ОРДЕНА ЛЕНИНА
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
им. М.В.Келдыша
Российской Академии Наук

М.К.Валиев, Е.Л.Китаев, М.И.Слепенков

Служба директорий LDAP как инструментальное средство для создания распределенных информационных систем

Москва, 2000
Настоящая работа поддержана Российским фондом фундаментальных исследований,
гранты № 99-01-00374 и № 99-01-00375

Аннотация

В работе дается подробное, систематизированное описание службы директорий LDAP, включая: структуру информации, представляемой в директории, принятый в LDAP способ организации и идентификации объектов, архитектуру и средства реализации распределенных директорий, основные операции, применимые к представленным в директории объектам, и способы защиты информации в директории от неавторизованного доступа. Работа базируется на материалах IETF, стандартизующих модели директорий X.500 и LDAP, а также на документации по различным реализациям LDAP-серверов и LDAP-клиентов, разработанных Мичиганским университетом, компаниями Netscape, IBM и др. Проведенный анализ позволил достаточно подробно описать такие важные аспекты LDAP как механизмы организации и администрирования распределенной директории, которые в настоящий момент не стандартизованы и поэтому часто опускаются в литературе.

LDAP directory service as a tool for implementation of distributed information systems

Abstract

The paper presents a detailed, systematized description of the LDAP directory service which includes: the structure of information represented in the directory, methods of the object organization and identification in LDAP, the architecture and possible implementation approaches to the distributed directory, basic operations applicable to the directory objects, and means for information protection against unauthorized access to the directory. The paper is based on the OSI and IETF standards for X.500 and LDAP, and documentation manuals on LDAP-servers and LDAP-clients developed by the Michigan University and companies Netscape, IBM, etc. The analysis enables to describe some important aspects of LDAP such as the mechanisms for organization and administering the distributed directory, which are not currently included into the standard and are not well described in the literature.

СОДЕРЖАНИЕ

Введение

Увеличение производительности глобальных сетей на основе применения новых телекоммуникационных технологий и появления гигабитных каналов связи открывает реальные перспективы создания новых классов глобальных информационных систем, способных обеспечить более тесную и эффективную интеграцию географически распределенных информационных ресурсов сети. Такие системы будут способны осуществлять сбор и актуализацию разнородных данных, предоставляя пользователям и сервисным программным компонентам сети оперативный доступ к аккумулируемой в них информации. Эффективность доступа к информации будет обеспечиваться развитыми поисковыми средствами, включающими возможности выполнения многокритериального поиска с учетом структуры представления данных, а также распределенных запросов.

В качестве примеров предметных областей, в которых такая глобальная степень интеграции информации является существенной, можно назвать географические информационные системы, справочные системы типа «Желтых страниц», в которых аккумулируется адресная информация об организациях, их штатной структуре, работающем персонале и т.п. К числу важных применений глобальных информационных систем относится также создание интегрированного описания вычислительных и коммуникационных ресурсов глобальной сети.

Последняя задача стала особенно актуальной в связи с начатыми в последние годы работами по созданию метакомпьютера - глобальной, географически распределенной вычислительной системы, в которой можно

будет объединить ресурсы многих в настоящее время изолированных вычислительных центров. Наиболее сложные проблемы создания интегрированного описания ресурсов связаны с высокой степенью распределенности информации. Поставщиками информации в такой системе являются вычислительные центры (сайты метакомпьютера), в задачу которых входит предоставление в определенном формате и последующая актуализация информации о тех имеющихся у них вычислительных и сетевых ресурсах, которые выделяются ими в распоряжение пользователей метакомпьютера. Информация может поступать также и из других источников, например, от специализированных узлов, осуществляющих мониторинг сетей. Таким образом, общее число поставщиков информации в подобной системе может исчисляться тысячами и даже десятками тысяч.

В настоящее время одним из наиболее развитых в и перспективных инструментов, способных обеспечить создание распределенных информационных систем в глобальной сети Интернет, является служба директорий LDAP (Lightweight Directory Access Protocol). К числу основных особенностей LDAP относится принятая в этой модели нетрадиционная организация данных, дающая ряд принципиальных преимуществ при представлении информации в распределенной директории. Кроме того, в отличие от традиционных инструментов (в частности систем баз данных), в службах директорий LDAP решены вопросы организации распределенной инфраструктуры, необходимой для представления и ведения информации (установка и обслуживание LDAP-серверов, организация репликации данных и т.п.).

Настоящая работа посвящена систематизированному рассмотрению архитектуры службы директорий LDAP. Вначале даются общие сведения о LDAP, описывается информационная модель директории и раскрываются принципы организации распределенных директорий. Далее приводится основной набор функций протокола LDAP, при этом подробно рассматривается операция поиска в директории. Наконец, описывается административная модель LDAP, включающая в себя представление пользователей в системе и разграничение доступа к хранимой информации. Особое внимание в процессе изложения уделяется таким важным аспектам LDAP как механизмы организации и администрирования распределенной директории, которые в настоящий момент не стандартизованы и поэтому часто опускаются в литературе.

При изложении архитектуры LDAP мы будем ориентироваться на задачу создания распределенной директории, описывающей вычислительные и коммуникационные ресурсы метакомпьютерной системы. Более подробно вопросы использования службы директорий LDAP для представления метаинформации в глобальных вычислительных системах будут рассмотрены нами в работе [ВКС2000].

1 Общие сведения о службе директорий LDAP

Служба директорий LDAP является облегченной версией директорий X.500, разработанных в 1988 г. в рамках ISO. Стандарт директорий X.500 определяет информационную модель директории, а также протокол для доступа к директории, получивший название DAP (Directory Access Protocol). К сожалению, протокол DAP оказался слишком сложным и ресурсоемким для реализации в сетях TCP/IP. В связи с этим комитет IETF начал работать над облегченной (LDAP) версией этого протокола, чтобы позволить клиентам глобальной сети Интернет вводить, удалять, изменять и извлекать информацию, содержащуюся в директориях X.500. В результате этого появился стандарт LDAP-1, описанный в RFC1487. В стандарте LDAP-1 были введены понятия LDAP-клиента и шлюза, через который отрабатываются клиентские запросы, адресованные к директории X.500.

Вскоре был создан стандарт LDAP-2 (RFC1777 и др.). Если LDAP-1 представляет собой, по сути, внешний интерфейс к директориям X.500, то в LDAP-2 появляется возможность создания самостоятельных, автономных по отношению к X.500 директорий. С этой целью в LDAP-2 вводится понятие LDAP-сервера. В настоящее время используется стандарт LDAP-3, описанный в RFC2251 и др. В стандарте LDAP-3 устраняются многие ограничения и недостатки предыдущей версии протокола, что превращает LDAP в мощный инструмент для создания распределенных информационных систем.

В стандарте [RFC2256] определена так называемая общая часть схемы (common schema) директории LDAP, совместимая со схемой директории X.500. Общая схема устанавливается на всех LDAP-серверах и ориентирована на представление данных об организациях, структуре их подразделений и работающем персонале. При этом администраторы LDAP-серверов имеют возможность расширять общую часть схемы, добавляя атрибуты и порождая новые классы объектов.

Основной особенностью модели директорий является принятая в ней организация данных, сильно отличающая ее от традиционных моделей реляционных или объектно-ориентированных баз данных. Организация баз данных основана на использовании понятия отношение (или класс объектов), с помощью которого содержимое базы данных разбивается на поименованные группы однородных записей (или объектов). Такая группа записей или объектов, адресуемых соответствующим именем отношения (или класса объектов), является естественной областью действия операций над базами данных. Например, поиск информации (команда SELECT в SQL) всегда выполняется в рамках определенных отношений, имена которых перечисляются в части FROM этого запроса. Аналогично, при назначении прав доступа к данным (команда GRANT языка SQL) в качестве параметра указывается имя отношения, к которому предоставляется доступ.

В основе устройства модели директорий лежит идея размещения объектов в вершинах специальной древовидной структуры – информационного дерева

директории. Естественным способом группирования объектов в такой модели является выделение множества объектов, размещенных в вершинах одного поддерева. В такую группу могут входить как однородные, так и разнородные объекты. Для именования такого фрагмента директории достаточно указать идентификатор корня соответствующего поддерева. В качестве области действия базовых операций LDAP (поиск, обновление, администрирование и т.п.) принимаются фрагменты указанного вида (т.е. поддеревья директории). Например, при выполнении запроса в директории необходимо указать базовую вершину, начиная с которой выполняется поиск. Естественными единицами администрирования в модели директорий также являются поддеревья.

Модель директорий LDAP является довольно сложной, поэтому для ее лучшего описания полезно выделить следующие принципиальные аспекты (взгляды), которые будут последовательно рассматриваться в данной работе.

- **Модель информационной базы директории** - описывает структуру информации, представляемой в директории LDAP.
- **Модель информационного дерева директории** - описывает принятый в директории LDAP способ организации и идентификации информации.
- **Модель распределенной директории** - описывает архитектуру и средства реализации распределенных в сети директорий.
- **Функциональная модель** - описывает операции, применимые к представленной в директории информации.
- **Модель обеспечения безопасности** - определяет способы защиты информации в директории от неавторизованного доступа.

2 Модель информационной базы директории

Информация размещается в директории в виде *объектов*, иначе называемых элементами директории (directory entries). По смыслу, каждый содержащийся в директории объект отражает факт существования некоторой сущности реального мира: организации, человека, сервера и т.д. Объект характеризуется своим набором *атрибутов*. Отдельный атрибут обладает *именем* и может принимать одно или несколько *значений* определенного типа. В модели директорий LDAP используется фиксированный набор типов значений. Тип определяет формат допустимых значений, а также способы сопоставления и упорядочивания значений. Наиболее важные из типов, на которые мы будем ссылаться в дальнейшем, перечислены ниже:

- **cis** - строка символов. При сравнении строк этого типа регистры символов (т.е. заглавные и прописные буквы) не учитываются.
- **ces** - строка символов. При сравнении строк этого типа регистры символов учитываются.
- **int** - символьное представление целого числа
- **tel** - телефонный номер

- **dn** - уникальное имя (ссылка)
- **bin** - бинарная информация, представленная (закодированная) в виде строки символов

Обратим внимание, что значения любого типа представляются в LDAP в виде последовательностей символов, содержащих буквы латинского алфавита (заглавные и прописные), цифры, знаки пунктуации, пробел и символ перехода на новую строку. При этом значения с различными типами могут иметь одинаковое символьное представление. Так, 1234567 может представлять собой строку символов (типа cis или ces), целое число или телефонный номер.

Некоторые наиболее часто используемые в директориях LDAP имена атрибутов имеют общепринятые синонимы - краткие имена, которые можно использовать как эквиваленты соответствующих полных, длинных имен. Так, **o** используется как синоним для **organization** (название организации), **ou** – для **organizationalUnit** (название подразделения организации), **c** – для **country** (страна), **cn** - для **commonName** (общепринятое имя), **sn** - для **surname** (фамилия), **hn** - для **hostName** (DNS-имя хоста). С целью иллюстрации рассмотренных понятий перечислим некоторые атрибуты объекта, который может представлять в директории Институт Прикладной Математики (левая колонка), а также объекта, представляющего вычислитель spp1600.keldysh.ru (правая колонка). Обратим внимание, что в названии организации и доменном имени (domainName) заданы по два значения.

ObjectClass: organization
o: Keldysh Institute for Applied Mathematics
o: KIAM
postalAddress: Miusskaya sq. 4, Moscow, Russia
domainName: keldysh.ru
domainName: kiam.ru

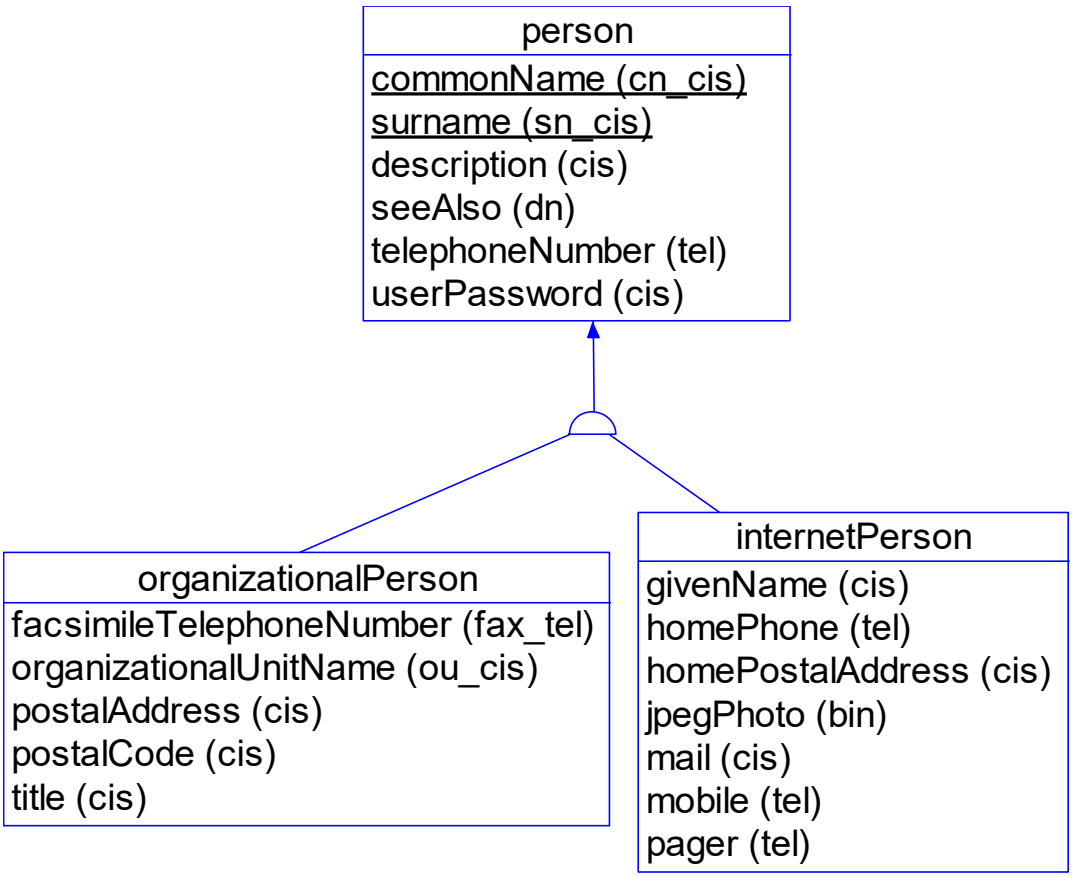
objectClass: computeResource
hn: spp1600.keldysh.ru
manufacturer: Convex
processorCount: 8
OSType: unix
OSVersion: hp ux 10.01

Для каждого объекта директории устанавливается, к какому классу или классам он принадлежит. Эта принадлежность отображается в форме задания обязательного атрибута с именем objectClass, значениями которого являются имена классов.

Классы объектов определяются в *схеме* директории. В объявлении класса задается его имя и перечисляются атрибуты, которые используются в объектах данного класса, а также фиксируются те из них, задание значений в которых обязательно. Кроме того, могут быть указаны один или несколько родительских классов, определения которых будут наследоваться данным классом. Все атрибуты, упоминаемые как обязательные в родительских

классах, становятся обязательными и в порождаемом классе. То же самое относится и к необязательным атрибутам.

Особенностью представления схемы в LDAP является то, что здесь определение атрибутов (устанавливающее соответствие между именем



атрибута и типом допустимых для него значений) не включается в объявления классов, а дается самостоятельно. Таким образом в схеме сначала определяется набор атрибутов, который затем используется при построении классов объектов. Такой способ определения схемы принят в LDAP для того, чтобы исключить ситуацию появления в классах одинаково поименованных атрибутов с различными типами значений.

Как было отмечено выше, объект может принадлежать к нескольким классам одновременно. Обязательными атрибутами такого объекта считаются все те, которые объявлены обязательными хотя бы в одном из этих классов. Соответственно, необязательным атрибутом является тот, который объявлен как необязательный во всех классах, включенных в объявление объекта.

Для иллюстрации сказанного рассмотрим следующий пример, в котором объявляются три класса, показанные на следующей диаграмме: person (человек) и organizationalPerson (должностное лицо) и internetPerson (пользователь Интернет).

Класс person определяет наиболее общие атрибуты, которые присущи любому человеку: полное имя (commonName, например, фамилия и имя);

фамилия (surname); произвольное текстовое описание (description); DN-ссылка на другие объекты, представляющие этого человека в директории (seeAlso); телефонный номер (telephoneNumber); пароль для входа в директорию в зашифрованном виде (userPassword). Атрибуты commonName и surname являются обязательными (выделены на диаграмме подчеркиванием).

Класс person является родительским для двух производных от него классов. Класс organizationalPerson добавляет дополнительные атрибуты, характеризующие человека как должностное лицо: номер факса, наименование подразделения организации, где он работает, почтовый адрес и код, должность. С другой стороны класс internetPerson характеризует человека как пользователя Интернет, вводя атрибуты: имя (givenName), домашние адрес и телефон, фотография, адрес электронной почты, номера мобильного телефона и пейджера.

Как можно заметить, схема определяет структуру представления информации в директории, номенклатуру отображаемых понятий, а также атрибутный состав информационных объектов. Таким образом схема играет важную роль, позволяя создателям директории ограничить круг представимой в ней информации. Одновременно, для пользователей директории схема полезна в информационном плане - как словарь терминов, которыми можно оперировать при поиске информации в директории. Ниже (в заключительной части ***раздела 4***) мы вернемся к понятию схемы еще раз, чтобы описать принятый в LDAP подход к реализации схемы в случае распределенной директории.

3 Модель информационного дерева директории

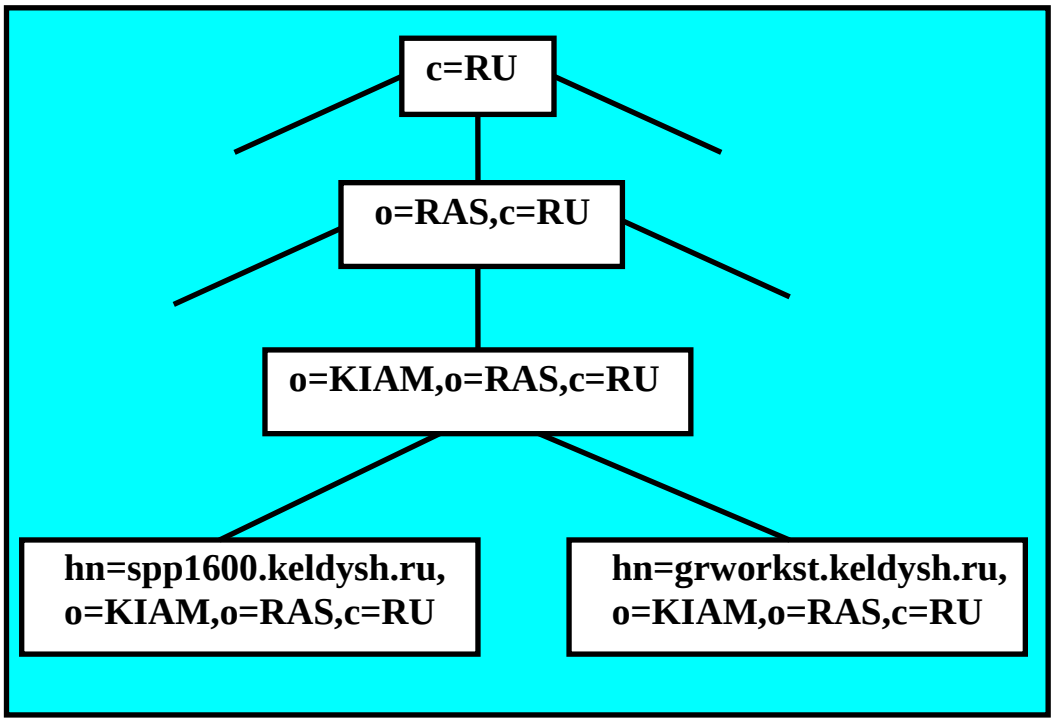
Объекты директории организуются в виде древовидной структуры, называемой информационным деревом директории. Подчиненность объектов в дереве директории, как правило, отражает географическую, организационную, политическую или иную подчиненность сущностей реального мира, соответствующих объектам директории. Например, дерево директории, где будет храниться информация об организациях, может иметь следующую структуру. На первом уровне дерева располагаются объекты, представляющие страны мира. На втором уровне располагаются объекты, отображающие организации. На более глубоких уровнях идут подразделения организаций, работающий в них персонал, имеющиеся у организаций и их подразделений вычислительные ресурсы.

При создании нового объекта в директории необходимо определить его атрибуты (задав имена и значения), указать существующий объект, которому он будет подчинен, а также зафиксировать *относительное уникальное имя* (RDN - relative distinguished name) этого объекта. Относительное имя задается посредством указания одного или нескольких атрибутов и их значений, имеющихся у объекта (синтаксически, в виде *<атрибут1>=<значение1>+…+<атрибут n>=<значение n>*).

Если атрибут обладает несколькими значениями, то выбирается одно из них. Например, если организация имеет два наименования - полное и сокращенное, Keldysh Institute for Applied Mathematics и KIAM, то в RDN может войти только одно из них. Относительное имя должно быть уникальным по отношению к относительным именам других объектов, подчиненных тому же родителю.

Уникальные имена объектов (DN - distinguished name), позволяющие идентифицировать объекты в директории, вне зависимости от их подчиненности, образуются посредством конкатенации относительных уникальных имен объектов на пути от корня до идентифицируемого объекта (аналогично полному имени файла в иерерхической файловой системе). Синтаксически уникальные имена записываются в обратной последовательности - от объекта к корню. Для разделения RDN'ов в уникальном имени используется запятая.

На приведенной схеме представлен пример фрагмента информационного



дерева, в котором присутствуют следующие объекты. На верхнем уровне иерархии находится объект с DN-именем cn=RU, используемый для группирования организаций по территориальной принадлежности (в данном случае - Российских организаций). На следующих уровнях располагается объект, представляющий РАН и, в подчинении ему, объект - ИПМ. В поддереве ИПМ содержится два объекта, которые описывают вычислители, принадлежащие ИПМ.

4 Модель распределенной директории

Одним из основных достоинств LDAP является возможность организации *распределенных директорий*. В распределенной директории различные

фрагменты информационного дерева физически размещаются и обслуживаются разными LDAP-серверами. Такая архитектура позволяет распределять между организациями, заинтересованными в размещении в директории своей информации, накладные расходы по установке и администрированию LDAP-серверов. Например, в зависимости от имеющихся организационных договоренностей, информация об организациях, входящих в систему Российской Академии Наук, может быть размещена на центральном сервере РАН, либо на собственных серверах организаций.

Для идентификации LDAP-серверов используется их Интернет-адрес, представляемый в формате ldap://*URL*:*port*, где *URL* задает IP-адрес или DNS-адрес сервера, а *port* - номер порта (обычно 389). Например, сервер ИПМ может иметь адрес ldap://ldap.keldysh.ru:389.

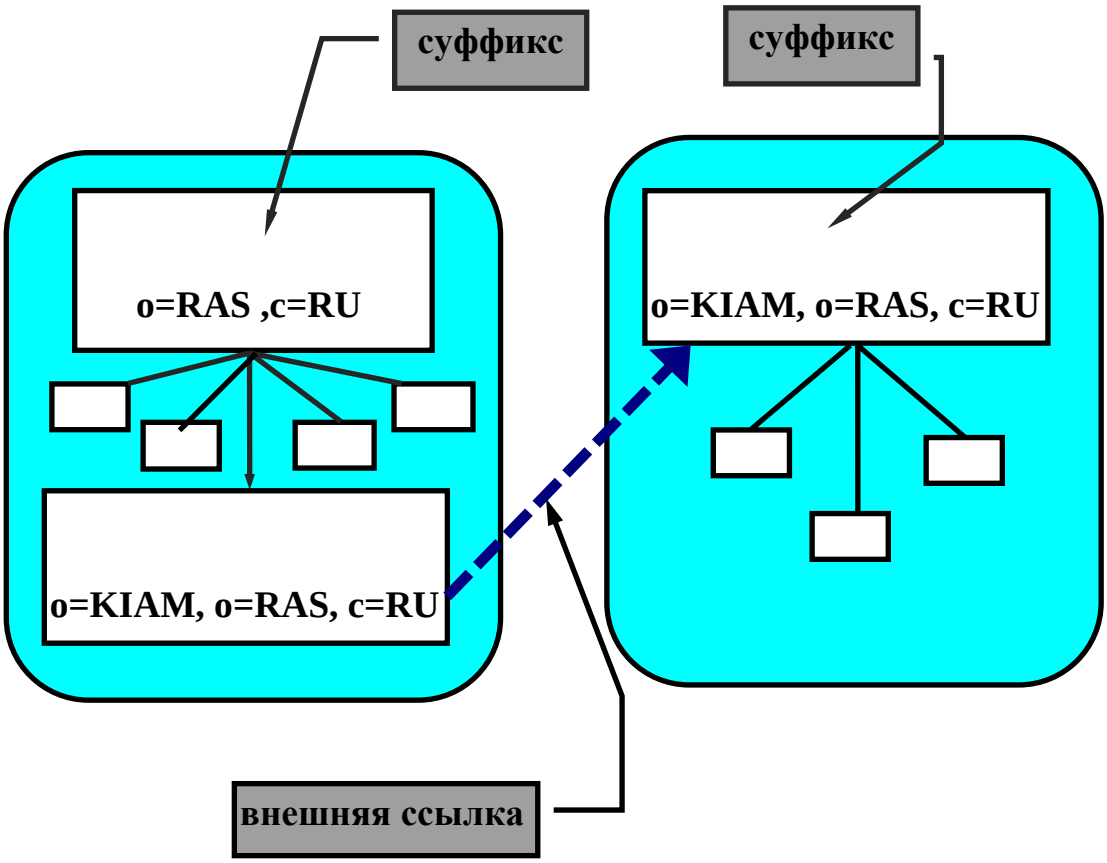
Добавление к идентификатору LDAP-сервера уникального имени (DN) объекта директории, например, ldap://ldap.keldysh.ru:389/o=KIAM, o=RAS, c=RU используется для идентифицикации объектов распределенной директории с учетом мест их размещения.

Каждый LDAP-сервер обладает собственной внутренней базой данных, в которой непосредственно хранится один или несколько фрагментов директории. Логически каждый фрагмент представляет собой поддерево информационного дерева директории. Идентификация каждого поддерева осуществляется при помощи задания *суффикса* - уникального имени (DN) объекта, являющегося корнем этого поддерева. Определение набора суффиксов (т.е. обслуживаемых сервером фрагментов директории) входит в обязанности администратора LDAP-сервера.

Каждое поддерево, размещаемое на LDAP-сервере и определяемое своим суффиксом, может содержать специальные вершины типа *внешних ссылок* (referrals), адресующих объекты, размещенные на других LDAP-серверах. По смыслу, с помощью внешних ссылок можно исключить из хранимого фрагмента одно или несколько поддеревьев, предоставив при этом адреса тех LDAP-серверов, которые реально отвечают за ведение этих поддеревьев. Таким образом, внешние ссылки LDAP позволяют представить единое, в концептуальном смысле, информационное дерево директории. Внешняя ссылка оформляется в виде специального объекта, принадлежащего к классу Referral. Объекты этого класса обладают атрибутом ref, в котором и задается адрес объекта на соответствующем сервере в виде ldap://*URL*:*port/DN*.

Для иллюстрации определенных выше понятий: суффиксов и ссылок, рассмотрим следующий пример. Предположим, что в распределенную директорию, где будет размещаться информация о научных организациях, должны быть включены два сервера, один из которых имеет адрес ldap://ldap.keldysh.ru и принадлежит ИПМ им. Келдыша, а другой, с адресом ldap://ldap.ras.ru - Российской академии наук. Тогда, администратор сервера РАН должен определить на своем сервере суффикс o=RAS, c=RU, а администратор сервера ИПМ - суффикс o=KIAM, o=RAS, c=RU. Поскольку ИПМ собирается размещать информацию о своих ресурсах на собственном

сервере, то администратор сервера РАН должен создать у себя объект типа Referral с DN o=KIAM, o=RAS, c=RU, указав в качестве атрибута ref адрес ldap://ldap.keldysh.ru:389 /o=KIAM, o=RAS, c=RU. Описанная конфигурация изображена на следующей схеме.



Отдельный LDAP-сервер может отработать только такие запросы пользователей, которые относятся к обслуживаемым им фрагментам директории. Если поступивший запрос относится к другим фрагментам и, следовательно, не может быть выполнен данным сервером, то в ответ на такой запрос сообщается адрес вышестоящего в иерархии серверов LDAP-сервера. Эта возможность обеспечивается за счет указания *умалчиваемой внешней ссылки* на LDAP-сервере, в которой фиксируется адрес вышестоящего сервера. Например, на сервере ИПМ следует установить умалчиваемую ссылку ldap://ldap.ras.ru:389. В результате сервер ИМП будет перенаправлять серверу РАН все те запросы, которые касаются других организаций.

В распределенных информационных системах, базирующихся на модели LDAP, для обеспечения эффективности поиска информации используется репликация данных. Репликация - это механизм, посредством которого фрагменты директории, расположенные на одних серверах, автоматически копируются на другие. Использование репликации позволяет существенно повысить доступность данных, обеспечивая устойчивость к сбоям, более высокую производительность, сбалансированную загрузку серверов и т.п. Например, установление взаимной репликации между серверами РАН и ИПМ,

в результате которой сервер РАН будет иметь копию данных с сервера ИПМ и наоборот, позволит каждому из них обрабатыватывать без переадресации все запросы, относящиеся к информации об организациях РАН. При такой организации выход одного из серверов из строя не будет иметь критических последствий.

При установке репликационного механизма различают сервер-поставщик данных и сервер-получатель. Любой LDAP-сервер может одновременно выступать как в роли поставщика, так и в роли получателя. Для того, чтобы сервер мог использоваться как поставщик, на нем устанавливается специальный журнал, в котором фиксируются все изменения, происходящие в данных. С помощью журналов минимизируются объемы пересылаемых данных - не вся директория, а только изменения, произошедшие со времени предыдущей репликации.

Существуют два варианта инициализации репликации - со стороны сервера-поставщика, и со стороны сервера-получателя. В первом случае, на поставщике заводится список рассылки, в котором указываются адреса серверов получателей и суффиксы реплицируемых фрагментов директории. Во втором случае, когда репликация инициализируется со стороны сервера-получателя, сам получатель будет обращаться к поставщику, считывая из журнала соответствующие изменения. С точки зрения администрирования репликации оба варианта подобны друг другу, позволяя в конечном итоге обновлять данные на серверах-получателях по истечении определенного временного интервала. Однако, при инициализации репликации со стороны сервера-поставщика возможна установка такого режима репликации, когда каждое изменение в поставляемых данных немедленно рассылается всем получателям. Такой режим важен в случаях, когда необходима актуализация данных в режиме реального времени.

Каждый реплицируемый объект, размещаемый на сервере-получателе, помечается как объект-копия. При этом в объекте-копии фиксируется адрес LDAP-сервера, который является хозяином этого объекта - т.е. сервера, на котором объект был создан и который обладает исключительным правом его изменения. При попытке пользователя выполнить операции по изменению или удалению объекта-копии, такой запрос всегда пересылается серверу - хозяину объекта. Эта парадигма (у объекта всегда один хозяин) позволяет организовать достаточно сложные схемы репликации, например каскадную репликацию.

В завершение остановимся на вопросах реализации схемы в распределенной директории. На каждом LDAP-сервере в специальном объекте с DN-именем **cn=schema** перечисляются определения типов атрибутов и классов объектов, составляющие схему директории, поддерживаемую данным LDAP-сервером. При установке сервера в качестве начальной схемы загружается общепринятая схема (common schema), определенная в стандарте [RFC2256]. Администратору сервера предоставляется возможность модификации схемы, путем: добавления новых атрибутов и классов;

модификации существующих классов; удаления атрибутов и классов (при этом рекомендуется воздержаться от изменения атрибутов и классов общепринятой, начальной схемы). Эта возможность позволяет администратору расширять общепринятую схему, обеспечивая представление объектов, не поддерживаемых в общепринятой схеме (например, описание вычислительных ресурсов организации).

С целью облегчения модели LDAP в службе директорий не введены какие-либо штатные механизмы, обеспечивающие синхронизацию схем, установленных на различных LDAP-серверах распределенной директории. Результатом такого рассогласования схемы может стать снижение качества представления хранящейся в директории информации (например, одно и то же понятие будет отображено разными классами или атрибутами с разными именами), что естественно повлечет за собой снижение эффективности поиска в директории. В связи с этим общепринятая практика, обеспечивающая успешное создание распределенных информационных приложений на базе службы директорий LDAP, заключается в том, чтобы на начальном этапе разработки зафиксировать схему, отражающую понятия предметной области. Дальнейшая эволюция схемы предполагает согласованное внесение изменений на всех LDAP-серверах, входящих в состав прикладной системы.

5 Функциональная модель

Функциональные операции, поддерживаемые службой директорий LDAP, базируются на клиент-серверной архитектуре и позволяют: открыть соединение клиента с сервером, произвести аутентификацию клиента, выполнить поиск и модификацию объектов в директории, закрыть соединение при завершении работы. Для большинства современных систем программирования (C++, Perl, Java и т.д.) имеются прикладные программные интерфейсы (API), реализующие вышеназванные базовые функции клиента LDAP.

Отметим, что сами по себе операции поиска и модификации данных не являются распределенными, т.е. область их действия ограничивается фрагментом директории, размещенном на том LDAP-сервере, с которым у клиента открыто соединение. При этом в интерфейс этих операций заложена возможность перенаправления запросов клиента на другие LDAP-сервера, которая реализуется за счет использования возвращаемых сервером *перенаправляющих ссылок* (continuation references). Подобный механизм позволяет создавать клиентские приложения, обеспечивающие прозрачность при выполнении операции поиска и модификации в распределенной директории.

Из имеющихся в настоящий момент клиентских приложений следует отметить набор командных утилит (ldapsearch, ldapmodify и др.), реализованный в большинстве операционных систем (Unix, Windows), которые позволяют осуществлять поиск и модификацию данных в

распределенной директории. Кроме того, имеются специализированные LDAP-броузеры, позволяющие в графическом режиме осуществлять навигацию и поиск в директориях, а также выполнять операции модификации. В последние версии Интернет-броузеров (Netscape Navigator, Microsoft Internet Explorer) также встроены возможности поиска в распределенных директориях LDAP.

В следующей таблице приведен список наиболее важных операций в LDAP, утилит и дана их краткая характеристика в соответствии с [RFC1823]. Отметим, что все операции могут выполняться как в синхронном, так и в асинхронном режимах.

Операция	**Утилита**	**Описание**
Search	ldapsearch	Осуществляет поиск в директории и возвращает результат в виде списка найденных объектов
Add	ldapadd	Обеспечивает возможность добавления новых объектов в директорию. При добавлении объекта задаются его атрибуты и их значения, а также определяется уникальное имя (DN) объекта. Добавлять можно только листовые объекты.
Delete	ldapdelete	Выполняет удаление объектов из директории. При этом возможно удаление только листовых объектов. Удаление поддерева может быть реализовано только путем последовательного применения этой операции.
Modify	ldapmodify	Обеспечивает возможность модификации (добавления, удаления, замены) атрибутов и значений атрибутов у существующих объектов директории.
Bind		Выполняет открытие сессии соединения клиента с сервером. При открытии сессии клиент может предоставить данные, аутентифицирующие его на сервере.
Unbind		Закрывает соединение между клиентом и сервером
Abandon		Завершает выполнение асинхронной операции, запущенной на сервере.

В настоящем разделе подробно рассматривается операция поиска, которая довольно интересна и нетривиальна в связи с иерархической организацией модели директорий. Для выполнения поиска необходимо задать три обязательных параметра: *базовый DN, глубину поиска* и *фильтр*. Запросы

клиента выполняются на том LDAP-сервере, с которым предварительно установлено соединение при помощи операции Bind.

Базовый DN и глубина поиска, указанные в запросе, определяют фрагмент директории, в котором будет выполняться поиск. Базовый DN задает уникальное имя объекта, принимаемого в качестве начальной вершины поискового фрагмента (поддерева) в информационном дереве директории. Глубина поиска определяет количество уровней в поисковом поддереве, на которых будет выполняться запрос. Допустимы три варианта задания значения этого параметра: SUBTREE (поиск во всем поддереве), SINGLE (поиск среди всех объектов, непосредственно подчиненных базовому) и BASE (только базовый объект). Задание глубины BASE, чаще всего, используется для получения конкретного объекта директории.

Фильтр специфицирует тот критерий, которому должны удовлетворить все объекты, возвращаемые в качестве результата поиска. Фильтр представляет собой набор элементарных предикатов, соединенных при помощи булевых операторов. Элементарный предикат имеет вид *имя-атрибута оператор значение*. Например, чтобы найти организацию с названием KIAM, можно задать предикат o=KIAM.

В качестве операторов используются = (совпадение), >= (больше или равно), <= (меньше или равно). Значение, указываемое в предикате = , может содержать специальный символ *. В этом случае проверяется не точное равенство значений, а их сопоставимость. Например, предикату o=K*M* удовлетворяют организации, название которых начинается с буквы K и включает букву M. Если атрибут какого-либо объекта обладает несколькими значениями, то такой объект будет включен в результирующее множество, если хотя бы одно из его значений удовлетворяет предикату.

Предикаты в фильтре могут соединяться посредством булевых операторов - отрицания (НЕ - !), логического объединения (ИЛИ - |) и пересечения (И - &). Для записи операторов используется префиксная нотация, имеющая следующее синтаксическое определение.

фильтр := **(предикат) | (! фильтр) | (& фильтр...) | (| фильтр...)**

Для иллюстрации понятий, описанных выше, рассмотрим следующий пример. Допустим, мы хотим найти в РАН и ее организациях вычислители с числом процессоров больше трех, на которых установлена операционная система UNIX. Это можно сделать при помощи следующего запроса.

БАЗА: **o=RAS, c=RU**
ГЛУБИНА: **SUBTREE**
ФИЛЬТР: **(& (objectClass=computeResource) (processorCount >= 4) (OSType=unix))**

Результаты поиска оформляются в виде списка найденных объектов. В список включаются DN объектов, атрибуты и их значения. Для уменьшения размеров выдачи, при вызове функции поиска можно указать перечень возвращаемых атрибутов. Если такой перечень не задается, то по-умолчанию возвращаются все атрибуты найденных объектов. Другим способом органичения размеров выдачи является возможность указания максимально допустимого числа объектов в результирующем списке.

Если поисковый фрагмент на LDAP-сервере, к которому изначально адресован запрос, содержит объекты класса Referral (внешняя ссылка), то помимо списка результатов сервер возвращает клиенту все обнаруженные им внешние ссылки. Таким образом, клиенту дается возможность продолжить поиск на других LDAP-серверах, с использованием того же критерия. Современные утилиты, устанавливаемые на клиентских местах, реализуют механизм автоматической переадресации запросов в случаях получения в ответ внешних ссылок. Формируемый ими список результатов образуется путем объединения ответов от всех серверов, опрошенных при выполнении подобного распределенного запроса. Для предотвращения зацикливания обычно устанавливается предел числа переходов по внешним ссылкам. Подобный подход позволяет обеспечить необходимую прозрачность поиска в распределенных директориях.

Для иллюстрации механизма выполнения запросов обратимся к примеру распределенной директории, приведенному в предыдущем разделе. Напомним, что в примере использовалось два LDAP-сервера: сервер РАН с суффиксом o=RAS,c=RU и сервер ИПМ с суффиксом o=KIAM,o=RAS, c=RU. На сервере РАН объект с DN-именем o=KIAM,o=RAS, c=RU представлен ссылкой на сервер ИПМ. В свою очередь сервер ИПМ имеет умалчиваемую внешнюю ссылку на сервер РАН.

Если к серверу РАН адресуется запрос по базе o=RAS,c=RU с глубиной SUBTREE, то в ответ возвращается множество найденных на сервере объектов, а также перенаправляющая ссылка на сервер ИПМ. Клиент перенаправляет запрос на сервер ИПМ, указывая в качестве базы o=KIAM,o=RAS, c=RU. Множество объектов, найденных на сервере ИПМ, добавляется к результирующему множеству, полученному на предыдущем шаге от сервера РАН. С другой стороны, если к серверу РАН адресуется запрос по базе o=KIAM,o=RAS, c=RU, то он просто возвращает ссылку, перенаправляющую поиск на сервер ИПМ.

Рассмотрим другую ситуацию, когда запрос по базе o=RAS,c=RU адресуется к серверу ИПМ. В этом случае сервер ИПМ возвращает клиенту свою умалчиваемую внешнюю ссылку, т.е. запрос переадресуется серверу РАН, где выполняется по описанной выше схеме. Наконец, если к серверу ИПМ адресуется запрос по базе o=KIAM,o=RAS, c=RU, то он выполняет его, возвращая точное результирующее множество (без перенаправляющих ссылок).

Отметим, что независимо от того, с какого из двух серверов начинается выполнение одного и того же запроса (с одинаковой базой, глубиной и фильтром), результирующее множество всегда будет одинаковым.

6 Модель обеспечения безопасности

Обеспечение безопасности в LDAP имеет две основных стороны. Во-первых, LDAP определяет протокол взаимодействия клиента и сервера. Для обеспечения безопасности этого протокола принят традиционный подход, основанный на использовании механизмов Secure Socket Layer (SSL). Слой SSL обеспечивает взаимную аутентификацию клиента и сервера; гарантирует отсутствие искажений в информации, передаваемой по сети; а также позволяет сохранить приватность информации при передаче. В целом, использование SSL в LDAP постороено по аналогии с другими безопасными сетевыми протоколами, например https и ftps. В настоящей работе этот аспект модели безопасности LDAP подробно рассматриваться не будет.

С другой стороны, служба директорий LDAP представляет собой инструмент для реализации информационных систем и, поэтому, она должна обеспечивать безопасность хранящейся в системе информации, защищая ее от несанкционированного доступа. Разграничение доступа в LDAP в настоящий момент не стандартизовано и различные реализации LDAP могут иметь существенные отличия в этом аспекте. Однако существуют общие принципы разграничения доступа, которые прослеживаются во всех реализациях. При изложении этих принципов мы будем основываться на механизме обеспечения безопасности, принятом в службе директорий LDAP компании Netscape. Разграничение доступа базируется на следующих понятиях:

- представление пользователей и групп пользователей
- способы аутентификации пользователей в момент входа в систему
- представление правил разграничения доступа и используемая схема авторизации действий, выполняемых пользователями

Пользователи и группы пользователей в LDAP представляются в виде объектов директории. По большому счету эти объекты ничем не отличаются от прочих, информационных объектов. Для представления пользователей используется класс person или производные от него классы inetOrgPerson, organizationalPerson и др. Объекты этих классов используются для представления людей и могут, помимо прочей информации, иметь атрибуты userPassword и userCertificate. Если какой-либо из этих атрибутов у объекта задан (или заданы оба), то такой объект LDAP рассматривает как зарегистрированного пользователя. Уникальное имя (DN) этого объекта выполняет роль имени для входа (login name).

В атрибуте userPassword можно в зашифрованном виде задать пароль в формате "(способ-шифрования) зашифрованный-пароль", например *userPassword: {sha}FTSLQhxXpA05* . В атрибуте userCertificate можно

сохранить пользовательский сертификат в бинарном формате. С помощью этих атрибутов LDAP-сервер аутентифицирует пользователя в момент открытия соединения (операция Bind). При этом имеется два варианта аутентификации. Простая аутентификация требует указания DN пользователя и пароля в момент соединения. При этом проверяется совпадение заданного при соединении пароля с паролем в атрибуте userPassword соответствующего объекта. Сложная аутентификация основана на использовании механизмов SSL. При сложной аутентификации проверяется наличие у пользователя инициирующего соединение закрытого ключа, соответствующего сертификату в атрибуте userCertificate.

В соответствии с общепринятыми соглашениями для представления администраторов директории, обладающих правами изменения закрепленных за ними фрагментов, используют объекты с RDN **cn=Directory Manager**. Эти объекты размещаются непосредственно в том фрагменте, за ведение которого администратор отвечает. Например, администратор фрагмента директории РАН будет иметь DN **cn=Directory Manager,o=RAS,c=RU**, а администратор ИПМ - имя **cn=Directory Manager,o=KIAM,o=RAS,c=RU**.

Группы пользователей представляются при помощи объектов класса groupOfNames или производных классов, которые обладают атрибутом cn (commonName), используемым для задания имени группы, а также множественным атрибутом member, в котором перечисляются уникальные имена (DN) пользователей - членов группы. Для идентификации группы, аналогичным образом, используется DN соответствующего ей объекта.

Информация, определяющая правила разграничения доступа, представляется в директориях LDAP с помощью специальных атрибутов aci (Access Control Information instruction), которые могут быть добавлены к любому объекту директории. Атрибут aci является множественным, т.е. может принимать несколько значений, каждое из которых определяет самостоятельную инструкцию разрешающего или запрещающего характера.

Инструкция aci имеет следующий формат:

< target >(<permission><bind rule>)(<permission><bind rule>)...

Выражение *target* определяет некоторое множество объектов директории, составляющих область действия данной инструкции. При этом в *target* можно включать только объекты, входящие в поддерево, вершиной которого является объект с данным aci. Область действия можно задать посредством прямого указания DN объекта, либо шаблона DN'ов с использованием символа *. Например **target="o=*,o=RAS,c=RU"** задает все организации, входящие в РАН. Другой способ задания области действия инструкции основан на указании поискового фильтра в формате, принятом в операции поиска (Search), в виде **targetfilter=фильтр**. В *target* могут быть также перечислены атрибуты объектов, к которым относится инструкция. Для этого используется нотация **targetattr=список-атрибутов.**

Выражение *permission* определяет тип инструкции (разрешить или запретить) и вид операции (read, write, add, delete, search). Например, запрет на модификацию задается в виде **deny(write,add,delete)**.

При определении прав доступа к директории учитывается ряд параметров, которые характеризуют текущее соединение (открытое в результате выполнения операции Bind) клиента с сервером. В части *bind rule* (правила соединения) aci-инструкции могут быть введены ограничения на параметры соединения, при удовлетворении которых доступ к директории будет разрешен или запрещен. Правила соединения могут быть простыми, например, такое правило может разрешать доступ к директории всем пользователям, принадлежащим к определенной группе. С другой стороны, правила соединения могут быть более сложными. Например, доступ к директории может быть предоставлен пользователям определенной группы в период времени с 8 утра до 5 вечера, при этом допускается обращение к директории только с компьютеров с определенными IP-адресами. Формально, в части *bind rule* можно задать следующие виды ограничений или их комбинацию:

- ограничение на идентификаторы пользователей, которое принимает вид **userdn=DN** или **userdn != DN**, где **DN** - уникальное имя, шаблон имен (с символом *) или одно из специальных имен: Anyone - любой, в том числе анонимный, пользователь; All - любой зарегистрированный пользователь; и др.
- ограничение группы, к которой должен принадлежать пользователь, установивший соединение. Имеет вид **groupdn = DN** или **groupdn != DN**, где **DN** - имя группы или шаблон.
- ограничение сетевого адреса компьютера, с которого устанавливается соединение, которое задается в виде **ip = IP-адрес** (**ip != IP-адрес**) или **dns = DNS-адрес** (**dns = DNS-адрес**). В позиции адреса можно указывать точный адрес либо шаблон.
- ограничение даты соединения **dateofweek** и времени соединения **timeofday**. Например, **dayofweek = "Sun, Mon, Tue"** или **timeofday >= "1200"**.
- ограничение на используемый при соединении способ аутентификации: простая аутентификация **authmethod = "simple"** или сложная **authmethod = "ssl"**.

Для иллюстрации управления доступом к директории LDAP рассмотрим следующий пример. Чтобы разрешить всем пользователям директории, включая анонимных, доступ на чтение и поиск в фрагменте директории, представляющем информацию об ИПМ и, одновременно, предоставить право ведения этой информации администратору директории, нужно установить следующие aci-инструкции у объекта с dn "o=KIAM,o=RAS,c=RU":

aci: (target = "ldap:///o=KIAM,o=RAS,c=RU ") (targetattr=*) allow (read,search)

userdn = ldap:///anyone"
aci: (target = "ldap:///o= o=KIAM,o=RAS,c=RU") (targetattr=*) allow (add,modify,delete)
userdn = "ldap:///cn=Directory Manager,o=KIAM,o=RAS,c=RU "

При авторизации доступа клиента к определенному объекту директории LDAP-сервер использует следующую схему. Сначала суммируются все aci-инструкции, приписанные к самому объекту, а также ко всем объектам, расположенным в информационном дереве директории на пути, соединяющем данный объект с корнем директории. После этого рассматриваются все aci-инструкции, *запрещающие* выполнение авторизуемого действия. Если находится хотя бы одна запрещающая инструкция, то выполнение действия блокируется. Далее рассматриваются все *разрешающие* aci-инструкции. Если не находится ни одной инструкция, разрешающей это действие, то его выполнение также блокируется. Таким образом, запрещающие инструкции имеют безусловный приоритет по отношению к разрешающим инструкциям.

Заключение

В настоящей работе было дано подробное, систематизированное описание службы директорий LDAP, включая: структуру информации, представляемой в директории, принятый в LDAP способ организации и идентификации объектов, архитектуру и средства реализации распределенных директорий, основные операции, применимые к представленным в директории объектам, и способы защиты информации в директории от неавторизованного доступа.

Настоящая работа базируется на материалах ISO и IETF, стандартизующих модели директорий X.500 и LDAP, а также на документации по различным реализациям LDAP-серверов и LDAP-клиентов, разработанных Мичиганским университетом, компаниями Netscape, IBM и др. Проведенные исследования позволили достаточно подробно описать такие важные аспекты LDAP как механизмы организации и администрирования распределенной директории, которые в настоящий момент не стандартизованы и поэтому часто опускаются в литературе.

Следующая работа [ВКС2000] будет посвящена использованию LDAP в качестве инструментального средства для создания информационной службы метакомпьютера (ИСМ). Этот подход впервые был предложен авторами проекта Globus [FFK97]. В работе будет подробно рассмотрена схема LDAP-директории для метакомпьютерной системы, введены классы объектов, необходимых для представления вычислительных и сетевых ресурсов. Информация о состоянии таких ресурсов (метаданные) играет важную роль при организации высокопроизводительных вычислений в глобальной вычислительной среде. Кроме того, в работе будет дано достаточно полное представление об организации процесса сбора и актуализации метаданных, а также рассмотрены вопросы использования ИСМ с целью обеспечения

различных схем оптимального запуска приложений и планирования вычислений в метакомпьютерной среде.

Список литературы

[FFK97] S.Fitzgerald, I.Foster, C.Kesselman, G.von Laszewski, W.Smith, and S.Tuecke, “A Directory Service for Configuring High-Performance Distributed Computations”, In Proc 6th IEEE Symp. on High Performance Distributed Computing, pp.365-375, IEEE Computer Society Press, 1997

[JBH98] H.Johner, L.Brown, F.-S. Hinner, W.Reis, and J.Westman, “Understanding LDAP”, ITSO Redbook SG24-4986-00, June 1998

[JLM99] H.V.Jagadish, L.V.S.Lakshmanan, T.Milo, D.Srivastava, and D.Vista, “Querying Network Directories”, In Proc. ACM SIGMOD Int.Conf. on Management of Data, June 1999

[RFC1609] G.Mansfield, T.Johannsen, and M.Knopper, “Charting networks in the X.500 directory”, RFC1609, March 1994, (Experimental)

[RFC1823] T.Howes, and M.Smith, "The LDAP Application Program Interface", RFC1823, August 1995

[RFC2251] M.Wahl, T.Howes, and S.Kille, “Lightweight Directory Access Protocol (v3)”, RFC2251, December 1997

[RFC2252] M.Wahl, A. Coulbeck , T.Howes, and S.Kille, “Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions”, RFC2252, December 1997

[RFC2253] M.Wahl, T.Howes, and S.Kille, “Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names”, RFC2253, December 1997

[RFC2254] T.Howes, “The String Representation of LDAP Search Filters”, RFC2254, December 1997

[RFC2255] T.Howes, and M.Smith, “The LDAP URL Format”, RFC2255, December 1997

[RFC2256] M.Wahl, “A Summary of the X.500(96) User Schema for use with LDAPv3”, RFC2256, December 1997

[Smi96] M.Smith, “Definition of the inetOrgPerson Object Class”, Internet Draft (work in progress), November 1996

[ВКС2000] М.К.Валиев, Е.Л.Китаев, М.И.Слепенков, “Использование службы директорий LDAP для представления метаинформации в глобальных вычислительных системах”, Препринт ИПМ, 2000