

МОБИЛЬНАЯ
РЕЛЯЦИОННАЯ
СУБД **ЛИНТЕР**[®]

Linter Standard
Linter Bastion
Linter RealTime
Linter Multiversion

**Удалённое управление
компонентами СУБД**

НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ

 **РЕЛАКС**[®]

Товарные знаки

РЕЛЭКСТМ, ЛИНТЕР® , НЕВОД® , LAV™, ЛАКУНА являются товарными знаками, принадлежащими ЗАО НПП «Реляционные экспертные системы» (далее по тексту – компания РЕЛЭКС). Прочие названия и обозначения продуктов являются товарными знаками их производителей, продавцов или разработчиков.

Интеллектуальная собственность

Правообладателем продуктов ЛИНТЕР®, НЕВОД®, LAV™, ЛАКУНА является компания РЕЛЭКС (1990–2011). Все права защищены. Данный документ является собственностью компании РЕЛЭКС. Ни одна часть данного документа не может быть воспроизведена, передана, преобразована, сохранена в системе поиска информации, переведена на другой язык или компьютерный язык в какой-либо форме, какими-либо средствами, электронными, механическими, магнитными, оптическими, химическими, ручными или иными без предварительного разрешения компании РЕЛЭКС.

О документе

Материал, содержащийся в данном документе, прошел тщательную проверку, но компания РЕЛЭКС не гарантирует, что документ не содержит ошибок и пропусков. Компания РЕЛЭКС оставляет за собой право в любое время вносить в документ исправления и изменения, пересматривать и обновлять содержащуюся в нем информацию.

Адрес

394006, г. Воронеж, ул. 20-летия Октября, 119.
Тел./факс: (473) 2-711-711, 2-778-333.
e-mail: market@relex.ru.

Адрес для корреспонденции

394000, г. Воронеж, а/я 137.

Техническая поддержка

Отдел поддержки и сопровождения программных продуктов:

телефон: (473) 2-711-711 с 9:00 до 18:00 мск.
e-mail: support@relex.ru, market@relex.ru.

С целью повышения качества разрабатываемых программных средств и предоставляемых услуг в компании РЕЛЭКС действует автоматизированная система учёта и обработки рекламаций. Обо всех обнаруженных недостатках и ошибках в программном продукте и/или документации на него просим сообщать нам на Internet–странице [рекламация](#).

Оглавление

Предисловие	1
Назначение документа.....	1
Для кого предназначен документ.....	1
Необходимые предварительные знания.....	1
Принятые обозначения и соглашения.....	1
Общие сведения	3
Необходимые условия	4
Структура системы	4
Настройка системы	7
Конфигурационный файл.....	9
Переменные информационной секции.....	9
SysDescr.....	9
SysLocation.....	9
SysName.....	9
SysContact.....	9
Переменные секции управления агентом.....	10
listen.....	10
snmpEngineID.....	10
usmUser.....	11
group.....	12
view.....	13
access.....	13
Переменные секции управления компонентами.....	14
LinService.....	15
LinServiceParam.....	15
LinServiceEnv.....	16
Проверка системы	17
Управление системой	18
Агент удалённого управления.....	18
Утилиты администрирования.....	19
Общие ключи утилит.....	20
Общие ключи для всех утилит, работающих по протоколу SNMPv1 и SNMPv2.....	20
Общие ключи для всех утилит, работающих по протоколу SNMPv3.....	20
Утилита snmpget.....	21
Утилита snmpgetnext.....	22
Утилита snmpset.....	22
Утилита linsnmp.....	23
Утилита snmpwalk.....	24
Утилита snmptrap.....	24
Утилита snmptrapd.....	26
Утилиты snmpusm.....	26
Утилита snmpvacm.....	27
Сообщения системы	28

Приложение. Пример конфигурационного файла..... 30

Предисловие

Назначение документа

Документ содержит описание системы удалённого управления компонентами СУБД ЛИНТЕР, реализованной на базе протокола SNMP.

Приведены сведения о структуре системы, её конфигурировании и управлении с помощью набора утилит.

Документ может использоваться для работы с любой версией СУБД ЛИНТЕР. Особенности конкретных версий оговариваются по тексту.

Для кого предназначен документ

Документ предназначен для администраторов СУБД ЛИНТЕР и разработчиков клиентских ЛИНТЕР–приложений.

Необходимые предварительные знания



Для работы системой удалённого управления компонентами СУБД ЛИНТЕР необходимо:

- знать основы функционирования локальных вычислительных сетей (ЛВС);
- знать программные средства, поддерживающие функционирование протокола SNMP;
- уметь работать в соответствующей операционной системе на уровне продвинутого пользователя (устанавливать переменные окружения, контролировать состояние процессов ОС и т. п.).

Принятые обозначения и соглашения

<u>Обозначение</u>	<u>Пример</u>	<u>Значение</u>
Курсив	<i>Растровым</i> называется изображение...	Новый термин в тексте
Полужирный шрифт	В этом случае необходимо переносить все физические файлы.	Выделение в тексте
Подчеркнутый шрифт	Подробную информацию о работе программы можно получить на сайте www.dmk.ru .	Адреса страниц Internet
Текст, разделенный знаком ⇒	Выполните команду View ⇒ Properties (Вид ⇒ Свойства).	Последовательность выполнения команд
Текст, заключенный в <>, со знаком + между ними	<Ctrl>+<C>	В <> заключаются клавиши клавиатуры, знак + означает сочетание клавиш
Крупный моноширинный текст	SQL> _q	Текст командной строки

Предисловие

<u>Обозначение</u>	<u>Пример</u>	<u>Значение</u>
Мелкий моноширинный текст	Page Time Count	Текст программы
Заглавные буквы	BROWSE	Названия команд, слова, зарезервированные в SQL, ключевые слова
Курсив в < >	<return statement>	Определяемый элемент синтаксической конструкции
Символ ::=		Равенство по определению. Слева от знака стоит определяемое понятие, справа – собственно определение понятия
Квадратные скобки []	DBSTORE [-d -n -o -p -r -t -u]	Необязательные элементы конструкции. В данном примере ключи не являются обязательными элементами команды
Вертикальная черта	<return value> ::= <value expression> NULL	Указывает на то, что все предшествующие ей элементы списка являются необязательными и могут быть заменены любым другим элементом списка после этой черты
Фигурные скобки { }	CODEPAGE {866 1251 KOI8}	Указывают на то, что все находящееся внутри них является единым целым
Многоточие «...»	Характеристики столбца MAKE CHAR(20) MODEL CHAR(20) ... SQL>	Означает, что предшествующая часть может быть повторена любое количество раз
Многоточие, внутри которого находится запятая «,...»		Указывает, что предшествующая часть оператора, состоящая из нескольких элементов, разделенных запятыми, может иметь произвольное число повторений
Текст со знаком  на сером фоне	 Если конфигурация страницы-шаблона не учитывала свойств, команда будет выполнена некорректно.	Примечание

Общие сведения

Для успешного администрирования удалёнными компонентами СУБД ЛИНТЕР (сервисами) в сетевой среде необходимо знать их текущее состояние и иметь возможность изменять параметры их функционирования.

Удалённое управление компонентами СУБД ЛИНТЕР реализовано на основе протокола SNMP (Simple Network Management Protocol – простой протокол управления сетью). Основной концепцией этого протокола является то, что вся необходимая для управления компонентом информация хранится у агента, в так называемой Базе данных управляющей информации (MIB - Management Information Base). MIB представляет собой набор переменных, характеризующих состояние контролируемого компонента.

Помимо стандартных переменных, поддерживаемых протоколом SNMP, в MIB можно включать дополнительные параметры, специфичные для данного компонента. Однако при этом не нарушается принцип представления и доступа к административной информации – все они будут переменными в MIB. Поэтому SNMP, как непосредственно сетевой протокол, предоставляет только набор команд для работы с переменными MIB (Таблица 1).

Таблица 1. Операции, выполняющиеся с использованием SNMP

Операция	Выполняемая функция
GetRequest	Запрос значения одной или более переменных из MIB
GetNextRequest	Последовательное чтение значений параметров. Обычно используется для чтения значений из таблиц. После запроса первой строки при помощи get-request get-next-request используется для чтения оставшихся строк таблицы
SetRequest	Установка значения одной или более переменных в MIB
GetResponse	Возвращает ответ на запрос get-request, get-next-request или set-request. Ответ на запрос get-request или get-next-request будет содержать запрошенные данные, ответ на запрос set-request – только уведомление о результате выполнения операции
Trap	Уведомительное сообщение о событии типа «ловушка». Содержит внутри себя специальный OID (идентификатор объекта), информацию о том, какой MIB– объект установил ловушку и данные этого объекта

Для того, чтобы проконтролировать работу компоненты, необходимо просто получить доступ к его MIB, и проанализировать значения некоторых переменных. Задача обеспечения выполнения команд состоит, таким образом, в регистрации специальных переменных MIB и реакции компонента на их изменения.

SNMP работает на основе протокола UDP и для общения с сетью использует порт с номером 161 (для отправки уведомлений – порт с номером 162). Использование UDP в качестве основы означает, что данные передаются без установления соединения. Это дает возможность существенно уменьшить требования к сетевой инфраструктуре и накладные расходы на передачу данных. Пакеты SNMP могут передаваться также поверх протоколов ATM, Ethernet, IPX. Утилиты СУБД ЛИНТЕР для своей работы с SNMP используют 1161 порт.

В SNMP могут одновременно поддерживаться версии протокола SNMPv1, SNMPv2c и SNMPv3. Это означает, что возможно использование как старого механизма авторизации с помощью community строк, так и нового с помощью пользователей и паролей (RFC3414 - User-based Security Model (USM)). В случае использования USM передача сообщений возможна с разными уровнями безопасности:

- noAuthNoPriv – без подписи и сокрытия тела сообщения;
- authNoPriv – подпись сообщения без шифрации;
- authPriv – подпись сообщения и шифрация тела сообщений.

Необходимые условия

Для работы с системой удалённого управления компонентами СУБД ЛИНТЕР необходима поддержка операционной системой сетевых протоколов TCP/IP И UDP.

Структура системы

В системе удалённого управления компонентами СУБД ЛИНТЕР реализована универсальная модель управления с возможностью расширения в дальнейшем её основных функциональных возможностей.

Функционально система удалённого управления включают агента системы и набор утилит администрирования.

Состав утилит удалённого управления:

- 1) `snmpget` – получение элемента из дерева MIB;
- 2) `snmpgetnext` – получение следующего элемента из дерева MIB;
- 3) `snmpset` – установка значения элемента дерева MIB;
- 4) `linsnmp` – доступ к LINTER-MIB.

Для обработки запросов управляющей станции, приходящих в виде SNMP пакетов, служит специальный модуль, называемый агентом, который реализован в виде демона. Агент принимает SNMP пакеты и выполняет соответствующие им действия, т.е. посылает значение запрашиваемой переменной, устанавливает значение переменных, выполняет периодическое обновление информации MIB, выполняет в ответ на установку соответствующих переменных некоторые операции.

Работа агента управляется менеджером – программой, работающей на управляющей станции. Функции менеджера удалённого управления компонентами СУБД выполняет набор утилит администрирования. Агент выступает посредником между внутренними структурами управляемого компонента и менеджером.

Обычно взаимодействие происходит по инициативе менеджера и выглядит следующим образом:

- менеджер отправляет запрос агенту;
- агент обрабатывает запрос, собирает требуемые данные и отправляет их назад менеджеру;

- менеджер получает запрошенные данные и обрабатывает их в соответствии с предусмотренным алгоритмом.

В некоторых случаях агент может самостоятельно инициировать обмен данными. Обычно у агента должен быть список важных событий, о наступлении которых он обязан оповестить менеджера. Менеджер по своему усмотрению выполняет какие-либо действия в ответ на оповещение. Например, такими событиями могут быть аварийное завершение работы наблюдаемого компонента, аварийная перезагрузка, вызванная потерей питания или любая другая критическая ситуация. Процедура оповещения в терминах протокола SNMP называется отправкой ловушки (SNMP Trap). В сообщении уведомления агент посылает данные, специфичные для данного события, если они есть в MIB (в описании соответствующего события NOTIFICATION-TYPE присутствует секция OBJECTS). Уведомление посылается сразу нескольким получателям, которые определены во внутренних таблицах с помощью SNMP-NOTIFICATION-MIB.



Получатели определены в таблице snmpTargetAddrTable.

Обычно MIB принято представлять в виде древовидной структуры. Определенные части этого дерева являются обязательными для всех реализаций SNMP.

В тоже время производитель программного обеспечения может встраивать внутрь этого дерева свои собственные поддеревья. Чаще всего они находятся в ветке iso.org.dot.internet.private. С помощью такого встраивания реализуется возможность получить доступ к функциям и данным, характерным только для этого компонента. Для примера посмотрим на стандартный образец данных, используемых агентом SNMP, в системе распределенного управления компонентами СУБД ЛИНТЕР (Рис. 1).

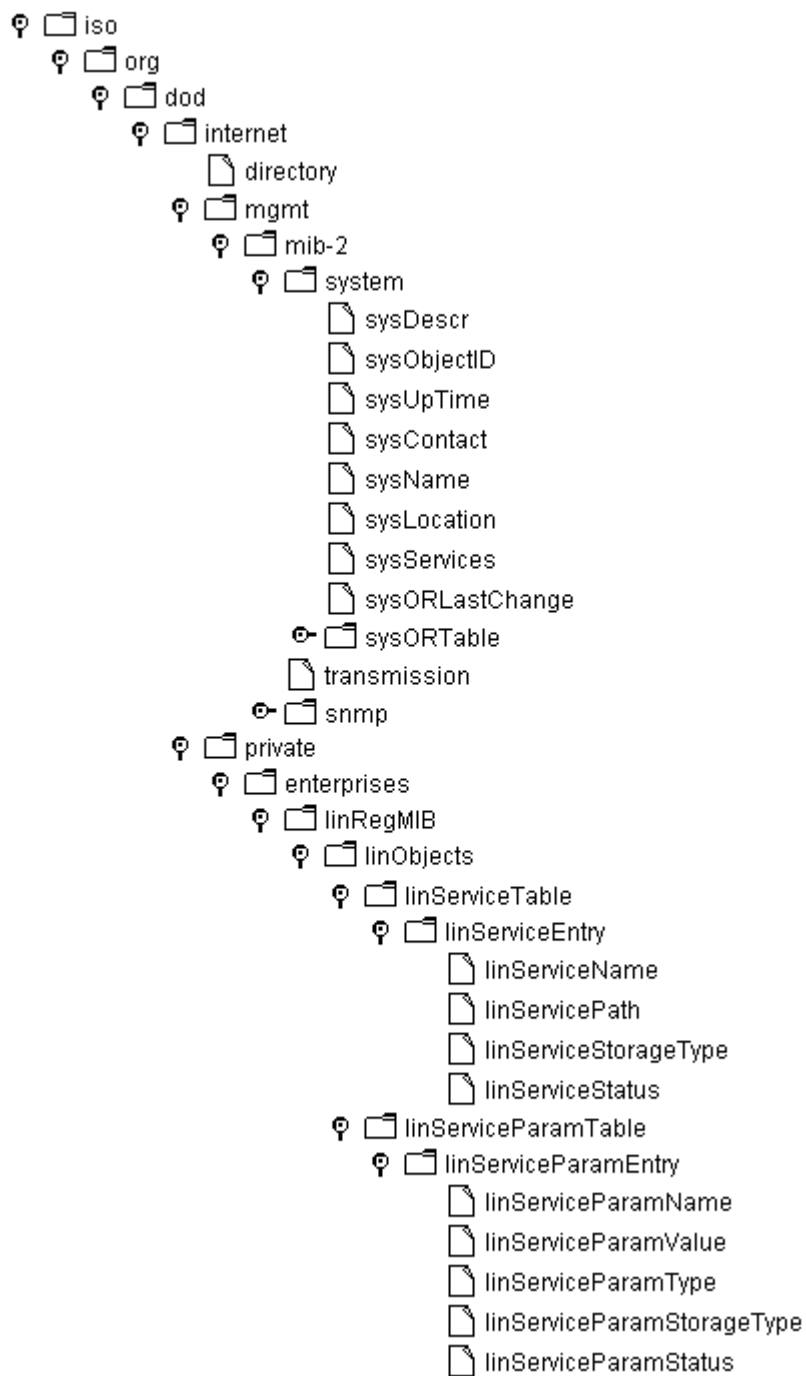


Рис. 1. Пример MBI дерева распределенного управления компонентами СУБД ЛИНТЕР

Например, для того чтобы узнать текущее состояние сервиса kernel, нужно пройти на ветку `linServiceStatus.kernel`. Буквенная запись используется для удобства человека, программные средства (менеджеры и агенты) оперируют числами. Соответственно, каждая часть пути имеет свой числовой идентификатор (Таблица 2).

**Таблица 2. Соответствие символьных обозначений
и цифровых идентификаторов дерева MIB**

Символьное обозначение	Цифровой идентификатор
Iso	1
Org	3
Dod	6
Internet	1
Mgmt	2
mib2	1
System	1
sysUpTime	3

Таким образом, местоположение интересующего объекта задается с помощью уникальной комбинации чисел, однозначно идентифицирующей объект, находящийся в любой ветви дерева. Эта комбинация чисел называется OID (Object identifier – «Идентификатор объекта»).

В связи с тем, что префикс iso.org.dod.internet.mgmt.mib2 встречается почти в каждом OID, в большинстве случаев его можно не прописывать. Подразумевается наличие этого префикса, поэтому очень часто можно использовать сокращенную запись пути в вид system.sysUpTime.

В случае, если экземпляр объекта один, ему присваивается номер "0". Доступ к его данным можно получить, обратившись к нему как system.sysUpTime.0.

Настройка системы

Настройка функционирования распределенной системы управления СУБД ЛИНТЕР выполняется с помощью конфигурационного файла, который представляет собой обычный текстовый файл и подготавливается вручную с помощью любого текстового редактора.

Конфигурационный файл является входным параметром при запуске агента.

Логически конфигурационный файл включает 3 секции:

- 1) информационная секция. Содержит справочную информацию о самом агенте;
- 2) секция управления агентом. Содержит параметры настройки функционирования собственно агента;
- 3) секция (таблица) управления компонентами. Содержит информацию о всех компонентах СУБД ЛИНТЕР и их параметрах, контролируемых данным агентом.

Одновременно на компьютере могут быть запущено любое число агентов с разными конфигурационными файлами. При старте агента он запускает описанные в конфигурационном файле компоненты в соответствии с заданными параметрами

настройки. Таким образом, с помощью нескольких агентов можно запустить несколько экземпляров одного и того же компонента, как с одинаковыми, так и разными параметрами запуска.

Общие правила формирования конфигурационного файла:

- каждая переменная должна занимать одну строку (длиной до 255 знаков), продолжение строк не разрешается;
- имена переменных регистронезависимы;
- если некоторая переменная в файле не описана, то будет использоваться ее значение по умолчанию;
- если некоторая переменная описана несколько раз, будет использоваться ее последнее значение (за исключением переменных, задающих строки таблицы);
- порядок расположения переменных в файле в общем случае произвольный, за исключением ситуаций, когда значение одной переменной зависит от другой;
- признаком комментария является знак #, который должен быть первым символом строки файла (возможно, после нескольких символов пробела и/или знака табуляции). Комментарий не должен помещаться внутри описания переменной;

Конфигурационный файл

Переменные информационной секции

SysDescr

Назначение

Описание агента.

Синтаксис

```
SysDescr <строка>
```

Пример

```
SysDescr SNMP-агент ЛИНТЕР-сервера DEMO
```

SysLocation

Назначение

Описание местоположения агента в сети.

Синтаксис

```
SysLocation <строка>
```

Примеры

- 1) SysLocation Узел Sale
- 2) SysLocation Router

SysName

Назначение

Задаёт краткое имя агента.

Синтаксис

```
SysName <строка>
```

Примеры

- 1) SysName А_Склад
- 2) SysName "А Склад"

(имя задано в кавычках, т. к. в написании имени встречается пробел)

SysContact

Назначение

Задаёт контактную информацию для связи с разработчиками агента (номер телефона, e-mail адрес и т. п.).

Конфигурационный файл

Синтаксис

`SysContact <строка>`

Пример

```
SysContact      me@somewhere.org
```

Переменные секции управления агентом

listen

Назначение

Устанавливает сетевой адрес (в формате протокола TCP/IP) и номер порта, по которым агент будет слушать входящие запросы.

Синтаксис

`Listen <адрес / [порт]> | <[адрес] / [порт]>`

Один из аргументов `<адрес>` или `<порт>` должен быть обязательно задан. Значение по умолчанию – 127.0.0.1/1161.

Примеры

- 1) `Listen 192.168.5.8/1161`
- 2) `Listen 192.168.5.8` по умолчанию используется `192.168.5.8/1161`
- 3) `Listen /162` по умолчанию используется `127.0.0.1/162`

snmpEngineID

Назначение


Задаёт уникальный (в пределах сети) идентификатор длиной от 5 до 32 шестнадцатеричных цифр SNMP-агента, который будет стартовать с данным конфигурационным файлом.

Синтаксис

`snmpEngineID <шестнадцатеричная строка>`

`<Шестнадцатеричная строка>` не может быть пустой (т. е. строкой нулевой длины) или состоять из всех нулей или значений 'ff'.

`<Шестнадцатеричная строка>` может быть конфигурировано с помощью приведенного ниже алгоритма.

 В случаях, когда используются несколько экземпляров агента в одной и той же самой системе, использование этого алгоритма НЕ допускается, поскольку это приведет к тому, что все экземпляры агента будут получать одно и то же значение идентификатора.

Алгоритм выбора идентификатора агента:

- 1) первый бит используется, чтобы указать, как сформирована остальная часть данных:
 - 0 - предприятием (см. пункт 2 ниже);

- 1 - одним из стандартных способов (см. пункт 3 ниже).

2) snmpEngineID имеет длину 12 байт.

Первые четыре байта представляют двоичный эквивалент личного номера организации. Номер присваивается IANA (Internet Assigned Numbers Authority – организацией, отвечающей за присвоение номеров. Например, если бы РЕЛЕКС был назначен номер предприятия 696, то первые четыре байта должны выглядеть как '000002b8'H.

Оставшиеся восемь байтов задаются с помощью одного из представленных организацией–разработчиком методов. Эти методы должны гарантировать уникальность значения snmpEngineID в административном домене агента.

Например, это может быть IP-адрес SNMP-объекта, или MAC-адрес одного из интерфейсов с добавлением к каждому адресу случайных байтов. Если предлагаются для применения несколько методов, то рекомендуется, чтобы первый байт указывал на используемый метод и оставшиеся байты были функцией этого метода.

3) длина байтовой строки переменная.

Первые четыре байта представляют двоичный эквивалент личного номера организации–разработчика (см. пункт 2 алгоритма). Первый бит в этом случае установлен в 1. Например, вышеупомянутое значение для НИИАА теперь должно быть представлено как '800002b8'H.

Пятый байт указывает, как сформированы остальные байты (6-ой и последующие).

Допустимые значения пятого байта:

- 0 – зарезервировано, не используется;
- 1 – IPv4-адрес (4 байта Ipv4- адреса);
- 2 – IPv6-адрес (16 байтов Ipv6- адреса);
- 3 – MAC-адрес (6 байтов IEEE MAC-адреса);
- 4 – административно задаваемая символьная строка. Максимальная длина – 27 символов;
- 5 – административно задаваемая байтовая строка. Максимальная длина – 27 байтов;
- 6-127 – зарезервировано, не используется;
- 28-255 – определяется разработчиком. Максимальная длина – 27 байтов.

По умолчанию – генерируется автоматически (используется имя компьютера) при каждом запуске агента, т. к. сохраняется только на время сеанса работы агента.

Пример

```
EngineID 8F270000046167656E7431
```


usmUser

Назначение

Задаёт создание пользователя данного агента. Имя пользователя должно быть уникальным в пределах сети. При указании протокола аутентификации и шифрации и

Конфигурационный файл

соответствующих паролей для пользователя будут сгенерированы локализованные ключи для безопасной передачи сообщений с помощью указанных протоколов.

 В данной версии системы удалённого управления компонентами СУБД ЛИНТЕР не поддерживается.

Синтаксис

```
usmUser <имя пользователя>  
[<параметры аутентификации>  
[<параметры шифрации>] ]
```

<параметры аутентификации> ::= <протокол аутентификации>
<пароль>

<параметры шифрации> ::= <протокол шифрации> <пароль>

<протокол аутентификации> ::= **MD5** | **SNA**

<протокол шифрации> ::= **DES**

<имя пользователя> ::= символьная строка длиной не более 32 знака

<пароль> ::= не пустая символьная строка

Примеры

- 1) usmUser "А. В. Петров"
- 2) usmUser sys_admin
- 3) usmUser user1 MD5 Fc76\$gv
- 4) usmUser user2 MD5 Fc76\$gv DES acφrst
- 5) usmUser templateMD5 MD5 "" DES ""
- 6) usmUser templateSHA SHA "" DES ""

group

Назначение

Включает указанного пользователя в группу пользователей с соответствующей моделью безопасности. Пользователь агента должен быть предварительно создан (см. описание переменной usmUser). Если при запуске агента требуемая группа не существует, то она будет создана.

Синтаксис

```
group <имя группы> <модель безопасности> <имя  
пользователя>]
```

<модель безопасности> ::= **v1** | **v2c** | **usm**

<имя пользователя> ::= символьная строка длиной не более 32 знака

Примеры

- 1) group MyRWGroup v1 local
- 2) group MyRWGroup usm local
- 3) group initial usm "А. В. Петров"

```
4) group "Сов. секретно" v2c SYS_ADMIN
```

view

Назначение

В реальных системах дерево MIB может содержать достаточно много элементов, администрировать которые может быть затруднительно. Для облегчения этой работы используется механизм представлений. Представление – это некоторое именованное подмножество элементов MIB (поддерево), рассматриваемое как единое целое. Установки администрирования, примененные к представлению, распространяется сразу на все элементы этого представления.

Синтаксис

```
view <имя представления> <тип операции> <поддерево>  
<тип операции> ::= included | excluded
```

<имя представления > ::= символьная строка длиной не более 32 знака

<поддерево> ::= идентификатор, задающий иерархическое местоположение добавляемого (исключаемого) элемента дерева MIB.

Переменная view описывает включение (опция included) или исключение (опция excluded) указанного элемента MIB-дерева в данное представление.

Идентификатор должен задавать иерархический адрес элемента дерева в виде цепочки уровней иерархии: <n>.<m>.<p>.<s>.... В представление можно включать (исключать) сразу все дочерние элементы некоторого уровня, в этом случае вся группа элементов обозначается знаком "*": <n>.*.*.<s>...

Примеры

- 1) view all included 1
- 2) view internet included 1.3.6.1
- 3) view restricted excluded 1.3.*.*

access

Назначение

Устанавливает модель и уровень безопасности при работе с элементами дерева MIB при работе с ними в различных режимах.


Синтаксис

```
access <имя группы>  
    <контекст> <применение контекста>  
    <модель безопасности>  
    <уровень безопасности>  
    <представление для чтения>  
    <представление для записи>  
    <представление для уведомлений>
```

<имя группы> ::= символьная строка длиной не более 32 знака.

<контекст> ::= ""

Конфигурационный файл

 В данной реализации должна быть пустая строка (используется контекст по умолчанию)

<применение контекста> ::= **exact** | **prefix**

<модель безопасности> ::= **v1** | **v2c** | **usm**

<уровень безопасности> ::=

1) для модели безопасности v1 и v2c:

- noAuthNoPriv – без подписи и сокрытия тела сообщения.

2) для модели безопасности usm:

- noAuthNoPriv – без подписи и сокрытия тела сообщения;
- authNoPriv – подпись сообщения без шифрации;
- authPriv – подпись сообщения и шифрация тела сообщений.

<представление для чтения> ::= символьная строка длиной не более 32 знака.

<представление для записи> ::= символьная строка длиной не более 32 знака.

<представление для уведомлений> ::= символьная строка длиной не более 32 знака.

Указание несуществующего представления (любого из <представление для чтения>, <представление для записи>, <представление для уведомлений>) запрещает соответствующее действие.

Если <применение контекста > равно exact, то поле <контекст> в точности должно соответствовать контексту, которому назначаются права доступа, если prefix – право доступа можно назначать нескольким контекстам с этим префиксом.

Примеры

```
1) access MyROGroup "" v1 noAuthNoPriv exact all none none
2) access MyRWGroup "" v1 noAuthNoPriv exact all all none
3) access initial "" v2c noAuthNoPriv exact restricted
   restricted restricted
4) access initial "" usm authNoPriv exact internet
   internet internet
```

Переменные секции управления компонентами

Секция распределенного управления компонентами содержит необходимую информацию для базы данных управляющей информации управления (LINTER-MIB). LINTER-MIB содержит две таблицы:

1) первая таблица (linServiceTable) содержит сервисы, которыми можно удаленно управлять. Сервис определяется своим уникальным именем (linServiceName) имеющим не более 32 символов и удобным для восприятия администратора (например, kernel59, jdbc и т.д.). Имя сервиса связано с определенным исполняемым файлом (linServicePath). Несколько сервисов могут иметь один и тот же исполняемый файл. Состояние каждого сервиса отображается в поле linServiceStatus. Сервис может быть в активном состоянии (active), при этом он запущен и работает, и в неактивном состоянии (notInService).

Изменение состояние (запуск и останов) производится изменением соответствующего значения поля `linServiceStatus`;

2) вторая таблица (`linServiceParamTable`) содержит параметры сервисов. Параметры имеют свое реальное имя (`linServiceParamName`), тип (`linServiceParamType`) и значение (`linServiceParamValue`). Допустимыми типами являются значение 1, определяющее переменную среды окружения (например, `LINTER_MBX`), и значение 2, определяющее параметр командной строки (например, `pool`).

Перечисленные таблицы в системе задействованы, из конфигурационного файла их создавать нельзя, т.е. секции `LinService`, `LinServiceParam`, `LinServiceEnv` СУБД ЛИНТЕР не поддерживаются.

Для нормальной работы сервис не должен становиться демоном после создания своего процесса, иначе система мониторинга не сможет контролировать работу этого процесса. Чтобы запустить сервис в обычном режиме, в большинстве случаев можно указать соответствующий параметр.

LinService

Назначение

Задаёт создание сервиса для системы удалённого управления компонентами СУБД ЛИНТЕР.

Синтаксис

```
LinService <имя сервиса> <исполняемый файл>
```

<имя сервиса> ::= символьная строка длиной не более 32 знака.

<исполняемый файл> ::= символьная строка длиной не более 255 знаков, задающая местоположение и имя исполняемого файла данного сервиса

Примеры

- 1) `LinService kernel /home/fbv/linter/bin/linter`
- 2) `LinService linapid /home/fbv/linter/bin/linapid`
- 3) `LinService dbs_tcp /home/fbv/linter/bin/dbs_tcp`

LinServiceParam

Назначение

Добавление параметра командной строки для запуска сервиса.

Синтаксис

```
LinServiceParam <имя сервиса> <имя параметра><значение параметра>
```

<имя сервиса> ::= символьная строка длиной не более 32 знака.

<имя параметра> ::= символьная строка длиной не более 32 знаков.

<значение параметра> ::= литерал, задающий значение параметра (тип данных литерала должен приводиться к типу данных параметра).

Примеры

- 1) `LinServiceParam kernel pool 1000`
- 2) `LinServiceParam kernel debug ""`
- 3) `LinServiceParam linapid port 1090`

LinServiceEnv

Назначение

Добавление переменной среды окружения для сервиса.

Синтаксис

```
LinServiceEnv <имя сервиса>  
                <имя переменной окружения>  
                <значение переменной окружения>
```

<имя сервиса> ::= символьная строка длиной не более 32 знака.

<имя переменной окружения > ::= символьная строка длиной не более 32 знаков.

<значение переменной окружения > ::= литерал, задающий значение параметра (тип данных литерала должен приводиться к типу данных параметра).

Проверка системы

Для проверки активности агента необходимо:

1. Проверить наличие процесса `linsnmpd` в списке процессов с помощью команды `ps`.

Пример:

```
$ ps -ax | grep linsnmpd
20403  ??  Ss      0:00.00  ./linsnmpd -c snmp.conf
```

2. Удаленно или локально попробовать получить некоторые данные из части дерева MIB, например, значение `sysDescr.0` (1.3.6.1.2.1.1.1.0), содержащее описание удаленного агента.

Пример:

```
$ snmpget -v 1 -c public 127.0.0.1/161 1.3.6.1.2.1.1.1.0
1.3.6.1.2.1.1.1.0 = (OCTET STRING) SNMP agent.
```

Управление системой

Агент удалённого управления

Командная строка запуска

Запуск агента осуществляется стандартными средствами запуска задач операционной системы.

Исполняемый файл агента `linsnmpd`.

```
linsnmpd [-c <config-файл>]
          [-f]
          [-l <log-файл>]
          [-p <pid>]
```

Запуск агента должен выполнять суперпользователь операционной системы, чтобы иметь возможность использовать стандартные порты SNMP 161 и 162.

Ключи агента

`-c [<config-файл>]`

Задаёт путь к конфигурационному файлу агента. Если не задан ключ и/или параметр `<config-файл>`, по умолчанию используется файл `snmp.conf`, который ищется в каталоге запуска агента

`-f`

Запрещает агенту переходить в режим демона. По умолчанию переход в этот режим разрешен,

`-l [<log-файл>]`

Задаёт путь к файлу протоколирования работы агента. Если ключ не задан, файл протоколирования по умолчанию не создается.

`-p <pid>`

Задаёт путь к `pid`-файлу.

Функции агента

Агент выполняет следующие функции:

- анализирует переданный конфигурационный файл и создает на его основе в памяти структуру данных, описывающую дерево MIB распределенного управления для данного агента;
- запускает по команде все сервисы, описанные в дереве MIB, в соответствии с установленными в MIB параметрами;
- если задан ключ `-l`, инициирует работу с файлом протоколирования, в который пишутся текстовые сообщения типа:

```

Registered transport domain 1.3.6.1.6.1.1
Initializing message queue
AuthPriv initialization
Added auth protocol HMAC-SHA with id 3.
Added auth protocol HMAC-MD5 with id 2.
Added privacy protocol DES with id 2.
USM initialization
Registered subtree: 1.3.6.1.2.1.1.1
Registered subtree: 1.3.6.1.2.1.1.2
Registered subtree: 1.3.6.1.2.1.1.3
...
Registered subtree: 1.3.6.1.4.1.9999.1.8.1
Registered subtree: 1.3.6.1.4.1.9999.1.9.1
snmpEngineID (11 bytes):
00000000 8F 27 00 00 04 61 67 65 6E 74 31
.'...agent1
engineBoots 95
udpOpen: transport created
udpListen: start listen on 192.168.5.8:1161
...
sysUpTime 0
Start processing messages


```

Если сообщению предшествует слово **ERROR**, то это говорит о том, что произошла ошибка, например:

```

ERROR: passwordToKeyAuth: no password given
ERROR: usmAddUser: unable to build authentication key

```

 Все изменения дерева MIB, выполненные агентом по заданию утилит менеджера, сохраняются только на время сеанса работы агента и в конфигурационный файл не вносятся.

Примеры запуска агента

```

linsnmpd
linsnmpd -l "d:\Program Files\linter\bin\log\snmp.txt"
linsnmpd -c snmp.conf -l snmp.log

```

Утилиты администрирования

Средства администрирования распределенного управления представлены набором утилит.

Все утилиты имеют однотипный формат командной строки:

```
<утилита> [{<ключ>... ] <адрес/порт> [<имя переменной >...]
```

Общие ключи утилит

-v1

Задаёт версию протокола SNMPv1. Используется по умолчанию.

-v2c

Задаёт версию протокола SNMPv2c.

-v3

Задаёт версию протокола SNMPv3.

-r <количество повторов>

Задаёт количество попыток отправки запроса. По умолчанию 3 попытки.

-t <тайм-аут>

Задаёт длительность ожидания ответа (в сек). По умолчанию 1.5 сек.

 Имена ключей утилит регистрозависимы.

Общие ключи для всех утилит, работающих по протоколу SNMPv1 и SNMPv2

-s <сообщество>

Задаёт сообщество пользователей агента. По умолчанию public.

Общие ключи для всех утилит, работающих по протоколу SNMPv3

-a <протокол аутентификации>

Задаёт протокол аутентификации (MD5|SHA). По умолчанию протокол аутентификации не используется.

-A <пароль аутентификации>

Задаёт пароль протокола аутентификации.

-e <ID-агента>

Задаёт идентификатор агента (например, 800000020109840301). Умолчания нет. Если идентификатор не указан, то он определяется автоматически.

-E <ID-контекста>

Задаёт идентификатор контекста агента, например, 800000020109840301). По умолчанию используется идентификатор агента.

-l <уровень безопасности>

Задаёт уровень безопасности (noAuthNoPriv | authNoPriv | authPriv). По умолчанию noAuthNoPriv.

-n <контекст >

Задаёт имя контекста, по умолчанию используется пустая строка.

-u <имя пользователя>

Задаёт имя пользователя агента.

-x <протокол шифрования>

Задаёт протокол шифрования (DES).

-X <пароль шифрования>

Задаёт пароль протокола шифрования.

-Z <количество>, <время>

Задаёт время запуска агента и количество запусков. Если ключ не задан, автоматически производится синхронизация задачи и агента.

Утилита snmpget

Назначение

Получение данных от удаленного SNMP агента.

Команда запуска

```
snmpget [<ключ>...]<адрес/порт> [<имя переменной>... ]
```

<Имя переменной> необходимо задавать:

- 1) в цифровом формате – полностью;
- 2) в текстовом виде – полностью или как часть полного имени.

Например, переменную sysDescr можно задавать следующим образом:

- 1.3.6.1.2.1.1.1.0;
- iso.org.dod.net.internet.mgmt.mib-2.system.sysDescr.0;
- как часть полного текстового имени: mgmt.mib-2.system.sysDescr.0.

Примеры

1) получить значение переменной `sysDescr.0`:

```
$ snmpget -v 1 -c public 192.168.5.8/161 1.3.6.1.2.1.1.1.0
1.3.6.1.2.1.1.1.0 = <ОКТЕТ STRING> My agent
```

2) получить значение переменной `sysName.0` в защищенном режиме:

```
$ snmpget -v 3 -e 800000020109840301 -u admin -l authPriv -a
MD5 -A password -x DES -X password 192.168.5.8/161
1.3.6.1.2.1.1.5.0
1.3.6.1.2.1.1.5.0 = <ОКТЕТ STRING> agent1
```

Утилита `snmpgetnext`

Назначение

Получение следующего значения (в лексикографическом порядке) элемента дерева MIB после заданной переменной.

Команда запуска

```
snmpgetnext [<ключ>...] <адрес/порт> [<имя переменной>... ]
```

<Имя переменной> – см. описание утилиты `snmpget`.

Примеры

```
1) $ snmpgetnext -v 1 -c public 192.168.5.8/161 1.3.6.1.2.1.1.1
1.3.6.1.2.1.1.1.0 = <ОКТЕТ STRING> My agent
```

```
2) $ snmpgetnext -v 1 -c public 192.168.5.8/161
1.3.6.1.2.1.1.1.0
1.3.6.1.2.1.1.2.0 = <OBJECT IDENTIFIER> 1.3.6.1.4.1.9999
```

Утилита `snmpset`

Назначение

Создание новых и/или изменение существующих данных в дереве MIB.

Команда запуска

```
snmpset [<ключ>... ] <адрес/порт>
[ {<имя переменной> <тип> <значение>}... ]
```

<Имя переменной> – см. описание утилиты `snmpget`.

<тип> – тип переменной:

- i – INTEGER;
- u – UNSIGNED;
- c – COUNTER32;
- s – STRING;

- x – HEX STRING;
- d – DECIMAL STRING;
- n – NULLOBJ;
- o – OBJID;
- t – TIMETICKS;
- a – IPADDRESS;
- b – BITS.

Пример

```
$ snmpset -v 1 -c private 192.168.5.8/161 1.3.6.1.2.1.1.4.0 s  
admin@relex.ru
```

Утилита linsnmp

Назначение

Управление списком сервисов на удалённом ЛИНТЕР-сервере и параметрами их запуска.

Команда запуска

```
linsnmp [ <ключ>... ] <адрес/порт> [<команда>...]
```

<команда> – одна из нижеследующих команд:

```
show Service
```

Вывод списка сервисов на удаленном ЛИНТЕР-сервере. Элемент списка содержит название сервиса, путь к исполняемому файлу и состояние сервиса.

```
service <сервис> <файл>
```

Создание сервиса с именем <сервис> на удаленном ЛИНТЕР-сервере с заданным исполняемым <файлом> или модификация пути исполняемого файла соответствующего сервиса.

```
no service <сервис>
```

Удаление <сервиса> на ЛИНТЕР-сервере.

```
start service <сервис>
```

Запуск <сервиса> на удаленном ЛИНТЕР-сервере.

```
stop service <сервис>
```

Останов <сервиса> на удаленном ЛИНТЕР-сервере

```
show param <сервис>
```

Управление системой

Предоставление списка параметров <сервиса> на удаленном ЛИНТЕР-сервере.

<сервис> <параметр> <значение>

Установка <значения> <параметру> для <сервиса> на удалённом ЛИНТЕР-сервере

no param <сервис> <параметр>

Удаление <параметра> из <сервиса> на удаленном ЛИНТЕР-сервере.

Примеры

```
1)$ linsnmp -v 3 -u admin -l authPriv -a MD5 -A password -x DES
-X password 192.168.5.8/1161 service kernel
/usr/local/linter/bin/linter
```

```
2)$ linsnmp -v 3 -u admin -l authPriv -a MD5 -A password -x DES
-X password 192.168.5.8/1161 start service kernel
```

```
3)$ linsnmp -v 3 -u admin -l authPriv -a MD5 -A password -x DES
-X password 192.168.5.8/1161 show service
kernel /usr/local/linter/bin/linter active
```

Утилита snmpwalk

Назначение

Получение значений всех потомков заданного узла дерева MIB.

Команда запуска

```
snmpset [<ключ>... ] <адрес/порт> [ {<имя узла>}... ]
```

Если <имя узла> не задано, то выводится все дерево MIB.

Примеры

```
1) вывести все дерево MIB
snmpwalk -c public localhost/1161
```

```
2) вывести на экран все переменные из поддерева
iso.org.dod.internet.private.enterprises.relex
```

```
snmpwalk -c public localhost/1161 relex
```

Утилита snmptrap

Назначение

Посылает уведомление об установленной ловушке по заданному адресу.

Команда запуска

Для SNMPv1:

```
snmptrap [<ключ>... ] <адрес/порт>
<получатель>
<адрес>
<тип>
<детализация>
<время>
[
{<идентификатор переменной>
<тип переменной>
<значение переменной>}
...
]
```

Для SNMPv2 и SNMPv3:

```
<время>
<источник уведомления>
[
{<идентификатор переменной>
<тип переменной>
<значение переменной>}
...
]
```

<получатель> – целевой компьютер или Интернет-адрес;

<адрес> – сетевое имя для доступа к целевому компьютеру;

<тип> - стандартный тип уведомления (1 – coldStart , 2 – wormStart, 3 – linkDown и др.);

<детализация> – пользовательская детализация стандартного типа уведомления (целочисленное значение);

<время> – длительность работы (в миллисекундах);

<источник уведомления> – идентификатор источника уведомления.

Пример

```
snmptrap -v 1 -c public host TRAP-TEST-MIB::demotraps localhost
6 17
```

```
' ' \SNMPv2-MIB::sysLocation.0 s "Just here"
```

Утилита snmptrapd

Назначение

Вывод на консоль информации о полученных уведомлениях.

Команда запуска

```
snmptrapd [-c <файл конфигурации>]
```

```
[-l <файл протоколирования>]
```

```
[-f]
```

-f – запускать в режиме обычного приложения (не демона).

Утилиты snmpusm

Назначение

Управление правами SNMPv3 пользователей.

Команда запуска

```
snmptrap [<ключ>... ] <адрес/порт> <команда>
```

<команда> – одна из нижеследующих команд:

```
show counters
```

Отображать счетчик ошибок USM (User-based Security Module – модуль безопасности пользователя).

```
show users
```

Отображать пользователей.

```
create user <имя>
```

Создание нового пользователя.

```
clone user <имя> <прототип>
```

Создание нового пользователя по образцу прототипа.

```
authpasswd <имя> <старый пароль> <новый пароль>
```

Изменение пароля аутентификации.

`privpasswd <имя> <старый пароль> <новый пароль>`

Изменение личного пароля.

Утилита `snmpv3m`

Назначение

Создание и поддержка записей управления доступом SNMPv3 пользователей.

Команда запуска

`snmpv3m [<ключ>...] <адрес/порт> <команда>`

<команда> – одна из нижеследующих команд:

`view <имя представления> <поддерево> [include]`

Создание представления с включением в него заданного поддерева.

`view <имя представления> <поддерево> [exclude]`

Создание представления с исключением из него заданного поддерева.

`no view <имя представления>`

Удаление представления.

`group <имя группы> <модель безопасности> <защищаемое имя>`

Создание записи в таблице групп для пользователя с <защищаемым именем>.

`no group <имя группы> <модель безопасности> <защищаемое имя>`

Удаление записи в таблице групп для пользователя с <защищаемым именем>.

Сообщения системы

При возникновении ошибочной ситуации утилиты выдают на экран текстовое сообщение об ошибке. Если ошибка относится к определенному запрашиваемому или устанавливаемому элементу, то дополнительно выводится идентификатор этого элемента.

Список сообщений об ошибках и рекомендуемые способы их устранения приведены в таблице 3.

Таблица 3. Сообщения утилит удалённого управления компонентами

Сообщение	Причина	Способ устранения
Timeout. No Response from ...	В течение заданного промежутка времени не получен ответ агента на запрос	1. Увеличить интервал ожидания; 2. Проверить активность агента 3. Проверить что заданный уровень безопасности совпадает с уровнем безопасности указанного пользователя.
To big	Превышен максимальный размер сообщения	Уменьшить количество запрашиваемых элементов
No such name	Нет такого элемента	Правильно указать идентификатор элемента
Bad value	Неправильное значение при установке значения элемента	Изменить устанавливаемое значение
Read only	Элемент дерева не предполагает изменение	Проверить правильность задания идентификатора элемента
Generic error	Произошла ошибка, не поддающаяся классификации	1. Повторить попытку. 2. Обратиться к разработчику
No access	Отказано в доступе	1. Проверить правильность задания идентификатора элемента; 2. Проверить правильность настройки прав агента
No creation allowed	Элемент дерева не предполагает создание	Проверить правильность задания идентификатора элемента
Inconsistent value	Неправильно значение элемента при установке	Проверить правильность устанавливаемого значения элемента
Resource unavailable	Некоторый ресурс (например, память) не могут быть использованы в настоящий момент	Повторить попытку
Commit failed	Произошла ошибка в процессе записи значения на этапе фиксации изменений	Повторить попытку
Authorization error	Ошибка авторизации	Проверить параметры авторизации
Not writable	Изменение элемента не	

Сообщение	Причина	Способ устранения
	доступно	
Inconsistent name	Неправильное имя объекта при записи (например, неправильный индекс таблицы)	Проверить правильность имени
Error parse incoming message	Ошибка протокола	Повторить попытку
Unavailable context	Контекст временно не доступен	Повторить попытку
Unknown context	Запрашиваемый контекст не существует	Проверить правильность имени контекста
Unknown security model	Не поддерживаемая модель безопасности	Проверить правильность задания модели безопасности
Invalid message	Неправильная структура пакета	Повторить попытку
Unknown PDU handlers	Неизвестный тип пакета	Повторить попытку
Unsupported security level	Не поддерживаемый уровень безопасности	Проверить правильность задания уровня безопасности и модели безопасности
Not in time window	Рассинхронизация времени посылки сообщения	Повторить попытку
Unknown security name	Неизвестное имя пользователя	Проверить правильность задания имени пользователя
Unknown engineID	Заданное значение EngineID не совпадает с EngineID агента	Проверить правильность задания идентификатора агента
Authentication failure	Неверная электронная подпись сообщения	Проверить правильность параметров электронной подписи (имя пользователя, алгоритм и пароль)
Decryption error	Ошибка расшифровки сообщения	Проверить правильность параметров шифрования (имя пользователя, алгоритм шифрования и пароль)

Приложение

Пример конфигурационного файла

```
#
# An SNMP engine's administratively-unique identifier.
#
snmpEngineID 8F270000046167656E7431

listen 192.168.5.8/1161

SysDescr    SNMP agent.
SysObjectID 1.3.6.1.4.1.9999
SysLocation  Right here, right now.
SysName     agent1
SysContact  me <me@somewhere.org>

#
# The table of users configured in the SNMP engine's
# Local Configuration Datastore (LCD).
#
# user SEC_NAME [AUTH AUTH_PASSWORD [PRIV PRIV_PASSWORD]]
usmUser guest
usmUser test1
usmUser test2 MD5 testtest
usmUser test3 MD5 testtest DES testtest
# recommended template clone-from users [RFC3414]
usmUser templateMD5 MD5 "" DES ""
usmUser templateSHA SHA "" DES ""

#
# This table maps a combination of securityModel and
# securityName into a groupName which is used to define
# an access control policy for a group of principals.
#
# group GROUP SEC_MODEL SEC_NAME
group MyRWGroup v1 local
group MyRWGroup v2c local
group MyRWGroup usm local
group MyROGroup v1 guest
group MyROGroup v1 mynetwork
group MyROGroup v2c mynetwork
```

```
group MyROGroup usm mynetwork
group initial usm test1
group initial usm test2
group initial usm test3
group initial v1 guest
group initial v2c guest
group initial v1 admin
group initial v2c admin
group initial usm initial

#
# Locally held information about families of subtrees
# within MIB views.
#
# view VIEW TYPE SUBTREE
view all included 1
view internet included 1.3.6.1
view restricted included 1.3.6.1

#
# The table of access rights for groups.
#
# access GROUP CONTEXT SEC_MODEL SEC_LEVEL MATCH READ WRITE NOTIFY
access MyROGroup "" v1 noAuthNoPriv exact all none none
access MyROGroup "" v2c noAuthNoPriv exact all none none
access MyROGroup "" usm noAuthNoPriv exact all none none
access MyRWGroup "" v1 noAuthNoPriv exact all all none
access MyRWGroup "" v2c noAuthNoPriv exact all all none
access MyRWGroup "" usm noAuthNoPriv exact all all none
access initial "" v1 noAuthNoPriv exact restricted restricted restricted
access initial "" v2c noAuthNoPriv exact restricted restricted restricted
access initial "" usm noAuthNoPriv exact restricted none restricted
access initial "" usm authNoPriv exact internet internet internet
access initial "" usm authPriv exact internet internet internet

#
# The table of community strings configured in the SNMP
# engine's Local Configuration Datastore (LCD).
#
# community COMMUNITY SEC_NAME CONTEXT TARGET_TAG [CONTEXT_ENGINE_ID]
```

Приложение

```
community public guest "" ""
community private admin "" ""

#
# A table of transport addresses to be used in the generation
# of SNMP messages.
#
# snmpTargetAddr TARGET TDOMAIN TADDRESS TIMEOUT RETRIES TAGLIST PARAMS
snmpTargetAddr trap-receiver udp 192.168.5.76/162 5 5 "manager all" secure
snmpTargetAddr server1      udp 192.168.1.1/161 5 5 "server all" secure
snmpTargetAddr server2      udp 192.168.1.2/161 5 5 "server all" secure
snmpTargetAddr manager      udp 127.0.0.1/162 5 5 "manager all"
unsecure

#
# A table of SNMP target information to be used
# in the generation of SNMP messages.
#
# snmpTargetParams PARAMS MP SECMODEL SECNAME SECLEVEL
snmpTargetParams unsecure v1 v1 guest noAuthNoPriv
snmpTargetParams secure v3 usm admin authPriv

# snmpNotify NOTIFY TAG TYPE
snmpNotify all all inform

# snmpNotifyFilterProfile PARAMS FILTER
snmpNotifyFilterProfile secure secure-filter
snmpNotifyFilterProfile unsecure unsecure-filter

# snmpNotifyFilter FILTER INCLUDE SUBTREE MASK
snmpNotifyFilter secure-filter include 1.3.6.1 F0

# LinService SERVICE PATH
LinService test /home/fbv/projects/snmp/linsnmp/tests/mytest
LinService kernel /home/fbv/linter/bin/linter
LinService linapid /home/fbv/linter/bin/linapid
LinService dbs_tcp /home/fbv/linter/bin/dbs_tcp

# LinServiceParam SERVICE NAME VALUE
LinServiceParam kernel pool 1000
```

```
LinServiceParam kernel debug ""
LinServiceParam linapid port 1090
LinServiceParam dbs_tcp C ""

# LinServiceEnv SERVICE NAME VALUE
LinServiceEnv kernel LINTER_MBX 1234
```