRFCで定義されたスキーマが定義されている。
    - RFC 2252/2256 (LDAPv3)
    - RFC 1274 (uid/dc)
    - RFC 2079 (URI)
    - RFC 2247 (dc/dcObject)
    - RFC 2587 (PKI)
    - RFC 2589 (Dynamic Directory Services)
    - RFC 2377 (uidObject)
  - これだけでは何もできないが、CNやOUなど他のスキーマを使うための基本部分が定義されている。
- cosine.schema
  - X.500やX.400で規定されたアトリビュートなど以下のようなものが定義されている。
    - RFC1274で定義されるhost,manager, documentIdentifierなど
    - DNSレコードであるAレコード、MXレコード、NXレコード、SOAレコード、CNAMEレコード
  - これらからDNSレコードの格納先としてLDAPサービスが利用できることがわかる。

OpenLDAPが標準で提供するスキーマ(2)

- inetorgperson.schema
  - インターネット、特にメールアドレス帳のためのスキーマで、以下のようなものが定義される。
    - メールアドレス、社員番号、オフィスと自宅住所、会社と自宅の電話番号、写真、
- misc.schema
  - mailLocalAddressやnisMailAliasなどメールサーバが使うスキーマが定義される。
- nis.schema
  - posixAccountやposixGroupなどLinux/UNIXのユーザ認証統合に必須なスキーマが定義される。
  - NISをLDAPに置き換えるのに必要なスキーマも定義されている。
- samba.schema
  - このスキーマはOpenLDAPではなく、Sambaパッケージによって提供されるが、Sambaを使ってWindows/Linux/UNIXのユーザ認証統合に必須なスキーマが定義される。
  - WindowsドメインをSambaに置き換えるのに必要なスキーマも定義されている。
- java.schema
  - javaClassName, javaCodebaseなどJava Object (RFC 2713)を扱うためのスキーマが定義される。
- corba.schema
  - corbaIor、corbaRepositoryIdなどCorba Object (RFC 2714) を扱うためのスキーマが定義される。

アドレス帳の設定例

dn: uid=ユーザ名,ou=Users,dc=ドメイン名,dc=co,dc=jp
objectClass: posixAccount
objectClass: inetOrgPerson
cn: ユーザ名
sn: 名字
givenname: 名前
mail: メールアドレス
o: 会社名
ou: 所属
title: 役職
employeeNumber: 社員番号
telephoneNumber: 電話番号
facsimileTelephoneNumber: FAX番号
mobile: 携帯電話
st: 都道府県
l: 市区
street: 番地
postalAddress: 番地
postOfficeBox: ビル名
postalCode: 郵便番号
homePostalAddress: 自宅住所
homePhone: 自宅電話

```
dn: uid=odagiri, ou=Users, dc=osstech,dc=co,dc=jp
objectClass: posixAccount
objectClass: inetOrgPerson
cn: odagiri
sn: 小田切
givenname: 耕司
mail: odagiri@osstech.co.jp
o: オープンソース・ソリューション・テクノロジ株式会社
ou: 技術部
title: チーフアーキテクト
employeeNumber: 1
telephoneNumber: 03-1234-5678
facsimileTelephoneNumber: 03-8765-4321
mobile: 090-5432-1234
st: 東京都
l: 品川区西五反田
street: 2-6-3
postalAddress: 2-6-3
postOfficeBox: 東洋ビル
postalCode: 107-0052
homePostalAddress: 神奈川県藤沢市藤沢123-45
homePhone: 0466-23-4567
```

Part 2.

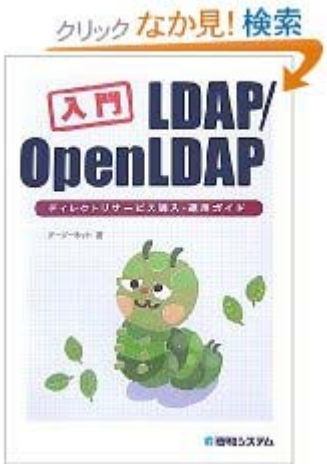
LDAP構築／設定入門

主題 302: インストールおよび開発

- 勉強方法としては、configure ; make によるインストールをやっておくこと。
- configureのオプションも確認しておくこと
- OpenLDAPをコンパイルするのに必要なライブラリ
  - BDB（今はLDBM、GDBMはほとんど使われないが、SQLを始めどんなバックエンドDBが使えるか知っておくこと）
  - OpenSSL（TLSライブラリとして使われる）
    - 通信の暗号化
  - Cyrus SASL
    - 安全な認証方式
  - Kerberos(MITかHeimdal)
    - 安全な認証
    - Kerberos認証のためのスキーマもLDAPに格納

OpenLDAP勉強のための参考書

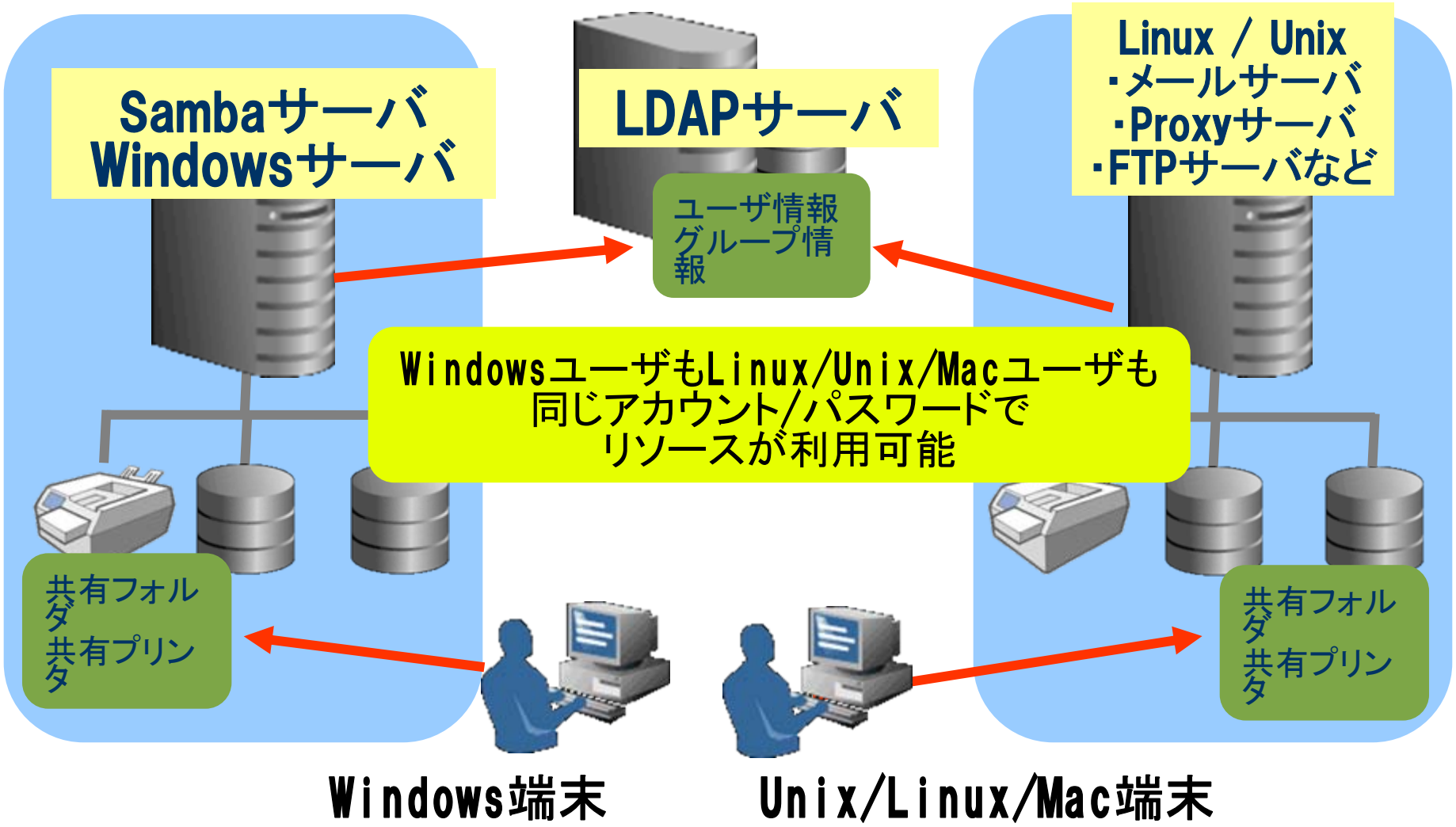
- OpenLDAP入門
  - オープンソースではじめるディレクトリサービス
  - 出版社: 技術評論社
  - 発売日： 2003/07
- 入門LDAP/OpenLDAP
  - ディレクトリサービス導入・運用ガイド
  - 出版社: 秀和システム
  - 発売日： 2007/10



インストール：実システムでの注意

- OpenLDAPはどんどん新しくなるので、書籍の情報では古いことがある。
  - www.openldap.org のドキュメントを読むしかない
- **実際の業務システムでは、configure ; make でインストールしないこと。**
- 業務システムではRPMやDEB、PKGなどOS標準のパッケージ管理システムを使うこと
- コンパイルするのに必要なライブラリは、OS標準のものを使うのが一般的だがBDBだけはOpenLDAP専用のものを使った方が良い。
  - Red Hat のRPMはBDBだけはOS標準を使わないようにSPECファイルが書かれているので、これを参考にすると良い。
    - ✓ 上記理由からRed HatではOpenLDAPのBDBリカバリに db_recover は使わない ！ slapd_db_recover を使う

主題 303: 設定 LDAPによる認証統合

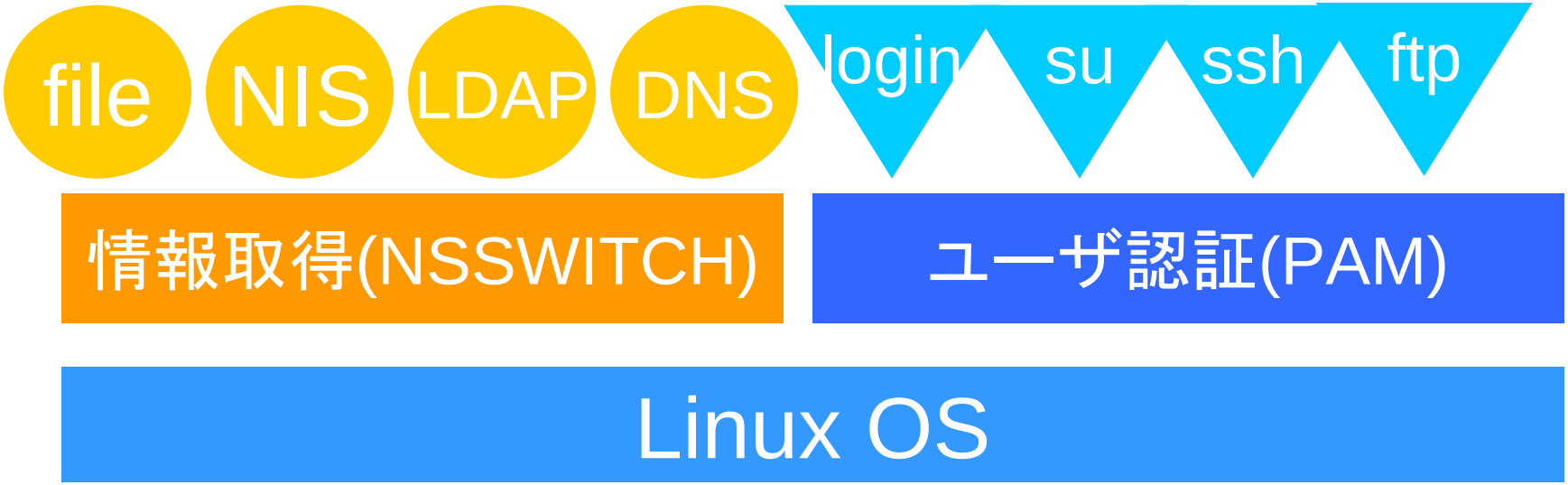


主題 303: 設定 LDAPの設定

- LDAPサーバとしての設定
  - slapd.confの設定
- LDAPクライアントとしての設定
  - NSS設定
  - PAM設定
  - ldap.conf設定

•LDAPクライアントとしての設定

- NSS（ネーム・サービス・スイッチ）機能
  - システムのユーザ名、グループ名、ホスト名の解決方法を設定
  - /etc/nsswitch.confで、各種情報の取得先を指定可能
- PAM認証機構
  - アプリケーション毎の認証方法を設定
  - /etc/pam.d/の中でアプリケーションごとの認証ルールを指定可能



ネームサービススイッチ機能

- LDAPを認証で使用するには/etc/nsswitch.confを以下のように変更

```
passwd:  files  ldap
group:   files  ldap
shadow:  files  ldap
hosts:   files  dns  wins
```

- /lib/libnss_ldap.so.2が呼ばれる。
- /lib/libnss_wins.so.2 を使うとWINS(Windows Internet Name Service)を使って名前解決可能

プラグマブル認証機能

- /etc/pam.d/system-authに以下を設定

```
[root@fs02 /etc]# cat /etc/pam.d/system-auth
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/pam_env.so
auth        sufficient    /lib/security/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/pam_ldap.so use_first_pass
auth        required      /lib/security/pam_deny.so

account     required      /lib/security/pam_unix.so
account     [default=ok user_unknown=ignore service_err=ignore system_err=ignore] /lib/security/pam_ldap.so

password    required      /lib/security/pam_cracklib.so retry=3
password    sufficient    /lib/security/pam_unix.so nullok use_authtok md5 shadow
password    sufficient    /lib/security/pam_ldap.so use_authtok
password    required      /lib/security/pam_deny.so

session     required      /lib/security/pam_limits.so
session     required      /lib/security/pam_unix.so
session     optional      /lib/security/pam_ldap.so
session     required      /lib/security/pam_mkhomedir.so skel=/etc/skel umask=0022
```

- /etc/pam.d/sshdなどに以下を設定

```
#%PAM-1.0
auth        required     /lib/security/pam_stack.so    service=system-auth
account     required     /lib/security/pam_stack.so    service=system-auth
password    required     /lib/security/pam_stack.so    service=system-auth
session     required     /lib/security/pam_stack.so    service=system-auth
```

OpenLDAPサーバの設定

設定ファイル

サーバ：/etc/openldap/slapd.conf

クライアント：

NSS,PAM用：/etc/ldap.conf

ldapaddなどの管理コマンド用:/etc/openldap/ldap.conf

OpenLDAP 管理者ガイド

http://www.ldap.jp/doc

http://www5f.biglobe.ne.jp/~inachi/openldap/

Red Hat Enterprise Linux 4 リファレンスガイド

http://www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/ja/pdf/rhel-rg-ja.pdf

/etc/slapd.confﾊﾟﾗﾒｰﾀ(1)

- suffix ベース・サフィックスを指定する
  通常はﾄﾞﾒｲﾝ名をﾍﾞｰｽに指定
  例）suffix　dc=osstech,dc=co,dc=jp
  suffix ″ou=sales,ou=yokohama,o=company,c=jp″

CN=commonName
L=localityName
ST=stateOrProvinceName
O=organizationName
OU=organizationalUnitName
C=countryName
STREET=streetAddress
DC=domainComponent
UID=userid

/etc/slapd.confパラメータ(2)

- rootdn
  LDAPサーバの管理者のDN(Distinguished Name：識別名)を指定する。
  なお管理者DNを含むユーザDNには、英大文字、英子文字の区別はない。
  管理者DNの例）
  - rootdn "cn=Manager,dc=osstech,dc=co,dc=jp"
- rootpw
  LDAPサーバの管理者パスワードを設定する。
  - そのままのパスワードを指定するか暗号化したものを設定する
  - 例）secret1234というパスワードをSSHAハッシュする
    # slappasswd -s secret1234 -h {SSHA}
  - rootdnをLDAPに登録されているユーザを指定し、LDAPの中にパスワードが格納されていれば、rootpwを指定する必要はない。

/etc/slapd.confﾊﾟﾗﾒｰﾀ(3)

- include
  - 与えたファイルから追加の設定情報を読み込む。
  - 通常はスキーマ定義ファイルを読み込むために使用する
    例） include /etc/openldap/schema/samba.schema
- database
  - LDAPのデータを格納するのに使用するバックエンド・データベースを指定。
- directory
  - databaseファイルを格納するディレクトリを指定
  - 例）directory /var/lib/ldap
- index
  - 作成する索引の属性とタイプを指定する。
    - 例1）uid,gidに関してequal（等値）検索用の索引を作成
      index uidNumber,gidNumber eq
    - 例2）mail（メールアドレス）、surname（名字）に関して、equal検索用とsubinitial（前方一致）の索引を作成
      index mail,surname eq,subinitial

/etc/slapd.confパラメータ(4)

- Slapd.confの例：サフィックスを"dc=osstech,dc=co,dc=jp"、管理者DNを"cn=Manager,dc=osstech,dc=co,dc=jp"、管理者パスワードをsecret1234

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema

database  bdb
directory /var/lib/ldap
suffix "dc=osstech,dc=co,dc=jp"
rootdn "cn=Manager,dc=osstech,dc=co,dc=jp"
rootpw secret1234
index  objectClass,uidNumber,gidNumber,memberUid  eq
index  cn,mail,surname,givenname                  eq,subinitial
index  uid                                        pres,eq
index  rid                                        eq
```

- 設定が終了したら、OpenLDAPデーモンを起動させる。
  # service ldap restart　　※Red Hat系
- システム起動時に自動的に動くように以下を設定
  # chkconfig ldap on　　※Red Hat系

Part 3.

LPIC例題解説

＜問題1＞

replog方式およびsyncrepl方式を使ったシングル・マスターとマルチ・スレーブ構成におけるOpenLDAPの特徴に関して正しい記述をすべてあげなさい。

① LDAPへの書き込み要求はマスターおよびすべてのスレーブLDAP書き込み完了でリターンする。

② LDAPへの書き込み要求はマスターLDAPのみの書き込み完了でリターンし、スレーブLDAPへの書き込みは保証しない。

③ LDAPへの書き込み要求はマスターおよび1台のスレーブLDAP書き込み完了でリターンする。2台目以降のスレーブLDAPへの書き込みは保証しない。

④ マスターのrootdnで指定したユーザDNはスレーブのデータを更新できる権限が必要である。

⑤ マスターのrootdnで指定したユーザDNはスレーブのデータを更新不可とするべきである。

⑥ スレーブを更新すると自動的にマスターに更新が伝搬される。

⑦ スレーブを更新するとエラー(referral)が返るのでクライアントの責任でマスターを更新しなければならない。

＜解説1＞

- LDAPの書き込みはスレーブへの反映を待たずに完了することに注意しましょう。
- シングルマスターの場合、マスターのrootdnを含めてすべてのユーザでスレーブは更新不可としておかないと、マスターのrootdnでスレーブを更新できてしまいマスター／スレーブ間で不整合が起きてしまいます。
- ユーザアプリケーションは直接スレーブを更新することはできません。
- スレーブサーバーからマスターサーバーへ更新が伝搬されることはないのでエラー(referral)が返ってきたらアプリケーションの責任でマスターを更新します。

＜問題2＞

OpenLDAPの複製方式を答えなさい。

① 複製はマスターサーバー側からスレーブサーバー側へ更新が伝搬することで行われる。
② 複製はスレーブサーバー側からマスターサーバー側へ更新を検索してスレーブ自身で更新することで行われる。

＜解説2＞

① replog方式の複製はマスターサーバー側から更新差分ログを使ってスレーブサーバー側へ更新が伝搬することで行われます。
また更新ログを使って行うため複製はスレーブが多いと複製遅延が発生しやすくなります。

② syncrepl方式の複製はスレーブサーバー側からマスターサーバー側へ更新内容を検索してスレーブ自身で更新することで行われます。
スレーブサーバーから検索する方式のため、replog方式よりも複製遅延が少なくなります。

今後replog方式はサポートされなくなりますので、新規サーバで利用するのは出来る限り避け、syncrepl方式を利用しましょう。

＜問題3＞
ユーザ情報（NSS）と認証機構（PAM）でLDAPを利用する時、LDAPサーバーがマスタ（IPアドレス：192.168.0.1)とスレーブ（IPアドレス：192.168.0.2)の2台構成の場合、マスター障害時にスレーブにアクセス出来るようにする設定はどれか？

① /etc/nsswitch.confにpasswd: "ldap 192.168.0.1 192.168.0.2"と記述
② /etc/ldap.conf(※)にhost 192.168.0.1 192.168.0.2と1行で記述
③ /etc/ldap.conf(※)に
host 192.168.0.1
host 192.168.0.2
と2行で記述
④ /etc/openldap/ldap.confにhost 192.168.0.1 192.168.0.2と1行で記述
⑤ /etc/openldap/ldap.confに
host 192.168.0.1
host 192.168.0.2
と2行で記述

(※)/etc/ldap.confはRed Hat系の場合、
Debian (Ubuntu) では /etc/libnss-ldap.conf に相当

＜解説3＞

- LDAPをNSS(Name Service Switch)とPAM(Pluggable Authentication Module)で利用する場合の設定ファイルは/etc/ldap.confになります。
（Debian系 (Ubuntuなど) では NSS は /etc/libnss-ldap.conf,
PAM は /etc/pam_ldap.conf です。）
- この中にマスターとスレーブを指定する場合、hostパラメータに1行で記述します。
この場合、1つ目のサーバーに接続できない場合、2つ目のサーバーへ接続します。
- 2行記述すると2台のサーバーの情報を連結することになります。
- /etc/nsswitch.confはNSSで使うLDAPモジュール名を指定するのみでサーバーアドレスは指定しません。
- /etc/openldap/ldap.confはldapsearchなどのLDAP管理コマンドの設定ファイルとなります。

＜問題4＞
LDAPサーバーとクライアント間の通信を暗号化するLDAPSで利用されるLDAPサーバーのポート番号は次のうちのどれか。

① 389
② 391
③ 631
④ 636

＜解説4＞

- OpenLDAPでは、LDAPの暗号化機能として、LDAPS(LDAP over TLS)を利用することができます。LDAPSを利用する場合、LDAPサーバーがクライアントの接続を待ち受けるポートは636となります。（TLS≒SSL v3）
- OpenLDAPでTLSを利用するためには、OpenLDAPのコンパイル時にOpenSSLライブラリを適切な場所にインストールしておき、configureの--with-tlsオプションが有効になるようにしておかなければなりません。
- なお、現在ではStartTLSと呼ばれる暗号化の方法がOpenLDAPを含む多くのLDAP製品で利用可能になっています。StartTLSを利用すると、LDAPサーバーのポートは通常の389番のままで、クライアント、サーバー間のセッション確立の際にネゴシエーションを行い、暗号化が可能な場合は、暗号化通信を利用することができるようになります。

＜問題5＞ replogサーバの構築

- OpenLDAPのreplog機能を使ったマスターサーバーとスレーブサーバを構築します。
- slapd.confの(ア)～(キ)に入るパラメータを①～⑬から選びなさい。
- 同じパラメータを何度使っても構いませんし、別な記号の場所に同じパラメータが入っても構いません。
- なおアクセス制御はパスワードのみ管理者にアクセス可能にし、他のフィールドは誰でも参照可能とします。
- slapd.confは一部のみを掲載しており、すべてではありませんが、access行はすべて掲載しています。

<マスターサーバーの設定：master.example.co.jpのslapd.conf>

```
suffix ″dc=example,dc=co,dc=jp″
rootdn ″cn=Manager,dc=example,dc=co,dc=jp″
rootpw mstpwd
access to (ア)
access to (イ)
replica uri=″(ウ)″
        binddn=″(エ)″
        bindmethod=simple
        credentials=(オ)
```

<スレーブサーバーの設定：slave.example.co.jpのslapd.conf>

```
suffix ″dc=example,dc=co,dc=jp″
rootdn ″(カ)″
rootpw slvpwd
access to (キ)
access to (ク)
updateref "(ケ)"
```

① dc=example,dc=co,dc=jp
② cn=manager,dc=example,dc=co,dc=jp
③ cn=replica,dc=example,dc=co,dc=jp
④ mstpwd
⑤ slvpwd
⑥ attrs=userPassword
by anonymous auth
by * none
⑦ attrs=userPassword
by dn="cn=manager,dc=example,dc=co,dc=jp" write
by anonymous auth
by * none
⑧ attrs=userPassword
by dn="cn=manager,dc=example,dc=co,dc=jp" read
by anonymous auth
by * none
⑨ * by * read
⑩ * by dn="cn=replica,dc=example,dc=co,dc=jp" read
⑪ * by dn="cn=manager,dc=example,dc=co,dc=jp" write

⑫ ldap://master.example.co.jp
⑬ ldap://slave.example.co.jp

＜正解＞(ア)-⑥, (イ)-⑨, (ウ)-⑬, (エ)-③, (オ)-⑤, (カ)-③, (キ)-⑥（または⑧も正解）, (ク)-⑨, (ケ)-⑫

- replog方式でマスター／スレーブサーバーを構築する時の注意点はマスターのrootdnでスレーブが更新できてはいけない、ということに注意しましょう。
- よってマスターとスレーブのrootdnは同一にしない方が良く、rootdn “(カ)”に
  ② cn=manager,dc=example,dc=co,dc=jp は入れてはいけません。
  ③ cn=replica,dc=example,dc=co,dc=jp の専用のDNを指定します。
- replogではマスターからスレーブに更新に行きますのでreplica uri=“(ウ)”にスレーブのURIを指定し、rootdn "(カ)"とbinddn="(エ)"は同じスレーブのrootdnを指定し、パスワードの(オ)はスレーブのrootdnパスワード⑤を指定します。
- access行は上から順に評価されますので(ア)に⑨ * by * readを入れてはいけません。
  （誰でもすべてをreadできようになってしまいます）
  ⑥を記述し、制限をかけてから、(イ)に⑨を記述します。
- replog方式でのrootdnはそのサーバの全データの更新権限がありますから明示的にwrite権限を与える必要はありませんので(ア)に⑦⑧を指定する必要はありません。
- (キ)(ク)は(ア)(イ)と同じ値で構いません。
  なぜなら通常マスターサーバーのrootdnも認証でしかUserPasswordにアクセスしないからです。しかし、⑧のように明示的にread権を与えても構わないですが、⑦のように更新権は与えてはいけません。
- スレーブは更新できないですから、アプリケーションが更新しにきた場合はマスターのURIを通知できるようにupdateref "(ケ)"にはマスターのURIを指定します。
- なおこのような設定をしたreplog方式ではLDAPデータベースの中に（スレーブサーバにcn=managerでアクセスにくる可能性があるため）②
  cn=manager,dc=example,dc=co,dc=jp
  のエントリは登録する必要がありますが、（マスターサーバにcn=replicaでアクセスにくることはないので）
  ③ cn=replica,dc=example,dc=co,dc=jp
  のエントリは登録する必要がありません。

＜問題6＞ syncreplサーバの構築

- OpenLDAPのsyncrepl機能を使ったマスターサーバーとスレーブサーバーを構築します。
- slapd.confの(ア)～(キ)に入るパラメータを①～⑬から選びなさい。
- 同じパラメータを何度使っても構いませんし、別な記号の場所に同じパラメータが入っても構いません。
- なおアクセス制御はパスワードのみ管理者にアクセス可能にし、他のフィールドは誰でも参照可能とします。
- slapd.confは一部のみを掲載しておりすべてではありませんが、access行のみすべて掲載しています。
- ただし、LDAPデータベースの中に
  ② cn=manager,dc=example,dc=co,dc=jp
  はエントリとして登録されていますが、
  ③ cn=replica,dc=example,dc=co,dc=jp
  は登録されていないものとします。

<マスターサーバーの設定：master.example.co.jpのslapd.conf>

suffix ″dc=example,dc=co,dc=jp″
rootdn ″cn=Manager,dc=example,dc=co,dc=jp″
rootpw mstpwd
access to (ア)
access to (イ)
overlay syncprov

① dc=exsample,dc=co,dc=jp
② cn=manager,dc=exsample,dc=co,dc=jp
③ cn=replica,dc=exsample,dc=co,dc=jp
④ mstpwd
⑤ slvpwd
⑥ attrs=userPassword
by anonymous auth
by * none
⑦ attrs=userPassword
by dn="cn=manager,dc=exsample,dc=co,dc=jp" write
by anonymous auth
by * none
⑧ attrs=userPassword
by dn="cn=replica,dc=exsample,dc=co,dc=jp" write
by anonymous auth
by * none
⑨ attrs=userPassword
by dn="cn=replica,dc=exsample,dc=co,dc=jp" read
by anonymous auth
by * none
⑩ * by * read
⑪ * by dn="cn=replica,dc=exsample,dc=co,dc=jp" read
⑫ * by dn="cn=manager,dc=exsample,dc=co,dc=jp" write
⑬ ldap://master.example.co.jp
⑭ ldap://slave.example.co.jp

<スレーブサーバーの設定：slave.example.co.jpのslapd.conf>

```
suffix "dc=example,dc=co,dc=jp"
rootdn "(ウ)"
rootpw (エ)
access to (オ)
access to (カ)
syncrepl rid=1
        provider="(キ)"
        type=refreshAndPersist
        retry="5 10 30 +"
        searchbase="(ク)"
        scope=sub
        schemachecking=off
        binddn="(ケ)"
        bindmethod=simple
        credentials=(コ)
updateref "(サ)"
```

＜問題6＞ syncreplサーバの構築

① dc=exsample,dc=co,dc=jp
② cn=manager,dc=exsample,dc=co,dc=jp
③ cn=replica,dc=exsample,dc=co,dc=jp
④ mstpwd
⑤ slvpwd
⑥ attrs=userPassword
　　by anonymous auth
　　by * none
⑦ attrs=userPassword
　　by dn="cn=manager,dc=exsample,dc=co,dc=jp" write
　　by anonymous auth
　　by * none
⑧ attrs=userPassword
　　by dn="cn=replica,dc=exsample,dc=co,dc=jp" write
　　by anonymous auth
　　by * none
⑨ attrs=userPassword
　　by dn="cn=replica,dc=exsample,dc=co,dc=jp" read
　　by anonymous auth
　　by * none
⑩ * by * read
⑪ * by dn="cn=replica,dc=exsample,dc=co,dc=jp" read
⑫ * by dn="cn=manager,dc=exsample,dc=co,dc=jp" write
⑬ ldap://master.example.co.jp
⑭ ldap://slave.example.co.jp

＜正解＞(ア)-⑥, (イ)-⑩, (ウ)-②（または③）, (エ)-④（上記が③の時は⑤）
(オ)-⑥,(カ)-⑩,(キ)-⑬,(ク)-①,(ケ)-②,(コ)-④,(サ)-⑬

- syncrepl方式でもマスター／スレーブサーバーを構築する時の注意点はマスターのrootdnを含めすべてのユーザDNでスレーブが更新できてはいけない、ということになります。
- しかし、syncrepl方式ではスレーブのrootdnでスレーブデータを更新できないようにupdaterefがあると誰も更新できない機能が入っています。
- このため、マスターとスレーブのrootdnを同一にしても問題ありませんし、replogと同じように変えても構いません。
- よって(ウ)(エ)に
  rootdn "cn=Manager,dc=example,dc=co,dc=jp“
  rootpw mstpwd
  という組み合わせにしても構いませんし、
  rootdn "replica,dc=exsample,dc=co,dc=jp“
  rootpw slvpwd
  の組み合わせでも構いません。
- ただし、マスターとスレーブのrootdnを同一にする場合はLDAPデータベースの中に
  cn=manager,dc=example,dc=co,dc=jp
  のエントリは登録する必要がありませんが、マスターとスレーブのrootdnを別にする場合はLDAPデータベースの中に
  cn=manager,dc=example,dc=co,dc=jp
  のエントリを登録する必要があります。
  （スレーブに cn=manager,dc=example,dc=co,dc=jp でアクセスできないため）
- この問題では
  cn=replica,dc=example,dc=co,dc=jp
  のエントリは登録されていないとなっているため(ケ)に③をいれることはできません。
- もし、登録されているのなら、マスターのrootdnでアクセスするよりも参照専用の
  cn=replica,dc=example,dc=co,dc=jp
  を（ケ）に入れる方が良いでしょう。
- しかし、この場合access to (オ)に⑨を指定する必要がでてきます。

＜問題7＞

OpenLDAPのバックアップ／リストアについて正しい文をすべて選びなさい。

①slapcatコマンドを使えばLDAPサーバーの起動中／停止中に関わらずバックアップできる。

②slapcatで取り出したのデータは他のLDAP製品にldapaddしてデータリストアできる。

③replogのスレーブサーバーでマスターサーバーとのデータ不整合が起きた場合、マスターサーバーからデータをslapcatして取り出し、スレーブサーバーへslapaddすると不整合が解消できる。

④syncreplのスレーブサーバーのデータが消失した場合、データディレクトリを空にして、スレーブを起動すれば自動的にマスターからオンラインでコピーされる。

＜解説7＞

① LDAPサーバーの起動中に実行したslapcatコマンドの結果はメモリ上のデータがディスクに書き出されていない可能性があり内容が保証されません。 LDAPサーバーを停止させてから実行しましょう。
（最新版のOpenLDAPではバックエンドにBDB、HDBを使った場合はオンラインでslapcatしても問題ない、となっていますが、LDBMなど他のバックエンドでは保証されていません）

② slapcatで得られたデータは上位ツリーから順番に出力される保証がないため、ldapaddの入力として利用することはできません。

③ slapaddは原則LDAPデータが空の状態で行います。データが入っている状態で不整合解消のために使ってはいけません。なぜならば例えばマスターで削除されて、スレーブに残ってしまっているデータにマスターのデータをslapaddしてもスレーブの上にデータが残ったままになってしまうからです。
正しくはreplogのエラーログを参照し、エラー原因を取り除いてから .rejの拡張子を持ったリジェクトファイルをslurpd -o -r でOne-Shotモードで再実行します。

④ syncreplはスレーブからマスターを検索する方式のためスレーブのデータが破壊されてなくなってしまったらデータディレクトリを空にして再起動するだけで自動的にマスターからコピーされて復元できます。
replogの場合は、マスターから差分をコピーする方式のため、スレーブのデータが消失したら復元する手だてがありませんので マスターを停止させて（コピー中に更新されると困るため）スレーブへ手動でコピーする必要があります。

＜問題８＞
OpenLDAPでreplog方式のマスターサーバーとスレーブサーバーがすでに構築して運用している状態で、スレーブサーバーをもう１台追加しようと思います。
手順として正しいものを選びなさい。

① マスターサーバーを停止し、スレーブサーバーにマスターデータを手動コピーし、スレーブサーバーを起動後、マスターのslapd.confを修正の後、マスターを起動する。

② マスターサーバーを起動したまま、スレーブサーバーにマスターデータを手動でコピーし、スレーブサーバーを起動する。

③ マスターサーバーを起動したまま、スレーブサーバーを設定して起動すれば自動的にマスターデータがコピーされ、スレーブサーバーができあがる。

＜解説9＞

- replog方式はマスターサーバーから差分のみを転送する方式ですからスレーブサーバーを起動する前に既存のマスターサーバーのデータをコピーしておく必要があります。
- コピー中にデータ更新されると不整合が起きますので必ずマスターサーバーは停止させてオフラインでコピーしましょう。
- コピーはデータをscpしてもslapcat/slapaddの組み合わせでも構いません。
  (マスターを止める必要があるのでマスターからのldapsearch/ldapaddは使えません。
  １台目のスレーブは止めなくて良いので、そちらからldapsearch/ldapaddすることはできます）

＜問題10＞

OpenLDAPでsyncrepl方式のマスターサーバーとスレーブサーバーがすでに構築して運用している状態で、スレーブサーバーをもう1台追加しようと思います。

手順として正しいものを選びなさい。

①マスターサーバーを停止し、スレーブサーバーにマスターデータをコピーし、　スレーブサーバーを起動後、マスターのslapd.confを修正の後、マスターを起動する。

②マスターサーバーを起動したまま、スレーブサーバーにマスターデータをコピーし、スレーブサーバーを起動後、マスターのslapd.confを修正の後、マスターを起動する。

③マスターサーバーを起動したまま、スレーブサーバーを設定して起動すれば自動的にマスターデータがコピーされ、スレーブサーバーができあがる。

＜解説10＞

- syncrepl方式はスレーブサーバー側から差分を転送する方式ですからスレーブサーバのデータが空の状態で起動しても、既存のマスターサーバーのデータは自動的にすべてコピーしてくれます。
- スレーブの追加でマスターを止める必要もなく簡単にスレーブサーバーを追加できるのがsyncreplの利点です。
- ただし、データが大量にあって初期複製に時間がかかる場合はマスターサーバーを停止させてオフラインでコピーしても構いません。
- この場合コピーはデータをscpするかslapcat/slapaddを使います。（ldapsearch/ldapaddではデフォルトでentryCSNなどの制御情報がコピーされないので使わないほうが良いでしょう）

Part 4.

やってはいけないOpenLDAPサーバ構築

Webの情報を鵜呑みにしないこと！

- LDAP（OpenLDAPやRedHatDS,ApacheDS）に関する情報はとても少ない。特に日本語は少ない
- 本当に正しい（推奨）設定に関する情報が少ない
- OpenLDAPの品質は近年急速に良くなった
- ディストリビューションに含まれるOpenLDAPのバージョンに注意が必要
- 心配なら有償サポートやLDAPユーザ会メーリングリストなどに聞きましょう

やってはいけないOpenLDAPサーバ構築

- バージョンの古いOpenLDAPは使うな！
- replog(slurpd)は使うな！
- マスターとスレーブのrootdnは同じにするな！
- TLSを使おう（SSLじゃあないんだよ）

バージョンの古いOpenLDAPは使うな！

- OpenLDAP 2.2以前はサポート終了
- OpenLDAP 2.3.40以前は複製が抜ける、BDBアクセスでデッドロックなどのバグあり

	OpenLDAP 2.0	OpenLDAP 2.1	OpenLDAP 2.2	OpenLDAP 2.3	OpenLDAP 2.4
初期リリース	2000年8月	2002年6月	2003年12月	2005年6月	2007年10月
最終リリース	2002年9月	2004年4月	2005年11月	2008年7月	2008年11月
最新版	2.0.27	2.1.30	2.2.30	2.3.43	2.4.13
サポートの有無	× 終了	× 終了	× 終了	○ サポート中	○ サポート中
採用Linux	RHEL3 (2.0.27) 2002/9		RHEL4 (2.2.13) 2004/6	RHEL5 (2.3.27) 2006/8	Fedora10 (2.4.12) 2008/11
推奨複製方式	replog	replog	replog	syncrepl	syncrepl

replog(slurpd)は使うな！

- replogは運用が大変
  - エラーリカバリは手操作
  - スレーブの追加時にマスターを止める必要あり
  - スレーブ故障後の修復でもマスターを止める必要あり
  - スレーブ台数が多いと性能劣化
- syncreplは運用が楽
  - エラーリカバリは自動
  - スレーブの追加時にマスターを止める必要なし
  - スレーブ故障後の修復でもマスターを止める必要なし
    データを空にして再起動すれば自動修復
  - syncreplはOpenLDAP 2.2.30 / 2.3.41以降が安全

マスターとスレーブのrootdnは同じにするな！

- replog方式はマスターからスレーブを更新するので
  マスターとスレーブで同じrootdnを使うと危険
- 以下の設定ではスレーブが破壊される可能性がある
  - スレーブにupdateref があればOpenLDAP 2.1以降は大丈夫
    OpenLDAP2.0ではupdateref があっても壊れる

＜マスター設定＞

```
suffix "dc=example,dc=jp"
rootdn "cn=Manager,dc=example,dc=jp"
rootpw secret

replica uri=ldap://slave.example.co.jp
  binddn="cn=Manager,dc=example,dc=jp"
      bindmethod=simple
      credentials=secret
```

＜スレーブ設定＞

```
suffix "dc=example,dc=jp"
rootdn "cn=Manager,dc=example,dc=jp"
rootpw secret
```

※updaterefの設定が足りない！

マスターとスレーブのrootdnは同じにするな！

- replog方式の推奨

＜マスター設定＞
```
suffix "dc=example,dc=jp"
rootdn "cn=Manager,dc=example,dc=jp"
rootpw secret

replica uri=ldap://slave.example.jp
  binddn="cn=replica,dc=example,dc=jp"
      bindmethod=simple
      credentials=himitu
```

＜スレーブ設定＞
```
suffix "dc=example,dc=jp"
rootdn "cn=replica,dc=example,dc=jp"
rootpw himitu

updateref ldap://master.example.jp
updatedn "cn=replica,dc=example,dc=jp"
```

- syncrepl方式の推奨

＜マスター設定＞
```
suffix "dc=example,dc=jp"
rootdn "cn=Manager,dc=example,dc=jp"
rootpw secret
```

＜スレーブ設定＞
```
suffix "dc=example,dc=jp"
rootdn "cn=replica,dc=example,dc=jp"
rootpw himitu
syncrepl rid=1
   provider="ldap://master.example.jp"

binddn="cn=replica,dc=example,dc=jp"
      bindmethod=simple
      credentials=himitu
```


updateref ldap://master.example.jp

TLSを使おう(SSLじゃあないんだよ)

- Mac OS XをLDAPクライアント(LDAP認証)にするにはOpenLDAPでTLSかSASLの設定が必要
- 暗号なしのSimple認証はMac OS Xでは受け付けない
- セキュリティ強化のためにはTLSを使った方が良い
- OpenLDAPはSSLではなく、TLSをサポート
  - 正確にはSSLとTLSは違う
  - OpenLDAPはOpenSSLで実装されており、OpenSSLはSSLとTLSの両方をサポートしているのでOpenLDAPはSSLと思われているが正確にはTLSを使う

実は知らないと困るBDBコマンドとパラメータ

- 現在OpenLDAPの推奨バックエンドはBDBなので、BDBのチューニングやコマンドを知ることも重要
- slapd.conf
  - checkpoint <更新量> <間隔>
  - cache size <エントリ数>
- DB_CONFIG
  - cachesize
  - DB_LOG_AUTOREMOVE
  - lg_max
- db_recover (slapd_db_recover)コマンド
  予期しないアプリケーション、データベース、またはシステムの障害が発生した後、データベースを整合性のある状態に復元します。
- db_verify (slapd_db_verify)コマンド
  ファイルおよびファイル内に含まれるデータベースの構造を検証します。
- db_archive(slapd_db_archive)
  不要になったログファイルを表示したり、削除する

Part 5.

LPIC勉強法
OpenLDAP編まとめ

LPICのための勉強方法

- LDAP概要
  - DIT,ObjectClass,Schema
  - マスター／スレーブやプロバイダー／コンシューマー、リファラルの正しい理解
  - LDAPの本質を理解すること、
    検索専用とか検索が早いなどは本質ではない
  - 負荷分散と冗長化
- OpenLDAPの導入
  - configureオプションの理解、自分でmakeしてみること
  - BDBやSASL,SSL,Kerberosを理解すること
- OpenLDAP構築
  - 複製の仕組みの理解
  - セキュリティに対する考慮
- OpenLDAP運用
  - バックアップ/リカバリ

実システム構築での注意点

- DITは複雑にしない、組織にマッピングしない、管理者にあわせる
- 自分でconfigure,makeはしない
- OS標準のBDBは使わない
- 複製の仕組みの理解
- セキュリティに対する考慮
  - TLS通信やSASL GSSAPI機構による認証
  - rootdnと複製dnを分ける
  - プログラム毎に使用するdnを分ける
  - 細かなアクセス制御