LDAP Linux HOWTO

Luiz Ernesto Pinheiro Malere, *malere@yahoo.com*, перевод Шепелевича Константина, *shki@ukr.net*
v1.01, 15 February 2000,

В этом документе представлена информация о том, как устанавливать, настраивать, запускать и сопровождать LDAP сервер в системе Linux (Lightweight Directory Access Protocol - Облегченный Протокол Доступа к Каталогам /прим. перевод. здесь и далее во избежание путаницы терминов, "каталог" используется в смысле LDAP directory, а "директория" - в смысле file system directory/). Описано также как создавать LDAP базы, как обновлять и удалять информацию в базе, как организовать Roaming Access и как использовать адресную книгу Netscape. Этот документ создан по материалам информационных страниц Мичиганского Университета об LDAP.

Содержание

1 Введение

Главная цель этого документа - установить и научиться использовать LDAP сервер на Вашем компьютере (здесь и далее подразумевается, что у Вас установлена Linux OS). Вы изучите как устанавливать, настраивать, запускать и поддерживать LDAP сервер. После этого Вы также изучите как можно сохранять, получать и обновлять информацию в Вашем Каталоге используя LDAP клиенты и утилиты. Демон для LDAP сервера называется *slapd* и работает на многих UNIX платформах (а также семействе MacOS X от Apple. прим.перев.).
Существует отдельный демон, который занимается дублированием и синхронизацией баз LDAP серверов. Он называется *slurpd*, и ... это пока все о нем. Следуя этому документу Вы настроите slapd, который будет предоставлять сервис Каталогов только для Вашего локального домена, без дублирования, а следовательно, без slurpd.
Здесь представлена простая конфигурация сервера, хорошая для начала и легкая в развитии и расширении в дальнейшем, если у Вас появится желание сделать это. Информация, представленная в этом документе, является неплохой отправной точкой в использовании LDAP протокола. Возможно после прочтения этого документа Вы почувствуете себя способным расширить возможности Вашего сервера и даже написать свой собственный клиент, используя C, C++ или Java Development Kits.

1.1 Что такое LDAP?

LDAP это клиент-серверный протокол доступа к сервису Каталогов. Изначально он использовался для доступа к X.500 серверам, однако все чаще используется автономно или для доступа к другим

типам серверов Каталогов.

1.2 Что такое сервис Каталогов (Directory Service)?

Каталог - это та же база данных, склонная содержать однако более описательную, основанную на атрибутах информацию. В общем, информация из Каталогов читается намного чаще нежели записывается. Как следствие, Каталоги обычно не реализуют сложных транзакций или схем отката, которые как правило используют настоящие базы данных при выполнении комплексных обновлений информации. Обновление в Каталоге происходит по простой схеме "изменяем-все-или-ничего", если это вообще позволено.
Каталоги "заточены"для быстрого ответа на объемные запросы или операции поиска. Они имеют способность многократно дублировать информацию для увеличения доступности и надежности, и в то же время сокращать время ответа. Когда информация дублируется, может возникать временное несоответствие между копиями, но в этом нет ничего страшного, поскольку копии время от времени синхронизируются.
Существует много способов предоставления сервиса Каталогов. Различные методы позволяют сохранять в Каталоге разную информацию, описывать различные требования к тому, как к этой информации ссылаться, запрашивать ее или обновлять, как ее защищать от неавторизованого доступа. Некоторые Каталоги локальны, т.е. предоставляют сервис только ограниченной аудитории (например, сервис finger на отдельной машине, не присоединенной к какой-либо сети). Другие предоставляют сервис глобально, разрешая доступ намного большей аудитории.

1.3 Как LDAP работает?

Сервис Каталогов LDAP основан на клиент-серверной модели. Один или несколько LDAP серверов содержат данные формируя дерево LDAP Каталога или LDAP базу данных. LDAP клиент соединяется с LDAP сервером и задает ему вопросы. Сервер возвращает ответ или ссылку, по которой клиент может получить дополнительную информацию (обычно это другой LDAP сервер). Не имеет значения к какому LDAP серверу присоединился клиент, он получит тот же вид Каталога; имя присутствующее на одном LDAP сервере ссылается на такую-же запись /entry/, как это было бы на другом серевере. Это особое свойство глобальных Каталогов, таких как LDAP.

1.4 LDAP базы, объекты и атрибуты

slapd поставляется с тремя системами баз данных /backends/, из которых можно выбирать. Это LDBM, высокопроизводительная дисковая система баз данных; SHELL, система интерфейсов к обычным командам UNIX или скриптам shell; и PASSWD база обыкновенного файла паролей.
В этом документе, я предполагаю, что вы выбрали LDBM.
LDBM присваивает компактный четырехбайтовый уникальный идентификатор для каждой записи в базе данных. Она использует этот идентификатор как связку записи с индексом при индексации базы. База содержит один главный индексный файл, называемый id2entry, который отображает уникальный идентификатор записи (EID) на текст, представляющий саму запись. Также поддерживаются другие индексные файлы.
Для импорта и экспорта информации Каталогов между LDAP серверами, или для описания набора изменений, которые нужно сделать в Каталоге, обычно используется файловый формат LDIF, LDAP Data Interchange Format /формат обмена данными LDAP/. Файл LDIF сохраняет информацию в объектно-ориентированной иерархии записей. Дистрибутив LDAP, который Вы будете использовать, содержить утилиту для конвертирования LDIF файлов в LDBM формат.
Обычный LDIF файл имеет следующий вид:

```
dn: o=TUDelft, c=NL
o: TUDelft
objectclass: organization
dn: cn=Luiz Malere, o=TUDelft, c=NL
```

```
cn: Luiz Malere
sn: Malere
mail: malere@yahoo.com
objectclass: person
```

Как Вы видите каждая запись уникально идентифицируется выделенным именем /distinguished name/, или DN. DN состоит из имени записи плюс путь всех имен, определяющий положение данной записи в иерархии Каталога.
В LDAP объектный класс определяет набор аттрибутов, которые можно использовать для определения записи. Стандарт LDAP предоставляет следующие базовые типы объектных классов:

- Groups - Группы в каталоге, включая неупорядоченный список отдельных объектов и групп объектов.
- Locations - Местоположения, такие как название страны и описание.
- Organizations - Организации в каталоге.
- People - Люди в каталоге.

Запись может принадлежать к более чем одному объектному классу. К примеру, запись персоны определяется объектным классом *person*, но может быть также определена атрибутами в объектных классах inetOrgPerson, groupOfNames, и organization. Структура объектных классов Каталога (его схема) определяет общий список требуемых и разрешенных атрибутов для конкретной записи. Данные Каталога представлены парами атрибут-значение. Каждая отдельная часть информации ассоциирована с описывающим атрибутом.
Например, commonName, или cn, атрибут используется для сохранения имени персоны. Некто по имени Jonas Salk может быть представлен в Каталоге как

```
cn: Jonas Salk
```

Каждый добавленный в Каталог человек определяется набором атрибутов в объектном классе *person*. Другие атрибуты используемые для определения такой записи могли бы включать:

```
givenname: Jonas
surname: Salk
mail: jonass@airius.com
```

Обязательные атрибуты - это атрибуты, которые _всегда_должны_присутствовать_ в записях, использующих объектный класс. Все записи должны содержать objectClass атрибут, по которому хранится список объектных классов, к которым принадлежит запись.
Разрешенные атрибуты - это атрибуты, которые _могут_присутствовать_ в записях, использующих объектный класс. Например, в объектном классе person, атрибуты cn и sn обязательные, а description, telephoneNumber, seeAlso и userpassword разрешенные, но не обязательные.
Каждый атрибут имеет соответствующее определение синтаксиса. Определение синтаксиса описывает тип информации предоставляемой атрибутом:

- bin бинарные данные
- ces чувствительная к регистру строка (регистр учитывается при сравнениии)
- cis не чувствительная к регистру строка (регистр игнорируется при сравнении)
- tel строка телефона (как cis, но пробелы и тире '-' при сравнении игнорируются)
- dn выделенное имя

Можете перейти к 3 (разделу 3), чтобы узнать, где в Вашей системе находятся определения объектых классов и атрибутов.

1.5 Новые версии этого документа

Этот документ может исправлятся и обновлятся по отзывам читателей. Новые версии следует искать по адресу:
http://dutedin.et.tudelft.nl/~malere/LDAP-Linux-HOWTO.html

1.6 Отзывы и предложения

Если у Вас есть какие-либо сомнения об информации, представленной в этом документе, пожалуйста, свяжитесь со мной по адресу:
malere@yahoo.com
Если у Вас есть комментарии и/или предложения, также сообщайте.

1.7 Благодарности

Этот Howto явился результатом моего сотрудничества с университетом TUDelft University - Netherlands. Я бы хотел поблагодарить людей, которые сподвигли меня написать этот документ: Rene van Leuken и Wim Tiwon. Большое Вам спасибо. Они тоже фаны Linux, как и я.

1.8 Copyright и Disclaimer

2 Установка LDAP сервера

Необходимы четыре шага для установки сервера: скачать дистрибутив, распаковать, сконфигурировать Makefiles и собрать.

2.1 Скачиваем дистрибутив

Существует два дистрибутива LDAP серверов: LDAP сервер Мичиганского Университета и OpenLDAP сервер. Существует также Netscape Directory Server, который распространяется бесплатно только под определенными условиями (например, для учебных заведений). В основе OpenLDAP сервера лежит последняя версия сервера Мичиганского Университета, более того по нем есть списки рассылки и дополнительная документация. В этом документе предполагается, что Вы используете OpenLDAP сервер.
Последняя версия доступна по адресу:
http://www.openldap.org
Если Вы хотите получить последнюю версию сервера Мичиганского Университета, посмотрите здесь:
ftp://terminator.rs.itd.umich.edu/ldap
Для написания этого документа я использовал последюю стабильную версию OpenLDAP и OpenLDAP 1.2.8 на дистрибутиве Slackware, который имел ядро версии 2.2.6. На сайте OpenLDAP Вы найдете последние разрабатываемую и стабильную версии OpenLDAP сервера. На момент

последнего обновления этого документа, последней стабильной версией был openldap-stable-990918.tgz, также доступный через ссылку openldap-stable.tgz. Последней разрабатываемой версией был openldap-1_2_8.tgz.

2.2 Распаковка сервера

Теперь, когда у Вас на локальной машине есть скачанный запакованый дистрибутив, Вы можете распаковать его.
Для начала скопируйте дистрибутив в желаемую директорию, например, /usr/local.
Затем используйте следующую комманду:

```
tar xvzf openldap-stable.tgz
```

Вы также можете использовать другую комманду:

```
gunzip openldap-stable.tgz | tar xvf -
```

2.3 Настройка дистрибутива

Существует несколько параметров, которые Вам следует изменить, чтобы получить исполняемый модуль, наилучшим образом соответствующий Вашей системе.
Настройка дистрибутива происходит в 2 шага:

- Отредактируйте файл *ldapconfig.h.edit*, который находится в поддиректории *include*, той директории, куда Вы распаковали дистрибутив.
- Запустите скрипт configure (если Вы ТОТ парень/девушка, Вы можете редактировать файл напрямую, без использования скрипта :)

В файле include/ldapconfig.h.edit Вы можете изменять параметры наподобие местоположения демонов slapd и slurpd. Сам по себе этот файл очень хорошо документирован и имеет установки по умолчанию, которые удовлетворят большинство администраторов, так что если Вы спешите, можете пропустить этот шаг и сразу выполнить:

```
vi include/ldapconfig.h.edit
```

Исходники сервера OpenLDAP распространяются с конфигурационным скриптом для установки параметров наподобие директории инсталяции, флагов компилятора и линковщика. Выполните следующую комманду в директории, где Вы распоковали дистрибутив:

```
./configure --help
```

Эта комманда выдаст все параметры, которые Вы можете изменять с помощью скрипта configure до сборки сервера. Вот некоторые полезные параметры –prefix=pref, –exec-prefix=eprefix и –bindir=dir для директории инсталирования. Обычно, если Вы запускаете configure без параметров, он автоматически определяет подходящие параметры и подготавливает дистрибутив для установки в директорию по умолчанию. Так что просто выполните:

```
./configure
```

и убедитесь по выходным данным, что все прошло успешно.

2.4 Сборка сервера

После конфигурирования дистрибутива, Вы можете начинать собирать его. Для начала постройте зависимости, используя комманду:

```
make depend
```

После этого собирайте сервер, используя комманду:

```
make
```

Если все шло хорошо, серевер соберется как был сконфигурирован. Если нет, вернитесь на предыдущий шаг и пересмотрите параметры конфигурации. В первую очередь нужно ознакомится с коментариями по специфике различных платформ, они находятся в поддиректории doc/install/hints директории, где Вы распаковали дистрибутив.
Теперь инсталируем выполняемые модули и страницы man. Вам могут потребоваться права администратора, чтобы сделать это (в зависимости от того, куда Вы собираетесь инсталировать):

```
su
make install
```

Это все, теперь Вы имеете бинарники сереверов и еще некоторых утилит. Можете прямо перейти к 3 (следующему разделу) для ознакомления с настройкой Вашего LDAP сервера.
Если перед началом изучения, как конфигурировать Ваш LDAP серевер, Вы хотите протестировать только что собранные бинарники, то последние версии OpenLDAP серевера поставляются с тестовым скриптом. Во время последнего обновления этого документа тестовый скрипт не был на 100% стабильным во всех тестах, которые он проводил. В любом случае Вы можете попробовать запустить его, и если что-то со скриптом будет не так, прервать его выполнение нажатием Ctrl-C. В моем случае до того как скрипт завис, я смог увидеть несколько сообщений о благополучном прохождении наиболее общих тестов. Для запуска скрипта, перейдите в поддиректорию `/test` места распаковки дистрибутива и выполните:

```
make
```

3 Настройка LDAP сервера

Как только дистрибутив собран и инсталирован, Вы готовы настраивать его для использования на Вашем компьютере. Все настройки параметров времени выполнения серевера slapd производятся в файле slapd.conf, инсталированного в директории prefix, которую Вы указали при выполнении конфигурационного скрита, или по умолчанию в /usr/local/etc/openldap.
В этой директории Вы также найдете следующие файлы *slapd.oc.conf* и *slapd.at.conf*, которые включаются в slapd.conf файл (см. параметр include в 3.2 (разделе 3.2)) и которые содержат определения объектных классов и атрибутов для LDAP, соответственно. Далее последует описание общего формата конфигурационного файла, а за ним детальное описание каждого параметра в этом файле.

3.1 Формат конфигурационного файла

Файл slapd.conf содержит серию глобальных параметров, которые применяются к серверу slapd вцелом (включая все типы баз дынных), за которой следуют (а могут и вообще отсутствовать) определения специфичные для конкретной базы данных.
Глобальные параметры могут быть переопределены в специфичной секции (для параметров, которые встречаются в файле slapd.conf несколько раз, последнее из опреледений имеет силу действия). Пустые и закомментированные - начинающиеся с символа "#",- строки игнорируются. Если строка начинается с пробела, она рассматривается как продолжение предыдущей строки. Общий формат файла slapd.conf следующий:

```
# comment - these options apply to every database
<global config options>
# first database definition & config options
database <backend 1 type>
<config options specific to backend 1>
# second database definition & config options
database <backend 2 type>
```

```
<config options specific to backend 2>
# subsequent database definitions & config options
...
```

Аргументы конфигурационной строки разделяются пробелами. Если аргумент содержит пробел, он должен быть заключен в кавычки "как здесь". Если аргумент содержит кавычки или обратную косую черту '\', этим символам должен предшествовать символ обратной косой черты '\' (например, '\\d').
Дистрибутив содержит пример конфигурационного файла, который можно использовать как рабочий. Также предоставляется файл slapd.at.conf, который содержит много определений наиболее часто используемых атрибутов, и файл slapd.oc.conf с определениями часто используемых объектных классов.

3.2 Глобальные Параметры

Параметры описываемые в этом разделе применяются ко всем базам данных, если не будут где-то специально переопределены в секции определений конкретной базы. Аргументы параметра, которые должны быть заполнены реальными значениями, показаны в скобках <>.

```
access to <what> [ by <who> <accesslevel> ]+
```

Этот параметр предоставляет доступ (определенный уровнем <accesslevel>) для оперирования записями и/или атрибутами (определено в <what>) для одного или более клиентов (определено в <who>). Более детальное описание смотри в разделе "Примеры Управления Доступом".

```
attribute <name> [<name2>] { bin | ces | cis | tel | dn }
```

Этот параметр ассоциирует синтиксис с именем атрибута. По умолчанию атрибут имеет синтаксис cis. Атрибут может иметь необязательное альтернативное имя. Существуют следующие возможные синтаксисы и их описание:
bin : двоичные данные ces : чувствительная к регистру строка (регистр учитывается при сравнении строк) cis : нечувствительная к регистру строка (регистр не учитывается при сравнении строк) tel : строка телефонного номера (как cis, плюс пробелы и тире '-' игнорируются при сравнении) dn : выделенное имя

```
defaultaccess { none | compare | search | read | write }
```

Этот параметр определяет доступ по умолчанию, который предоставляется всем клиентам, не удовлетворяющим ни одному другому правилу доступа (смотрите ниже "Примеры Управления Доступом"). Имейте в виду, что более высокий уровень доступа включает все более ограниченные уровни доступа (например, доступ на запись, разрешает чтение, поиск и сравнение.
По умолчанию: defaultaccess read

```
include <filename>
```

Этот параметр определяет, что slapd должен прочитать дополнительную информацию из обозначенного файла, до того как продолжит вычитывать текущий файл. Включаемый файл должен иметь формат конфигурационного файла slapd. Вы можете использовать этот параметр для включения определений объектный классов и атрибутов Вашей базы данных. Дистрибутив LDAP поставляется с файлами slapd.oc.conf и slapd.at.conf. Примечание: Будьте осторожны при использовании этого параметра, поскольку ни ограничений на количество включений, ни проверок на циклическое включение не существует.

```
loglevel <integer>
```

Этот параметр определяет уровень журнального протоколирования отладочной информации и оперативной статистики (на данный момент ведется протоколирование в syslogd(8)). Чтобы это работало, Вы должны собрать slapd с включенным флагом DLDAP_DEBUG (по умолчанию работают только два уровня протоколирования). Уровни протоколирования кумулятивны (т.е. последующие включают предыдущие). Чтобы узнать, какая информация соответствует определенному уровню, запустите slapd с ключом ?, или изучите нижеприведенную таблицу. Допустимые значения уровня <integer> следующие:

```
1 трассировка вызова функций
2 обработка отладочных пакетов
4 детальная отладочная трассировка
8 управление соединениями
16 протоколирование входящих и исходящих пакетов
32 обработка фильтра поиска
64 обработка конфигурационного файла
128 обработка параметров доступа
256 протокол статистики соединения/обработка/результаты
512 протокол статистики отсылаемых записей
1024 протокол взаимодействия с shell
2048 протокол отладочной информации парсера различных елементов
```

Например: loglevel 255
Это приведет к протоколированию огромного количества отладочной информации.
По умолчанию: loglevel 256

```
objectclass <name>
[ requires <attrs> ]
[ allows <attrs> ]
```

Этот параметр определяет схему правил для данного объектного класса. Используется совместно с параметром schemacheck.

```
referral <url>
```

Этот параметр определяет ссылку для выдачи клиенту, если slapd не может найти локальную базу для обработки запроса.
Например: referral ldap://ldap.itd.umich.edu
Здесь нелокальные запросы будут передаваться на LDAP сервер Мичиганского Университета. Умные клиенты могут повторить запрос на этот сервер, но имейте в виду, что большинство таких клиентов могут обработать только простые LDAP URLы, содержащие только сервер и, иногда, выделенное имя.

```
schemacheck { on | off }
```

Этот параметр включает и выключает проверку схемы. Если проверка схемы включена, то все добавляемые или изменяемые записи будут проверяться на соответствие правилам схемы, задаваемым их объектным классом(ами), как определено в соответствующем параметре(ах). Если проверка схемы выключена, то все это не проводится.
По умолчанию: schemacheck off

```
sizelimit <integer>
```

Этот параметр определяет максимальное количество статей возвращаемых после операции поиска.
По умолчанию: sizelimit 500

```
srvtab <filename>
```

Этот параметр определяет файл srvtab, в котором slapd может найти ключи kerberos, необходимые при аутентификации клиентов. Этот параметр имеет значение, только когда для целей аутентификации используется kerberos, что должно быть специально определено перед компиляцией сервера, включением соответствующий флагов в файле Make-common.
По умолчанию: srvtab /etc/srvtab

```
timelimit <integer>
```

Этот параметр определяет максимальное количество секунд (реального времени), которые slapd потратит отвечая на запрос поиска. Если запрос не закончится за это время, буде возвращен результат, сообщающий о превышении временного лимита.
По умолчанию: timelimit 3600

3.3 Общие Параметры Баз Данных

Параметры из этой секции действуют только на те базы, в которых они определены. Они поддерживаются всеми типами баз.

```
database <databasetype>
```

Этот параметр обозначает начало определения нового экземпляра базы данных. Аргумент <databasetype> должен быть одним из ldbm, shell, или passwd, в зависимости от того, какая из них будет обслуживать базу сервера.
Например: database ldbm
обозначает начало определения нового экземпляра базы данных под управлением LDBM.

```
lastmod { on | off }
```

Этот параметр определяет поддерживает ли автоматически slapd заполнение в записи атрибутов modifiersName, modifyTimestamp, creatorsName, и createTimestamp.
По умолчанию: lastmod off

```
readonly { on | off }
```

Этот параметр переводит базу данных в режим "только чтение". Любая попытка изменить базу данных вернет ошибку "unwilling to perform".
По умолчанию: readonly off

```
replica host=<hostname>[:<port>]
"binddn=<DN>"
bindmethod={ simple | kerberos }
[credentials=<password>]
[srvtab=<filename>]
```

Этот параметр определяет дублирующих сервер для этой базы данных. Аргумент host= определяет сервер и, опционально, порт, по которому может быть найден подчиненный экземпляр slapd. В качестве <hostname> можно использовать как доменный так и IP адрес. Если <port> не указан, используется стандартный номер порта для LDAP (389). Аргумент binddn задает DN в качестве привязки для обновления подчиненных slapd. Это должен быть DN с правами доступа чтение/запись для базы данных подчиненного slapd, обычно задается как "rootdn"в конфигурационном файле подчиненного slapd. Он также должен соответствовать параметру updatedn в конфигурационном файле подчиненного slapd. Поскольку DN обычно включает пробелы, строка "binddn=<DN>"должна быть заключена в кавычки. Аргумент bindmethod может быть либо simple либо kerberos, в зависимости от того, какой тип аутентификации (простой по паролю или керберос) используется при соединении с подчиненным slapd. Простая аутентификация требует задания правильного пароля, керберос - правильного srvtab файла. Аргумент credentials=, который используется только при простой аутентификации, задает пароль для binddn на подчиненном slapd. Аргумент srvtab=, который используется только при керберос аутентификации, определяет файл, содержащий ключ керберос для подчиненного slapd. Если этот аргумент опущен, используется /etc/srvtab.

```
replogfile <filename>
```

Этот параметр определяет имя дублирующего журнального файла, в который slapd протоколирует изменения. Дублирующий журнал обычно записывается slapd и читается slurpd. Как правило, этот параметр используется только когда для дублирования базы данных используется slurpd. Однако, Вы также можете использовать его для генерации журнала транзакций. если slurpd не запущен. В этом случае Вам необходимо периодически урезать этот файл, в противном случае он будет расти неограниченно.

```
rootdn <dn>
```

Этот параметр определяет точку входа, которая не подпадает под контроль доступа или административные ограничения для управления этот базой данных.
Например: rootdn "cn=Manager, o=U of M, c=US"

```
rootkrbname <kerberosname>
```

Этот параметр определяет керберос имя для указанного выше DN, и действует всегда, независимо от того имеет ли указанный DN атрибут krbName или существует вообще. Этот параметр полезен при создании базы данных, а также когда для дублирования используется slurpd.
Например: rootkrbname admin@umich.edu

```
rootpw <password>
```

Этот параметр определяет пароль для указанного выше DN, и действует всегда, независимо от того имеет ли указанный DN пароль или существует вообще. Этот параметр полезен при создании базы данных, а также когда для дублирования используется slurpd. Не задавайте пароль открытым текстом для этого параметра. Как минимум используйте crypto для шифрования (или запись с файла /etc/passwd). slapd поддерживает и другие типы шифрования.
Пример:

```
rootpw secret
rootpw {crypto}encrypted_password_here
```

```
suffix <dn suffix>
```

Этот параметр определяет DN суффикс запросов, которые будут пересылаться к этой базе данных. Может быть задано несколько суффиксов (в отдельных строках), но как минимум один необходим для каждой определяемой базы данных.
Пример: suffix "o=University of Michigan, c=US"
т.е. запросы с DN, оканчивающимися на "o=University of Michigan, c=US"будум переданы данной базе данных. Примечание: когда ищется база для обработки запроса, slapd просматривает определения суффиксов в том порядке, как они следуют в файле. Таким образом, если суффикс одной базы является префиксом другой, то первая должна быть определена в конфигурационном файле позже.

```
updatedn <dn>
```

Этот параметр применим только для подчиненных slapd. Он определяет DN, по которому разрешено вносить изменения в дубликат (обычно это DN, с которым свызывается slurpd, когда вносятся изменения в дубликат).

3.4 Параметры специфичные для LDBM

Параметры в этой секции применяются только для LDBM баз данных. Поэтому они должны размещаться после строки "database ldbm"и до любой другой строки с параметром "database".

```
cachesize <integer>
```

Этот параметр определяет размер кеша (измеряемого количеством записей) в памяти, поддерживаемого данным экземпляром LDBM базы данных.
По умолчанию: cachesize 1000
т.е. тысяча записей.

```
dbcachesize <integer>
```

Этот параметр определяет размер кеша в байтах, который ассоциируется с каждым открытым файлом индексов. Если это не поддерживается нижележащей базой, то параметр игнорируется безо всяких комментариев. Увеличениее значения этого парамтра приводит к использованию большего количества памяти, однако может значительно повысить производительность, особенно при изменениях в базе или во время построения индекса.
По умолчанию: dbcachesize 100000

```
directory <directory>
```

Этот параметр определяет директорию, в которой находятся файлы LDBM, содержащие базу данных и соответствующие индексы.
По умолчанию: directory /usr/tmp

```
index {<attrlist> | default} [pres,eq,approx,sub,none]
```

Этот параметр определяет поддерживаемые индексы для заданных атрибутов. Если задан только аргумент <attrlist>, поддерживаются все возможные индексы.
Например:

```
index cn
index sn,uid eq,sub,approx
index default none
```

Этот пример задает поддержку всех индексов для атрибута cn; индексы equality/совпадения/, substring/вхождения/, и approximate/подобия/ для атрибутов sn и uid; и отсутствие индексации для всех остальных атрибутов.

```
mode <integer>
```

Этот параметр определяет режим доступа, который по умолчанию присваивается ново-созданному файлу индекса.
По умолчанию: mode 0600
т.е. чтение/запись для владельца

3.5 Примеры Управления Доступом

Средства управления доступом, представленные в 3.2 (разделе 3.2) довольно мощные. Этот раздел описывает несколько примеров их использования. Для начала, довольно простой пример:

```
access to * by * read
```

Эта директива предоставляет всем право чтения. Если она одна, то это то же, что и задание параметра defaultaccess:

```
defaultaccess read
```

Следующий пример показывает использование регулярных выражений для выбора записей по DN в двух директивах доступа, причем порядок записи имеет значение.

```
access to dn=".*, o=U of M, c=US"
by * search
access to dn=".*, c=US"
by * read
```

Разрешается чтение записей в поддереве c=US, за исключением записей в поддереве "o=University of Michigan, c=US", для которых разрешен поиск. Если бы порядок этих директив был обратным, то специфика U-M директивы никогда бы не проявилась, поскольку все U-M статьи также и c=US статьи (т.е. подобные фильтры должны следовать в порядке от более специфичного к более общему, прим. перев.).
Следующий пример опять показывает важность порядка записи, как обоих директив доступа так и операторов "by". Он также показывает использование аргумента выбора атрибута /attr/, для предоставления доступа к определенному атрибуту, и задание различных клиентов <who>.

```
access to dn=".*, o=U of M, c=US" attr=homePhone
by self write
by dn=".*, o=U of M, c=US" search
by domain=.*\.umich\.edu read
by * compare
access to dn=".*, o=U of M, c=US"
by self write
by dn=".*, o=U of M, c=US" search
by * none
```

Этот пример применяется к записям в поддереве "o=U of M, c=US". Для всех аттрибутов, за исключением homePhone, сама запись имеет право модификации, другие U-M статьи могут использовать их для поиска, все остальные не имеют доступа. Атрибут homePhone доступен для модификации самой записью, для поиска - другими U-M записями, для чтения - клиентам с домена umich.edu, и для сравнения для всех остальных.
Иногда полезно позволить специфичному DN добавлять себя или удалять с атрибута. Например, если бы Вы хотели создать группу, при этом разрешая пользователям только добавлять и удалять свои собственные DN в атрибут member, Вы бы использовали директиву доступа подобно следующей:

```
access to attr=member,entry
by dnattr=member selfwrite
```

Аргумент dnattr <who> указывает на то, что доступ применяется к записям, обозначенным в атрибуте member. Аргумент selfwrite говорит о том, что такие member(ы) могут добавлять или удалять только свои собственные DN в/из атрибута, но не другие значения. Добавление атрбута entry необходимо, поскольку доступ к какому-либо атрибуту записи невозможен, при отстутствии доступа к записи вцелом.
Заметьте, что конструкция attr=member в операторе <what> является сокращенной формой оператора "dn=* attr=member"(т.е. выборка всех записей с аттрибутом member).

4 Запуск LDAP сервера

slapd может быть запущен в двух различных режимах: как автономный (stand-alone) сервис и из под inetd(8). Рекомендуется использовать автономный режим, особенно если в качестве базы данных Вы используете базу LDBM. Это позволяет базе использовать все преимущества кеширования, и избежать проблем совпадений в индексных файлах LDBM. Если Вы используете только PASSWD или SHELL базы, запуск из iinetd может быть приемлемым вариантом.

4.1 Параметры Коммандной Строки

slapd предполагает следующие параметры в коммандной строке:

```
-d <level> | ?
```

Этот параметр устанавливает отладочный уровень slapd в <level>. Когда в качестве level стоит символ ?, slapd распечатывает все доступные уровни и заканчивает работу независимо от остальных установленных параметров. Сейчас доступны следующие уровни отладки:

```
1 трассировка вызова функций
2 обработка отладочных пакетов
4 детальная отладочная трассировка
8 управление соединениями
16 протоколирование входящих и исходящих пакетов
32 обработка фильтра поиска
64 обработка конфигурационного файла
128 обработка параметров доступа
256 протокол статистики соединения/обработка/результаты
512 протокол статистики отсылаемых статей
1024 протокол взаимодействия с shell
2048 протокол отладочной информации парсера различных элементов
65535 режим полной отладки
```

Уровни отладки адитивны. Т.е., если Вы хотите оттрассировать вызовы функций и посмотреть обработку конфигурационного файла, в качесте уровня отладки Вы устанавливаете сумму интересующих уровней (в этом случае 65). Для более детальной информации см. <ldap.h>. Заметьте, что для того, чтобы все эти уровни (за исключением статистики) были доступны, slapd должен быть собран с ключем -DLDAP_DEBUG.

```
-f <filename>
```

Этот параметр определяет альтернативный конфигурационный файл для slapd.

```
-i
```

Этот параметр указывает slapd, что он запускается из под inetd, а не как автономный сервер. В следующем разделе Вы найдетее больше информации о работе slapd из под inetd.

```
-p <port>
```

Этот параметр определяет альтернативный TCP порт, на котором slapd должен отвечать на соединения. Порт по умолчанию - 389.

4.2 Запуск slapd в качестве автономного демона

В общем slapd запускается следующим образом:

```
$(ETCDIR)/slapd [<option>]*
```

где ETCDIR имеет значение, которое Вы задали в файле Make-common или передали скрипту configureперед сборкой, а <option> один из вышеописанных параметров. Если Вы только не задали уровень отладки, slapd автоматически отсоединяется от контрольного терминала, и используя fork переходит в режим демона. При такой форме запуска, slapd можно передавать любые из рассмотренных выше параметров, как то TCP порт, другой конфигурационный файл и т.д.
Вот пример запуска slapd:

```
$(ETCDIR)/slapd -f /home/malere/myslapd.conf -d 255
```

4.3 Запуск slapd из под inetd

Перво-наперво убедитесь в том, что запуск из под inetd действительно необходим. Если Вы используете LDBM, это не так. Если Вы работаете в требовательной к ресурсам системе, то накладные расходы от работы из под inetd, также переводят эту идею в разряд плохой. В противном случае, для достижения цели необходимо сделать два шага.
Шаг 1, добавить следующую строку в файл /etc/services:

```
ldap 389 # ldap directory service
```

Шаг 2, добавить следующую строку в файл /etc/inetd.conf:

```
ldap stream tcp nowait nobody $(ETCDIR)/slapd slapd -i
```

где ETCDIR имеет значение, которое Вы задали в файле Make-common или передали скрипту configureперед сборкой. В качестве финального акта пошлите HUP процессу inetd, и Вы получили то, что хотели.

5 Создание и Поддержка Базы Данных

Этот раздел рассказывает о том, как содать базу данных slapd с нуля. Есть два способа создания базы дынных. Первый, Вы создаете базу данных интерактивно используя LDAP. В этом методе Вы просто запускаете slapd и добавляете записи исользуя Ваш любимый LDAP клиент. Это подходит для относительно небольших баз данных (несколько сотен или тысяча записей, в зависимости от Ваших потребностей).
Второй метод подразумевает создание базы независимо от slapd, используя утилиты генерации индекса. Этот метод наилучший, если Вам нужно создать много тысяч записей, что отняло бы неприемлемо много времени используя первый метод, или если Вам нужно, чтобы база была недоступна на время создания.

5.1 Создание Базы в Интерактивном Режиме /online/

Дистрибутив OpenLDAP поставляется с утилитой ldapadd, используемой для добавления статей во время работы LDAP сервера. Если Вы выбрали интерактивный метод создания базы, Вы можете использовать утилиту ldapadd для добавления записей. После добавления первых записей, Вы все еще можете используя ldapadd добавлять следующие записи. Вы должны убедиться, что в файле slapd.conf установлены следующие конфигурационные параметры:

```
suffix <dn>
```

Как описано в 3 (разделе 3), этот параметр указывает место в иерархии каталога, куда будут добавляться записи. Вы должны установить этот параметр в DN корня поддерева, которое Вы собираетесь создать.
Например: suffix "o=TUDelft, c=NL"
Вы также должны указать директорию, куда будут помещаться вновь созданные индексные файлы:

```
directory <directory>
```

Например: directory /usr/local/tudelft
Вам также нужно предоствить возможность присоединиться к slapd под кем-то с правами на добавление записей. Это делается посредством установки следующих параметров в определении базы данных:

```
rootdn <dn>
rootpw <passwd>   /* Помните, что здесь нужно использовать зашифрованный пароль !!! */
```

Эти параметры задают DN и пароль, которые могут использоваться для аутентификации входа в базу под "администратором"(т.е. позволяющего делать все). Определенные здесь DN и пароль будут работать всегда, независимо от того существует ли в действительности такая точка входа или имеет данный пароль. Это решает проблему "курицы и яйца", т.е. как провести аутентификацию и добавление записей, если никаких записей еще вообще нет.
И, наконец, Вы должны удостовериться, что определение базы данных содержит желаемые определения индексов:

```
index {<attrlist> | default} [pres,eq,approx,sub,none]
```

Например, для индексирования атрибутов cn, sn, uid и objectclass, можно использовать следующие записи:

```
index cn,sn,uid
index objectclass pres,eq
index default none
```

Как только Вы получили желаемую конфигурацию, стартуете slapd, соединяетесь с ним используя Ваш любимый LDAP клиент, и начинаете добавлять записи. Например, чтобы добавить запись TUDelft, а за ней Postmaster, используя утилиту ldapadd, Вы можете создать файл со следующим содержимым:

```
o=TUDelft, c=NL
objectClass=organization
description=Technical University of Delft Netherlands

cn=Postmaster, o=TUDelft, c=NL
objectClass=organizationalRole
cn=Postmaster
description= TUDelft postmaster - postmaster@tudelft.nl
```

а затем выполнить команду, чтобы создать запись в базе:

```
ldapadd -f /tmp/newentry -D "cn=Manager, o=TUDelft, c=NL" -w secret
```

В приведенной выше комманде предполагается, что вы установили rootdn в "cn=Manager, o=TUDelft, c=NL"и rootpw в "secret". Если Вы не хотите набирать пароль открытым текстом в коммандной строке, используйте параметр -W для ldapadd вместо -w "password". После ввода Вам выведется приглашение ввести пароль:

```
ldapadd -f /tmp/newentry -D "cn=Manager, o=TUDelft, c=NL" -W
Enter LDAP Password :
```

5.2 Создание Базы без Сервера /offline/

Второй метод создание базы данных - сделать это без сервера, используя средства генерации индекса, описанные ниже. Этот метод лучше, если Вам нужно создать много тысяч записей, что заняло бы неоправданно много времени, если использовать описанный выше первый метод. Эти утилиты читают конфигурационный файл slapd и входной файл LDIF, содержащий текстовое представление записей, которые нужно добавить. Они создают непосредственно индексный файл LDBM. Есть несколько важных конфигурационных параметров, которые Вам нужно установить заранее в определении базы данных в конфигурационном файле:

```
suffix <dn>
```

Как описано в предыдущем разделе, этот параметр устанавливает начальное положение в Каталоге, где будут создаваться записи. Вы должны задать здесь DN корня поддерева, который Вы хотите создать.
Например: suffix "o=TUDelft, c=NL"
Вам нужно определить директорию в файловой системе, где будут созадны файлы индекса:

```
directory <directory>
```

Например: directory /usr/local/tudelft
Далее, Вы возможно захотите увеличить размер внутреннего кеша, используемого каждым открытым индексным файлом. Для лучшего быстродействия во время создания индекса, весь индекс должен размещаться в памяти. Если Ваши данные слишком большие для этого, или у Вас слишком мало памяти, Вы можете все равно задать его достаточно большим, поскольку есть виртуальная память. Этот размер задается следующим параметром:

```
dbcachesize <integer>
```

Например: dbcachesize 50000000
Это создаст кеш размером 50 MB, что достаточно много (в Университете Мичигана база данных содержит около 125 тыс. записей, и наибольший файл индекса имеет около 45 MB). Поэкспериментируйте немного с этим параметром и уровнем паралельности (объясняется ниже), чтобы подобрать наилучшие величины для Вашей системы. Не забудьте вернуть значение этого параметра назад сразу после создания индекса и до запуска slapd.
Наконец, Вам нужно определить, какой индекс Вы хотите создать. Это производится одним или несколькими индесными параметрами.

```
index {<attrlist> | default} [pres,eq,approx,sub,none]
```

Например:
index cn,sn,uid pres,eq,approx
index default none
Это создаст индексы presence/наличия/, equality/эквивалентности/ и approximate/подобия/ для атрибутов cn, sn, и uid, и никаких индексов для всех остальных атрибутов. См. конфигурационный файл в 3 (разделе 3) для более детального описания этого параметра.
Как только Вы отконфигурировали все по Вашему уподобанию, можете создавать индексы запустив утилиту ldif2ldbm:

```
ldif2ldbm -i <inputfile> -f <slapdconfigfile>
[-d <debuglevel>] [-j <integer>]
[-n <databasenumber>] [-e <etcdir>]
```

Аргументы имеют следующее толкование:

```
-i <inputfile>
```

Определяет входной LDIF файл, содержащий в текстовом формате записи, которые нужно добавить.

```
-f <slapdconfigfile>
```

Определяет конфигурационный файл slapd, в котором берется информация о том, где создавать индексные файлы, какие индексы создавать и т.п.

```
-d <debuglevel>
```

Включает протоколирование уровня, определенного в <debuglevel>. Уровни протоколирования такие же как и для slapd (см. 4.1 (раздел 4.1)).

```
-j <integer>
```

Необязательный аргумент, указывающий максимальное количество паралельных процессов, которые должны стартовать при построении индексов. По умолчанию 1. Если установить значение больше чем 1, ldif2ldbm во время построения индексов создаст максимально возможное количество подпроцессов. Для построения каждого индекса атрибута создается отдельный процесс. Паралельное выполнение этих процессов может существенно увеличить скорость создания индексов, но не следует создавать слишком много подпроцессов, поскольку каждый из них отбирает ресурсы диска и памяти, так что может быть и обратный эффект.

```
-n <databasenumber>
```

Необязательный аргумент, который определяет для какой по счету базы данных из конфигурационного файла производится построение индексов. Первая определенная в конфигурационном файле база данных обозначается "1", вторая - "2", и т.д. По умолчанию, используется первая ldbm база данных из определенных в конфигурационном файле.

```
-e <etcdir>
```

Необязательный аргумент, который определяет директорию в файловой системе, где ldif2ldbm сможет найти другие необходимые для работы конверторы (ldif2index и подобные). По умолчанию используется инсталяционная директория заданная скрипту configure. Посмотрите на пример использования утилиты ldif2ldbm:

```
/usr/local/sbin/ldif2ldbm -i new_entries -f myslapd.conf
```

5.3 Подробнее Об LDIF Формате

LDAP Data Interchange Format (LDIF) - формат обмена данными LDAP - используется для представления записей LDAP в простом текстовом виде. Общая форма записи следующая:

```
[<id>]
dn: <distinguished name>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
...
```

где <id> необязательный идентификатор записи (положительное целое число). Обычно Вы не задаете <id>, позволяя утилитам создания базы сделать это за Вас. Утилита ldbmcat создает формат LDIF, котороый включает <id>, так чтобы вновь созданные индексные файлы были корректными. Любая строка может быть продолжена в следующей, если последняя начинается с пробела или символа табуляции.
Например:

```
dn: cn=Barbara J Jensen,
 o=University of Michigan, c=US
```

Значения многозначных (векторных) атрибутов задаются в отдельных строках.
Например:

```
cn: Barbara J Jensen
cn: Babs Jensen
```

Если <attrvalue> содержит непечетные символы, или начинается с пробела или двоеточия ':', то после <attrtype> ставится двойное двоеточие и значение кодируется используя base64. Например, значение "begins with a space"будет закодировано следующим образом:

```
cn:: IGJlZ2lucyB3aXRoIGEgc3BhY2U=
```

Записи внутри одного LDIF файла разделяются пустыми строками. Вот пример LDIF файла с тремя записями:

```
dn: cn=Barbara J Jensen, o=University of Michigan, c=US
cn: Barbara J Jensen
cn: Babs Jensen
objectclass: person
sn: Jensen

dn: cn=Bjorn J Jensen, o=University of Michigan, c=US
cn: Bjorn J Jensen
```

```
cn: Bjorn Jensen
objectclass: person
sn: Jensen

dn: cn=Jennifer J Jensen, o=University of Michigan, c=US
cn: Jennifer J Jensen
cn: Jennifer Jensen
objectclass: person
sn: Jensen
jpegPhoto:: /9j/4AAQSkZJRgABAAAAAQABAAD/2wBDABALD
A4MChAODQ4SERATGCgaGBYWGDEjJR0oOjM9PDkzODdASFxOQ
ERXRTc4UG1RV19iZ2hnPk1xeXBkeFxlZ2P/2wBDARESEhgVG
...
```

Заметьте, что jpegPhoto в записи Jennifer Jensen закодировано используя base64. Для создания файлов LDIF можно использовать утилиту ldif, поставляемую с дистрибутивом OpenLDAP.
Имейте в виду, что крайние пробелы в значениях не убираются при интерпретации LDIF файла, также как и не сжимается многократное повторение пробела внутри значения. Так что если Вы не хотите увидить их в Ваших данных, не помещайте их туда.

5.4 Утилиты ldapsearch, ldapdelete и ldapmodify

ldapsearch - ldapsearch является по сути пользовательским интерфейсом к библиотечному вызову ldap_search(3). Вы можете использовать эту утилиту для поиска записей в базе данных Вашего LDAP сервера.
Вызов ldapsearch имеет следующую семантику (объяснения парамтеров смотрите в страницах man по ldapsearch):

```
ldapsearch [-n] [-u] [-v] [-k] [-K] [-t] [-A] [-B] [-L] [-R] [-d debuglevel] [-F sep] [-f
[-D binddn] [-W] [-w bindpasswd] [-h ldaphost] [-p ldapport] [-b searchbase] [-s base|on
[-a never|always|search|find] [-l timelimit] [-z sizelimit] filter [attrs...]
```

ldapsearch открывает соединение к LDAP серверу, связывается, и производит поиск используя заданный фильтр. Фильтр должен соответствовать строковому предствлению LDAP фильтров, как определено в RFC 1558. Если ldapsearch находит одну или более записей, выбираются атрибуты заданные attrs, и записи со значенями этих атрибутов выдаются на консоль. Если атрибуты не заданы, то возвращаются значения для всех атрибутов.
Вот несколько примеров использования ldapsearch:

```
ldapsearch -b 'o=TUDelft,c=NL' 'objectclass=*'

ldapsearch -b 'o=TUDelft,c=NL' 'cn=Rene van Leuken'

ldasearch -u -b 'o=TUDelft,c=NL' 'cn=Luiz Malere' sn mail
```

Ключ -b устанавливает searchbase (исходную точку - поддерево Каталога - откуда начнется поиск), а ключ -u включает легковоспринимаемый (форматированный) вывод результатов поиска.
ldapdelete - ldapdelete является пользовательским интерфейсом к библиотечному вызову ldap_delete(3). Используйте эту утилиту для удаления записей из базы данных на LDAP сервере.
Семантика вызова ldapdelete следующая (для более детального объяснения параметров смотрите соответствующую страницу man):

```
ldapdelete [-n] [-v] [-k] [-K] [-c] [-d debuglevel] [-f file] [-D binddn] [-W] [-w p
[-h ldaphost] [-p ldapport] [dn]...
```

ldapdelete открывает соединение к LDAP серверу, связывается и удаляет одну или более записей. Если задан один или более аргументов DN, записи с такими Выделенными Именами удаляются.

Каждый DN должен быть строковым представлением DN, как описано в RFC 1779. Если DNы как аргументы не присутствуют, ldapdelete запускается в режиме считывания их со стандартного входа (или из файла, если задан ключ -f).
Вот некоторые примеры использования ldapdelete:

```
ldapdelete 'cn=Luiz Malere,o=TUDelft,c=NL'

ldapdelete -v 'cn=Rene van Leuken,o=TUDelft,c=NL' -D 'cn=Luiz Malere,o=TUDelft,c=NL' -W
```

Ключ -v устанавливает режим детализированного ответа, ключ -D устанавливает binddn (dn для аутентификации) и ключ -W включает приглашение для ввода пароля.
ldapmodify - ldapmodify является пользовательским интерфейсом к библиотечным вызовам ldap_modify(3) и ldap_add(3). Используйте эту утилиту для изменения записей в базе данных Вашего LDAP сервера.
ldapmodify имеет следующую семантику вызова (для более детального описания смотрите соответствующие страницы man):

```
ldapmodify [-a] [-b] [-c] [-r] [-n] [-v] [-k] [-d debuglevel] [-D binddn] [-W] [-w p
[-h ldaphost] [-p ldapport] [-f file]

ldapadd [-b] [-c] [-r] [-n] [-v] [-k] [-K] [-d debuglevel] [-D binddn] [-w passwd] [-h lda
[-p ldapport] [-f file]
```

ldapadd реализован как жесткая ссылка на ldapmodify. Когда ldapmodify запускается как ldapadd, ключ -a (добавить новую запись) устанавливается автоматически. ldapmodify открывает соединение к LDAP серверу, связывается и изменяет или добавляет записи. Сами записи считываются из стандартного ввода или из файла, указанного с помощью ключа -f.
Вот некоторые примеры использования ldapmodify:
Предположим, что файл /tmp/entrymods существует и имеет следующее содержание:

```
dn: cn=Modify Me, o=University of Michigan, c=US
changetype: modify
replace: mail
mail: modme@terminator.rs.itd.umich.edu
-
add: title
title: Grand Poobah
-
add: jpegPhoto
jpegPhoto: /tmp/modme.jpeg
-
delete: description
-
```

Тогда комманда:

```
ldapmodify -b -r -f /tmp/entrymods
```

заменит содержимое атрибута mail записи "Modify Me"значением "modme@terminator.rs.itd.umich.edu", добавит title со значением "Grand Poobah", и содержимое файла /tmp/modme.jpeg как jpegPhoto, и полностью удалит атрибут description.
Описанные выше изменения можно также сделать используя более старый формат ввода ldapmodify:

```
cn=Modify Me, o=University of Michigan, c=US
mail=modme@terminator.rs.itd.umich.edu
+title=Grand Poobah
+jpegPhoto=/tmp/modme.jpeg
-description
```

и следующую команду:

```
ldapmodify -b -r -f /tmp/entrymods
```

Предполагая, что файл /tmp/newentry существует и имеет следующее содержимое:

```
dn: cn=Barbara Jensen, o=University of Michigan, c=US
objectClass: person
cn: Barbara Jensen
cn: Babs Jensen
sn: Jensen
title: the world's most famous manager
mail: bjensen@terminator.rs.itd.umich.edu
uid: bjensen
```

комманда:

```
ldapadd -f /tmp/entrymods
```

добавит запись Barbara Jensen в базу данных.
Предполагая, что файл /tmp/newentry существует и имеет следующее содержимое:

```
dn: cn=Barbara Jensen, o=University of Michigan, c=US
changetype: delete
```

комманда:

```
ldapmodify -f /tmp/entrymods
```

удалит запись Babs Jensen's из базы.
Ключ -f указывает файл (прочитать информацию о вносимых изменениях из файла, а не стандартного ввода), ключ -b устанавливает признак двоичных данных (любые значения начинающиеся с '/' во входном файле интерпретируются как двоичные), ключ -r устанавливает признак замены (заменить существующие значения значениями по умолчанию).

6 Дополнительная информация и свойства

В этом разделе Вы найдете информацию о Netscape Address Book,- клиенте LDAP, который можно использовать для работы с Вашим Каталогом. Также описано как реализовать Roaming Access/блуждающий доступ?/ используя Netscape Navigator версии 4.5 и Ваш LDAP сервер. О Roaming Access было много разговоров в OpenLDAP конференциях, поскольку эта функциональность еще не полностью реализована. Большинству людей не нравится способ, которым Netscape Navigator оперирует с LDAP сервером во время скачивания и закачивания данных. Поэтому, если после прочтения этого раздела Вы найдете, что Roaming Access работает не так как Вам хотелось бы, не огорчайтесь, через это прошло уже много людей. Основная цель описания этой функциональности здесь, в большей мере, показать людям возможности протокола LDAP. И напоследок Вы получите некоторую информацию о безопасной остановке сервера slapd, а также о его журнальных файлах.

6.1 Roaming Access

Цель сервиса Roaming Access состоит в предоставлении доступа используя Netscape Navigator и LDAP сервера к Вашим закладкам, настройкам, почтовым фильтрам и т.п. из любой точки Сети, где-бы Вы ни находились. Это добольно приятное свойство,- предствьте, что откуда-бы Вы не вошли в Сеть, Вы получаете свои собственные настройки браузера. Если Вы собираетесь путешествовать и Вам нужен будет валютный сервер, который находится в Ваших локальных закладках, не беспокойтесь, закачиваете закладки и другие конфигурационный файлы на LDAP сервер, и после этого Вы можете получить их из любого места в Сети.
Для реализации сервиса Roaming Access Вам нужно сделать следующее:

- Изменить Ваш файл описания атрибутов
- Изменить Ваш файл описания объектных классов
- Включить профили в Ваш LDIF файл
- Настроить Netscape Navigator для использования LDAP сервера как Roaming Access сервера
- Перезапустить LDAP сервер с новыми настройками.

- Изменение файла атрбутов: Вам нужно добавить новые атрбуты к существующему списку атрибутов в файле slapd.at.conf (это файл, который Вы включаете в Ваш slapd.conf, который обычно находится в /usr/local/etc/openldap):

```
attribute       nsLIPtrURL              ces
attribute       nsLIPrefs               ces
attribute       nsLIProfileName         cis
attribute       nsLIData                bin
attribute       nsLIElementType         cis
attribute       nsLIServerType          cis
attribute       nsLIVersion             cis
```

- Изменение файла объектных классов: Вам также нужно добавить несколько новых классов к Вашему slapd.oc.conf (это другой файл, который обычно включается в slapd.conf и находится в /usr/local/etc/openldap) для активации roaming access:

```
objectclass nsLIPtr
requires
        objectclass
allows
        nsliptrurl,
        owner

objectclass nsLIProfile
requires
        objectclass,
        nsliprofilename
allows
        nsliprefs,
        uid,
        owner

objectclass nsLIProfileElement
requires
        objectclass,
        nslielementtype
allows
        owner,
        nslidata,
        nsliversion

objectclass nsLIServer
requires
        objectclass,
        serverhostname
allows
        description,
        cn,
        nsserverport,
        nsliservertype,
        serverroot
```

- Изменение файла LDIF: Теперь Вам нужно модифицировать Ваш LDIF файл, путем добавления профилей для каждого пользователя, который хочет использовать функциональность Roaming Access в Netscape. Вот простой пример файла LDIF с записями профилей:

```
dn: o=myOrg,c=NL
o: myOrg
objectclass: organization

dn: cn=seallers,ou=People,o=myOrg,c=NL
cn: seallers
userpassword: myPassword
objectclass: top
objectclass: person

dn: nsLIProfileName=seallers,ou=Roaming,o=myOrg,c=NL
changetype: add
objectclass: top
owner: cn=seallers,ou=People,o=myOrg,c=NL
objectclass: top
objectclass: nsLIProfile
```

Следующим шагом настраиваем Netscape для активации Roaming Access к Вашему LDAP серверу. Сделайте следующее:
- Перейдите в меню Edit -> Preferences -> Roaming User
Теперь Вы должны активировать Roaming Access для текущего профиля, щелкнув на соответствующем этой опции элементе выбора.
- Введите имя пользователя в соответствующую строку редактирования, например john
Раскройте иерархию закладки Roaming User в левой части панели Preferences для получения доступа к вложенным под-опциям Roaming Access.
- Щелкните на Server Information и активируйте параметр LDAP Server, а затем введите следующую информацию в соответствующие поля:

```
Address: ldap://<имя_вашего_сервера>/nsLIProfileName=$USERID,ou=Roaming,o=myOrg,c=NL

User DN: cn=$USERID,ou=People,o=myOrg,c=NL
```

ВНИМАНИЕ : Netscape автоматически заменяет переменную $USERID именем профиля, который Вы выбрали до запуска браузера. Так если вы выбрали профиль seallers, он подставит вместо $USERID значение seallers, если Вы выбрали профиль gonzales - он подставит gonzales. Если Вы не сильны в профилях, запустите приложение Profile Manager, которое поставляется с дистрибутивом Netscape Navigator. Это приложение разработано для поддержки многопользовательского режима использования браузера на одном компьютере, так чтобы каждый из пользователей имел свои собственные настройки браузера.
Финальным шагом есть перезапуск сервера, смотрите разделы 6.6 (6) и 4 (4) о том, как это сделать безопасно и чтобы после этого все работало.

6.2 Netscape Address Book

Как только Вы настроили и запустили LDAP сервер, Вы (и не только Вы) можете общаться с ним с помощью любого LDAP клиента (например, утилиты ldapsearch). Одним из таких клиентов является Netscape Address Book. Он доступен начиная с версии 4.x Netscape, но для стабильной работы с LDAP сервером лучше использовать версии выше 4.5.
Выполните следующее:

```
Запустите Netscape Navigator -> перейдите в Communicator Menu -> Address Book
```

Netscape Address Book запустится с настройками LDAP каталогов по умолчанию. Конечно, для роботы Вы должны добавить свой LDAP каталог в настройки Netscape Address Book.

```
Перейдите в File Menu -> New Directory
```

Заполните соответствующие поля информацией о Вашем LDAP серевере.
Например:

```
- Description : TUDelft

- LDAP Server : dutedin.et.tudelft.nl

- Server Root : o=TUDelft, c=NL
```

По умолчанию LDAP отвечает на 389 порту, не изменяйте его, если Вы конечно не изменили его, когда настраивали Ваш сервер.
Теперь отошлите на Ваш сервер простой запрос, используя поле Show Names Containing, или сложный,- используя кнопку Search for...

6.3 LDAP Migration Tools /Конвертеры/

LDAP Migration Tools - это коллекция скриптов на Perl, которые можно использовать для конвертирования различных конфигурационных файлов в LDIF формат. Эти скрипты предоствлены PADL Software Ltd и я рекомендую Вам ознакомиться с лицензией до их использования, не смотря на то, что они бесплатные. Если Вы планируете использовать Ваш сервер для аутентификации пользователей, эти скрипты будут очень полезны. Используйте эти утилиты для преобразования Ваших NIS или passwd архивов в LDIF формат, чтобы сделать их совместимыми с LDAP сервером. Их можно также использовать для перевода пользователей, групп, псевдонимов, машин, сетевых групп, сетей, протоколов, RPCs и сервисов из существующих серверов имен (NIS, NetInfo или файлов из /etc) в LDIF формат. Выкачать LDAP Migration Tools и получить дополнительную информацию по ним можно в Сети по следующему адресу:
http://www.padl.com/tools.html
Пакет скриптов поставляется с README файлом, а названия выполняемых файлов интуитивны. Так что сначала просмотрите README, а затем применяйте скрипты.

6.4 Аутентификация с помощью LDAP

Ваш LDAP сервер может аутентифицировать пользователей используя механизм называемый PAM (Pluggable Authentication Modules /подлючамые модули аутентификации/). С самого начала существования UNIX аутентификация пользователя проводилась посредством проверки введенного им пароля на соответствие зашифрованному паролю, хранимому в файле /etc/passwd.
Так было вначале. С того времени появилось много новых популярных методов аутентификации пользователей, включая более сложные альтернативы файлу /etc/passwd, а также специальные устройства, как Smart карты. Проблема состоит в том, что как только разработана новая схема аутентификации, для её поддержки необходимо переписать все соответствующие программы (login, ftpd и т.п.). PAM предоставляет способ разработки программ, которые не зависят от схемы аутентификации. Чтобы эти программы работали, к ним должны во время выполнения /run-time/ подключаться "модули аутентификации".
Модули аутентификации для LDAP доступны по следующему адресу:
http://www.padl.com/pam_ldap.html
Здесь я предполагаю, что Ваш дистрибутив Linux уже подготовлен для использования PAM. Если нет, сходите сюда: *http://www.kernel.org/pub/linux/libs/pam*. Конечно, различные дистрибутивы Linux используют разные параметры по умолчанию, установленные для PAM. Обычно файлы конфигурации PAM находятся в директории `/etc/pam.d/`. Там Вы найдете файл со всеми сервисами запущенными в Вашей системе. Например, если Вы хотите, чтобы Ваш LDAP серевер аутентивацировал пользователей после старта системы, Вы должны сделать Вашу систему PAM совместимой (см. начало параграфа), установить LDAP PAM модуль и отредактировать файл login в директории конфигурации PAM (/etc/pam.d) следующим образом:

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_nologin.so
auth       sufficient   /lib/security/pam_ldap.so
auth       required     /lib/security/pam_unix_auth.so try_first_pass
account    sufficient   /lib/security/pam_ldap.so
account    required     /lib/security/pam_unix_acct.so
password   required     /lib/security/pam_cracklib.so
password   required     /lib/security/pam_ldap.so
password   required     /lib/security/pam_pwdb.so use_first_pass
session    required     /lib/security/pam_unix_session.so
```

6.5 GUI программы для LDAP

- Kldap

Kldap графический LDAP клиет написанный для KDE. Kldap имеет хороший пользовательский интерфейс и способен показывать любую информацию сохраненную в Вашем Каталоге. Вы можете оценить некоторые проспекты и скачать само приложение по адресу:
http://www.mountpoint.ch/oliver/kldap

- GQ

GQ - другой графический LDAP клиент, который имеет более простой интерфейс, и написан для Gnome. Конечно, он работает и под KDE, в общем как и Kldap под Gnome. Вот адрес для скачивания и получения более детальной информации:
http://biot.com/gq/

6.6 Завершение LDAP сервера

Чтобы безопасно завершить работу slapd, Вы должны выполнить комманду, подобную следующей:

```
kill -TERM `cat $(ETCDIR)/slapd.pid`
```

Завершение slapd более кардинальным методом может привести к повреждению его LDBM базы, поскольку ей до завершения обычно нужно сохранить различные буферы. Заметьте, что slapd записывает свой pid в файл slapd.pid в директорию, которую Вы задали в файле slapd.conf, например: /usr/local/var/slapd.pid
Вы можете изменить местонахождение этого pid файла изменив переменную SLAPD_PIDFILE в файле include/ldapconfig.h.edit.
slapd также записывает свои аргументы в файл slapd.args в директорию, заданную в конфигурационном файле slapd.conf, например /usr/local/var/slapd.args.
Вы можете изменить местонахождение файла аргументов изменив переменную SLAPD_ARGSFILE в файле include/ldapconfig.h.edit.

6.7 Журналы

slapd использует syslog(8) сервис для генерации журналов. Сервис syslog(8) имеет пользователя по умолчанию - LOCAL4, но значения также разрешены и от LOCAL0, LOCAL1, и до LOCAL7.
Чтобы включить генерацию журналов, Вам нужно соответствующим образом отредактировать файл syslog.conf, который обычно находится в директории /etc.
Добавьте следующую запись:

```
local4.*     /usr/adm/ldalog
```

Эта запись использует пользователя по умолчанию LOCAL4 для сервиса syslog. Если Вы не сильны в приведенном синтаксисе, обратитесь к страницам man по syslog, syslog.conf и syslogd. Еслы Вы хотите изменить пользователя по умолчанию или установить уровень протоколирования, Вы можете оперировать следующими параметрами во время запуска slapd:

```
-s syslog-level
```

Этот параметр указывает slapd уровень отладочной информации, который должен протоколироваться в журнал сервиса syslog(8). Уровень описывает критичность сообщения, и задается одним из следующих упорядоченных (от высшего к низшему) ключевых слов: emerg, alert, crit, err, warning, notice, info, и debug.
Например:

```
slapd -f myslapd.conf -s debug
```

```
-l syslog-local-user
```

Выбирает локального пользователя для сервиса syslog(8). Значение может быть одним из LOCAL0, LOCAL1, и так до LOCAL7. По умолчанию LOCAL4. Однако этот параметр разрешен только на системах, поддерживающих локальных пользователей сервиса syslog(8).
Теперь рассмотрите сгенерированные журналы, они могут помочь решить много проблем, связанных с запросами, изменениями, связыванием и т.п.

7 Ссылки

В этом разделе Вы найдете ссылки на дополнительную документацию по LDAP: полезные URL, книги, RFC.

7.1 URLs

Здесь приведены URL, по которым содержится очень полезная информация об LDAP. В большинстве своем этот HOWTO и был содан по информации с этих серверов, так что если после чтения этого документа у Вас возникла потребность в дополнительной информации, скорее всего Вы найдете её здесь:

- LDAP страница Мичиганского Университета: *http://www.umich.edu/~dirsvcs/ldap/index.html*
- Страница документации по LDAP Мичиганского Университета: *http://www.umich.edu/~dirsvcs/ldap/doc/*
- Реализация Roaming Access *http://help.netscape.com/products/client/communicator/manual_roaming2.html*
- Настройка параметров LDAP для Communicator 4.5 : *http://developer.netscape.com/docs/manuals/communicc*
- Сервис Каталогов Linux *http://www.rage.net/ldap/*

7.2 Книги

Далее приведены наиболее популярные и полезные книги по LDAP:

- "Implementing LDAP"by Mark Wilcox
- "LDAP: Programming Directory-Enabled Applications with Lightweight Directory Access Protocol"by Howes and Smith
- "Understanding and Deploying LDAP Directory Servers"by Howes, Smith, and Good

7.3 RFCs

Ниже приведены RFC имеющие отношение к разработке и специфицированию LDAP.

- RFC 1558: A String Representation of LDAP Search Filters
- RFC 1777: Lightweight Directory Access Protocol
- RFC 1778: The String Representation of Standard Attribute Syntaxes
- RFC 1779: A String Representation of Distinguished Names
- RFC 1781: Using the OSI Directory to Achieve User Friendly Naming
- RFC 1798: Connectionless LDAP
- RFC 1823: The LDAP Application Programming Interface
- RFC 1959: An LDAP URL Format
- RFC 1960: A String Representation of LDAP Search Filters
- RFC 2251: Lightweight Directory Access Protocol (v3)
- RFC 2307: LDAP as a Network Information Service