СИСТЕМА
УПРАВЛЕНИЯ
БАЗАМИ
ДАННЫХ

ЛИНТЕР®

ЛИНТЕР БАСТИОН

ЛИНТЕР СТАНДАРТ

Удалённое управление компонентами СУБД

НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ

РЕЛЭКС

Товарные знаки

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

Интеллектуальная собственность

Правообладателем продуктов ЛИНТЕР® является компания РЕЛЭКС (1990-2026). Все права защищены.

Данный документ является результатом интеллектуальной деятельности, права на который принадлежат компании РЕЛЭКС.

Все материалы данного документа, а также его части/разделы могут свободно размещаться на любых сетевых ресурсах при условии указания на них источника документа и активных ссылок на сайты компании РЕЛЭКС: relex.ru и linter.ru.

При использовании любого материала из данного документа несетевым/печатным изданием обязательно указание в этом издании источника материала и ссылок на сайты компании РЕЛЭКС: relex.ru и linter.ru.

Цитирование информации из данного документа в средствах массовой информации допускается при обязательном упоминании первоисточника информации и компании РЕЛЭКС.

Любое использование в коммерческих целях информации из данного документа, включая (но не ограничиваясь этим) воспроизведение, передачу, преобразование, сохранение в системе поиска информации, перевод на другой (в том числе компьютерный) язык в какой-либо форме, какими-либо средствами, электронными, механическими, магнитными, оптическими, химическими, ручными или иными, запрещено без предварительного письменного разрешения компании РЕЛЭКС.

О документе

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

Контактные данные

394006, Россия, г. Воронеж, ул. Бахметьева, 2Б.

Тел./факс: (473) 2-711-711, 2-778-333.

e-mail: info@linter.ru.

Техническая поддержка

С целью повышения качества программного продукта ЛИНТЕР и предоставляемых услуг в компании РЕЛЭКС действует автоматизированная система учёта и обработки пользовательских рекламаций. Обо всех обнаруженных недостатках и ошибках в программном продукте и/или документации на него просим сообщать нам в раздел Поддержка на сайте ЛИНТЕР.

Содержание

Предисловие 2
Назначение документа 2
Для кого предназначен документ 2
Необходимые предварительные знания 2
Общие сведения 3
Необходимые условия 5
Структура системы 6
Настройка системы 9
Конфигурационный файл 9
Переменные информационной секции 9
SysDescr 9
SysLocation 10
SysName 10
SysContact 10
Переменные секции управления агентом 10
listen 10
snmpEngineID 11
usmUser 12
group 13
view 13
access 14
Переменные секции управления компонентами 15
LinService 16
LinServiceParam 16
LinServiceEnv 17
Проверка системы 18
Управление системой 19
Агент удалённого управления 19
Утилиты администрирования 21
Общие ключи утилит 21
Общие ключи для всех утилит, работающих по протоколу SNMPv1 и SNMPv2 21
Общие ключи для всех утилит, работающих по протоколу SNMPv3 21
Утилита snmpget 22
Утилита snmpgetnext 23
Утилита snmpset 23
Утилита linsnmp 24
Утилита snmpwalk 29
Утилита snmptrap 29
Утилита snmptrapd 30
Утилита snmpusm 31
Утилита snmpvacm 32
Сообщения системы 34
Приложение 1. Пример конфигурационного файла 36
Приложение 2. Пользовательский сценарий подключения к СУБД 39
Приложение 3. Пользовательский сценарий работы с СУБД 40

Предисловие

Назначение документа

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

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

Документ предназначен для СУБД ЛИНТЕР БАСТИОН 6.0 сборка 20.8, далее по тексту СУБД ЛИНТЕР.

Для кого предназначен документ

Документ предназначен для администраторов СУБД ЛИНТЕР и разработчиков клиентских приложений.

Необходимые предварительные знания

Для работы с 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).

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 компонента, 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 посылки 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, но без кодирования его тела;
- `authPriv`: подпись сообщения и кодирование тела сообщений.

Необходимые условия

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

Структура системы

В системе удаленного 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, то есть 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 определенны во внутренних таблицах с помощью SNMP-NOTIFICATION-MIB.

Примечание
Получатели определены в таблице `snmpTargetAddrTable`.

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

В то же 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 (рисунок).

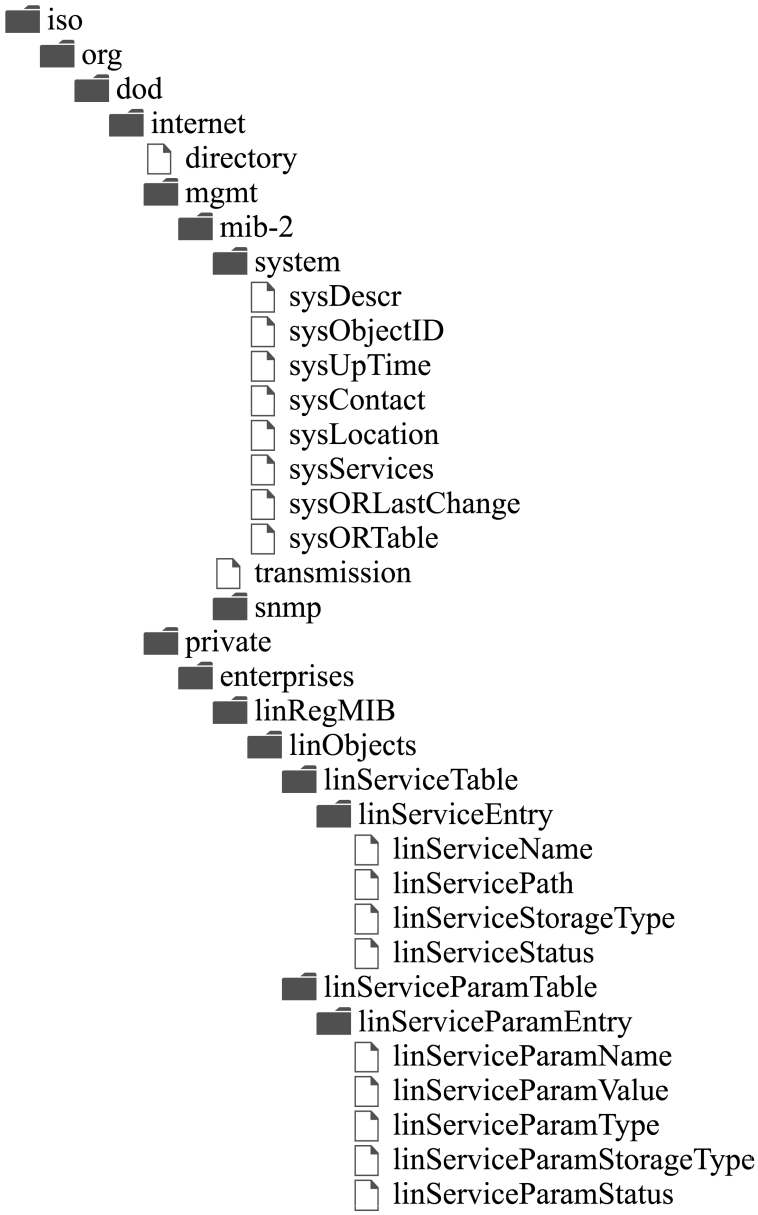


Рисунок. Пример MIB дерева распределенного управления компонентами СУБД ЛИНТЕР

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 древа. 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 большинство 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 распределенный 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 три 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 может 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

Пример конфигурационного файла приведен в приложении 1.

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   A_Склад
2) SysName   "A Склад"
(имя задано в кавычках, так как в написании имени встречается
 пробел)
```

SysContact

Назначение

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

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.

 © Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026

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) Listen192.168.5.8/1161
2) Listen192.168.5.8 по умолчанию используется  192.168.5.8/1161
3) Listen/162 по умолчанию используется  127.0.0.1/162
```

snmpEngineID

Назначение

Задает 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 (то есть строкой нулевой длины) или состоять из всех нулей или значений 'ff'H.

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

Примечание
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 РЕЛЭКС 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 РЕЛЭКС теперь должно быть представлено как '800002b8'H.

Пятый байт указывает, как сформированы остальные байты (шестой 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 задаваема 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, так как сохраняется только на время сеанса работы агента.

Пример

```
EngineID 8F270000046167656E7431
```

usmUser

Назначение

Задает создание пользователя данного агента. Имя пользователя должно быть уникальным в пределах сети. При указании протокола аутентификации и кодирования и соответствующих паролей для пользователя будут сгенерированы локализованные ключи для безопасной передачи сообщений с помощью указанных протоколов.

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

Синтаксис

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

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

 © Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026

```
<параметры кодирования>::= <протокол кодирования> <пароль>

<протокол аутентификации>::= MD5 | SNA

<протокол кодирования>::= DES

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

<пароль>::= не пустая символьная строка
```

Примеры

```
1) usmUser  "A.  B.  Петров"
2) usmUser  sys_admin
3) usmUser  user1 MD5 Fc76$gv
4) usmUser  user2 MD5 Fc76$gv DES acфrst
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 "A. Б. 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>::= ""
```

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

 © Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026

<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 кодирования;
   - authPriv: подпись сообщения и кодирование 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, kernel60, 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).

Примечания

1. Перечисленные таблицы в системе задействованы, из конфигурационного файла их создавать нельзя, то есть секции LinService, LinServiceParam, LinServiceEnv СУБД ЛИНТЕР не поддерживаются.
2. 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

Задает 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>
```

 © Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026

<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 127.0.0.1/1161 1.3.6.1.2.1.1.1.0
1.3.6.1.2.1.1.1.0 = (OCTET STRING) SNMP agent.
```

Пример пользовательского сценария подключения к СУБД рассмотрен в приложении 2.

Пример пользовательского сценария работы с СУБД рассмотрен в приложении 3.

Управление системой

Агент удалённого управления

Командная строка запуска

Запуск агента осуществляется стандартными средствами запуска задач операционной системы.

Примечание
В ОС Windows для управления агентом удаленного управления при инсталляции СУБД ЛИНТЕР был создан сервис. Его запуск и останов необходимо выполнять из окна управления службами, имя службы - LinSNMPAgt. Для открытия окна управления службами необходимо в меню ПУСК выбрать Панель управления, затем пункт Администрирование и далее выбрать Службы.

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.

Примечание
Рекомендуется запускать как сервис.

Ключи агента

`-c [<config-файл>]`

Задает 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>]`

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

`-p <pid>`

Задает 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
```

Примечание
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 /home/tester/linter/bin/snmp.txt
linsnmpd -c snmp.conf -l snmp.log
```

 © Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026

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`

Задает версию протокола SNMPv1. Используется по умолчанию.

`-v2c`

Задает версию протокола SNMPv2c.

`-v3`

Задает версию протокола SNMPv3.

`-r <количество повторов>`

Задает количество попыток отправки запроса. По умолчанию три попытки.

`-t <тайм-аут>`

Задает длительность ожидания ответа (в секундах). По умолчанию 1.5 секунды.

`-m MIB[:…]`

Загрузка указанного списка MIB файлов (ALL – загрузка всех файлов MIB).

`-M DIR[:…]`

Загрузка MIB файлов из списка указанных директорий.

`-po`

Старое значение режима безопасности.

Примечание
Имена ключей утилит регистрозависимы.

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

`-c <сообщество>`

Задает сообщество пользователей агента. По умолчанию `public`.

Общие ключи для всех утилит, работающих по протоколу SNMPv3

`-a <протокол аутентификации>`

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

`-A <пароль аутентификации>`

Задает пароль протокола аутентификации.

`-e <ID-агента>`

Задает идентификатор агента (например, 800000020109840301). Значения по умолчанию нет. Если идентификатор не указан, то он определяется автоматически.

`-E <ID-контекста>`

Задает идентификатор контекста агента, например, 800000020109840301. По умолчанию используется идентификатор агента.

`-l <уровень безопасности>`

Задает уровень безопасности (`noAuthNoPriv | authNoPriv | authPriv`). По умолчанию `noAuthNoPriv`.

`-n <контекст>`

Задает имя контекста, по умолчанию используется пустая строка.

`-u <имя пользователя>`

Задает имя пользователя агента.

`-x <протокол кодирования>`

Задает протокол кодирования (DES).

`-X <пароль кодирования>`

Задает пароль протокола кодирования.

`-Z <количество>,<время>`

Задает 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;

 © Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026

- 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 127.0.0.1/1161 1.3.6.1.2.1.1.1.0
1.3.6.1.2.1.1.1.0 = <OCTET 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 127.0.0.1/1161 1.3.6.1.2.1.1.5.0
1.3.6.1.2.1.1.5.0 = <OCTET 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 127.0.0.1/1161 1.3.6.1.2.1.1.1
1.3.6.1.2.1.1.1.0 = <OCTET STRING> My agent
```

2)

```
$ snmpgetnext -v 1 -c public 127.0.0.1/1161 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;

- s: STRING;
- o: OBJID.

Пример

```
$ snmpset -v 1 -c private 127.0.0.1/1161 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 dist
```

Вывод списка переменных, хранящих пути к установочным каталогам СУБД ЛИНТЕР на ЛИНТЕР-сервере и их значений.

Пример

```
$linsnmp -v 1 -c public 127.0.0.1/1161 show dist
Distributives:
LINTER          full    C:/Linter
```

```
dist DIST PRODUCT TYPE PATH
```

Создать или изменить значение переменной, хранящей путь к установочному каталогу СУБД ЛИНТЕР на ЛИНТЕР-сервере.

```
no dist DIST
```

Удаление переменной, хранящей путь к установочному каталогу СУБД ЛИНТЕР.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 no dist LINTER1
Deleting distributive "LINTER1"... ok
```

```
show db DIST
```

Вывод списка БД, созданных в установочном каталоге СУБД ЛИНТЕР, путь к которому хранится в переменной DIST на удаленном ЛИНТЕР-сервере.

Пример

```
linsnmp -v 1 -c private 127.0.0.1/1161 show db LINTER
Databases:
Demo    active
```

 © Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026

```
db DIST.DB
```

Определить новую БД с именем DB, используя приложения из указанной в переменной DIST установочного каталога СУБД ЛИНТЕР на удаленном ЛИНТЕР-сервере.

Пример

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.TEST_DB
Creating database "LINTER.TEST_DB"... ok
```

```
db DIST.DB create
```

Создать новую БД с именем DB, используя приложения из указанной в DIST установочного каталога СУБД ЛИНТЕР на удаленном ЛИНТЕР-сервере.

Пример

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.TEST_DB create
Creating database "LINTER.TEST_DB"... ok
Creating database files "LINTER.TEST_DB"... ok
```

```
db DIST.DB attach
```

Подключить существующую БД с именем DB, находящуюся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге `/db` на удаленном ЛИНТЕР-сервере. Для БД будут созданы соответствующие сервисы (службы).

Примечание
Перед подключением к существующей БД необходимо выполнить команду определения БД (`db DIST.DB`).

```
storage DIST.DB attach
```

Подключить существующую БД-хранилище с именем DB (storage), находящуюся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге `/db` на удаленном ЛИНТЕР-сервере. Для БД не будут созданы соответствующие сервисы (службы).

Пример

```
linsnmp -v 1 -c private 127.0.0.1/1162 storage LINTER.storage
 attach
```

```
db LINTER.DB path
```

Объявить БД с именем DB, расположенную по указанному в значении path каталоге на удаленном ЛИНТЕР-сервере.

Пример

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.Demo path
/home/username/linter/db
Результат:
Defining database "LINTER.Demo"... ok
Updating database parameter "LINTER.Demo.SY00"... ok
```

```
db DIST.DB update
```

Применить измененные значения переменных указанной БД.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.TEST_DB update
Updating database "LINTER.TEST_DB"... ok
```

```
db DIST.DB start
```

Запустить все службы БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.TEST_DB start
Starting database "LINTER.TEST_DB"... ok
```

```
db DIST.DB stop
```

Остановить все службы БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.TEST_DB stop
Stopping database "LINTER.TEST_DB"... ok
```

```
db DIST.DB restart
```

Перезапустить все службы БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.TEST_DB restart
Stopping database "LINTER.TEST_DB"... ok
Starting database "LINTER.TEST_DB"... ok
```

```
no db DIST.DB
```

Удалить БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере. Перед удалением БД необходимо выполнить «db DIST.DB stop».

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 no db LINTER.TEST_DB
Deleting database "LINTER.TEST_DB"... ok
```

```
show dbparam DIST.DB
```

Вывод списка параметров БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 show dbparam
 LINTER.TEST_DB
Database "LINTER.TEST_DB" parameters:
SY00    C:/Linter/db/TEST_DB
```

```
PASSWORD        *******
USERNAME        SYSTEM
```

```
dbparam DIST.DB.PARAM VALUE
```

Установить значение параметра PARAM для БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

```
no dbparam DIST.DB.PARAM
```

Удалить параметр PARAM для БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

```
show service DIST.DB
```

Вывод списка сервисов БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db. Для просмотра общих сервисов, не связанных с конкретной БД, необходимо не указывать имя БД на удаленном ЛИНТЕР-сервере.

Примеры

```
1)
$linsnmp -v 1 -c private 127.0.0.1/1161 show service
 LINTER.TEST_DB
Database "LINTER.TEST_DB" services:
kernel.0        notInService
listener.0      notInService
2)
$ linsnmp -v 1 -c private 127.0.0.1/1161 show service LINTER.
Database "LINTER. " services:
netclient.0        notInService
linstmgr.0         notInService
linapid.0          notInService
```

```
service DIST.DB.SERVICE.SUBID
```

Создание сервиса с именем SERVICE для БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 service
 LINTER.TEST_DB.kernel.0
Creating service "LINTER.TEST_DB.kernel.0"... ok
```

```
service DIST.DB.SERVICE.SUBID start
```

Запуск сервиса с именем SERVICE для БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 service
 LINTER.TEST_DB.kernel.0 start
```

```
Starting service "LINTER.TEST_DB.kernel.0"... ok
```

```
service DIST.DB.SERVICE.SUBID stop
```

Останов сервиса с именем SERVICE для БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 service
 LINTER.TEST_DB.kernel.0 stop
Stopping service "LINTER.TEST_DB.kernel.0"... ok
```

```
service DIST.DB.SERVICE.SUBID restart
```

Перезапуск сервиса с именем SERVICE для БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -c private 127.0.0.1/1161 service
 LINTER.TEST_DB.kernel.0 restart
Stopping service "LINTER.TEST_DB.kernel.0"... ok
Starting service "LINTER.TEST_DB.kernel.0"... ok
```

```
no service DIST.DB.SERVICE.SUBID
```

Удаление сервиса с именем SERVICE для БД с именем DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -t 10 -c private 127.0.0.1/1161 no service
 LINTER.TEST_DB.kernel.0
Deleting service "LINTER.TEST_DB.kernel.0"... ok
```

```
show param DIST.DB.SERVICE.SUBID
```

Получение списка параметров сервиса SERVICE для БД DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге /db на удаленном ЛИНТЕР-сервере.

Пример

```
$linsnmp -v 1 -t 10 -c private 127.0.0.1/1161 show param
 LINTER.TEST_DB.kernel.0
Service "LINTER.TEST_DB.kernel.0" parameters:
LOG
SPX     0
TCP     0
BASE    C:/Linter/db/TEST_DB
NAME    TEST_DB
SY00    C:/Linter/db/TEST_DB
SYNC
```

```
LOGALL
NONAME
LOGQUERY
TCORRECT
AUTOINDEX
LINTER_MBX      TEST_DB
```

```
param DIST.DB.SERVICE.SUBID.PARAM TYPE VALUE
```

Установка значения параметра PARAM сервиса SERVICE для БД DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге `/db` на удаленном ЛИНТЕР-сервере.

```
no param DIST.DB.SERVICE.SUBID.PARAM
```

Удаление параметра PARAM сервиса SERVICE для БД DB, находящейся в указанном в DIST установочном каталоге СУБД ЛИНТЕР в подкаталоге `/db` на удаленном ЛИНТЕР-сервере.

Утилита snmpwalk

Назначение

Получение значений всех потомков заданного узла дерева MIB.

Команда запуска

```
snmpwalk [<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).

 © Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026

Утилита snmpusm

Назначение

Управление правами SNMPv3 пользователей.

Команда запуска

```
snmpusm [<ключ>…] <адрес/порт> <команда>
```

<команда> – одна из нижеследующих команд:

`showcounters`

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

`showusers`

Отображать пользователей.

`create <имя>`

Создание нового пользователя.

`clonefrom <имя> <прототип>`

Создание нового пользователя по образцу прототипа.

Примечание
При создании пользователя путём клонирования из шаблонного пользователя будет выполнена попытка заменить пустые пароли на новые, сгенерированные случайным образом, если пароли не пустые, то генерации пароля не будет. Будет выдано сообщение о замене пароля.

`delete <имя>`

Удаление пользователя.

`activate <имя>`

Активация пользователя.

`deactivate <имя>`

Отключение пользователя.

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

Изменение пароля аутентификации.

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

Изменение личного пароля.

Примечание
В некоторых ОС возможен конфликт имени утилиты `snmpusm` с одноименной утилитой ОС. Для устранения необходимо вызывать утилиту `snmpusm` в формате:

```
./snmpusm <ключи утилиты>
```

Примеры

1)

```
snmpusm -v 1 -c private -e 800057930430 127.0.0.1/1161
 showcounters
2 unsupported security levels
0 not in time windows
155202 unknown user names
21 unknown EngineIDs
0 wrong digests
0 decryption errors
```

2)

```
snmpusm -v 1 -c private -e 800057930430 127.0.0.1/1161 showusers
800057930430.admin admin Active
…
800057930430.templateSHADES templateSHADES SHA DES Active
```

3)

```
snmpusm -v 1 -c private -e 800057930430 127.0.0.1/1161 create
 test_user
```

4)

```
snmpusm -v 1 -c private -e 800057930430 127.0.0.1/1161 clonefrom
 clon_initial initial
Authentication password was changed from "" to "lw3r3jot"
```

5)

```
snmpusm -v 1 -c private -e 800057930430 127.0.0.1/1161 delete
 clon_initial
```

6)

```
snmpusm -v 1 -c private -e 800057930430 127.0.0.1/1161 activate
 test_user
```

7)

```
snmpusm -v 1 -c private -e 800057930430 127.0.0.1/1161 deactivate
 test_user
```

Утилита snmpvacm

Назначение

Создание и поддержка записей управления доступом SNMPv3 пользователей.

Команда запуска

```
snmpvacm [<ключ>… ] <адрес/порт> <команда>
```

<команда> – одна из нижеследующих команд:

 © Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026

```
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.

```
noview <имя представления>
```

Удаление представления.

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

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

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

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

```
createAccess <имя группы> [<префикс контекста>] <модель> <уровень>
<совпадение контекста> <уровень чтения> <уровень записи> <уровень
оповещения>
```

Назначение доступа к представлению группе.

```
deleteAccess <имя группы> [<префикс контекста>] <модель> <уровень>
```

Отмена доступа к представлению.

Примеры

1)

```
snmpvacm -v 3 -e 800057930430 -l authNoPriv -m ../snmp/mibs/ -a
 MD5 -A
 SecretPass -u UserN 127.0.0.1/1161 view MyView1 1.3.6
```

2)

```
snmpvacm -v 3 -e 800057930430 -l authNoPriv -m ../snmp/mibs/ -a
 MD5 -A
 SecretPass -u UserN 127.0.0.1/1161 deleteAccess GR1 1 1
```

3)

```
snmpvacm -v 3 -e 800057930430 -l authNoPriv -m ../snmp/mibs/ -a
 MD5 -A
 SecretPass -u UserN 127.0.0.1/1161 noview MyView1 1.3.6
```

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`	Неправильное 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

Сообщение	Причина	Способ устранения
`Not writable`	Изменение элемента не 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`	Неподдерживаемая 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 декодирования сообщения	Проверить правильность параметров кодирования (имя пользователя, алгоритм кодирования и пароль)

Приложение 1
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]
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-recever 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
```

Приложение 2
Пользовательский сценарий подключения к СУБД

Примечания

1) На ОС Windows команды выполнять из каталога `%LINTER_HOME%\snmp\`.
2) На ОС Linux, ЗОСРВ Нейтрино команды выполнять из каталога `$LINTER_HOME/bin/`.

Предварительные действия

Для ОС Windows необходимо выполнить операции:

1) отредактировать файл `snmp.conf` из каталога `%LINTER_HOME%\snmp`, изменив строку "listen 0.0.0.0/1161" на "listen 127.0.0.1/1161";
2) запустить службу LinSNMPAgt.

Для ОС Linux, ЗОСРВ Нейтрино необходимо выполнить операции:

1) отредактировать файл `snmp.conf` из каталога `$LINTER_HOME/snmp`, изменить строку "listen 0.0.0.0/1161" на "listen 127.0.0.1/1161";
2) запустить SNMP-агент командой

```
linsnmpd -c ../snmp/snmp.conf
```

Проверка подключения

Для ОС Windows необходимо выполнить операции:

1) убедиться, что служба LinSNMPAgt запущена;
2) в консоли попробовать получить некоторые данные, например, значение sysDesc.0 (1.3.6.1.2.1.1.1.0), содержащее описание удаленного агента

```
snmpget -v 1 -c public 127.0.0.1/1161 1.3.6.1.2.1.1.1.0
```

Если все настроено корректно, будет выведена строка:

```
sysDescr.0 = (OCTET STRING) Linter SNMP agent.
```

Для ОС Linux, ЗОСРВ Нейтрино необходимо выполнить операции:

1) убедиться, что процесс `linsnmpd` запущен:

```
ps -A | grep linsnmpd
28945 pts/1    S+     0:00 grep linsnmpd
```

2) в консоли попробовать получить некоторые данные, например, значение sysDesc.0 (1.3.6.1.2.1.1.1.0), содержащее описание удаленного агента

```
snmpget -v 1 -c public 127.0.0.1/1161 1.3.6.1.2.1.1.1.0
```

Если все настроено корректно, будет выведена строка:

```
sysDescr.0 = (OCTET STRING) Linter SNMP agent.
```

Приложение 3
Пользовательский сценарий работы с СУБД

Создание БД

Для создания БД необходимо выполнить операции:

1) определить новую БД с именем NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.NEWDB
```

2) создать новую БД с именем NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.NEWDB create
```

Убедиться, что в каталоге `%LINTER_HOME%\db` (ОС Windows) или `$LINTER_HOME/db` (ОС Linux, ЗОСРВ Нейтрино) создан подкаталог `NEWDB`, содержащий файлы БД.

Запуск и останов ядра СУБД

Для запуска и останова ядра СУБД необходимо выполнить операции:

1) проверить значения переменной LINTER_MBX для ядра СУБД, запущенного на БД с именем NEWDB (только для ОС Linux, ЗОСРВ Нейтрино)

```
linsnmp -v 1 -c private 127.0.0.1/1161 show param
 LINTER.NEWDB.kernel.0
```

2) запустить все сервисы БД с именем NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.NEWDB start
```

Убедиться, что запустились ядро и сетевой сервер (`dbs_tcp`) на БД NEWDB.

3) в командной строке установить переменную окружения LINTER_MBX=NEWDB для ОС Windows или LINTER_MBX=15783 для ОС Linux, где 15783 – значение переменной LINTER_MBX для ядра СУБД, запущенного на БД с именем NEWDB;

4) в командной строке запустить утилиту `inl` (из подкаталога `bin` установочного каталога)

```
inl -u SYSTEM/MANAGER8
```

5) в `inl` выполнить запросы

```
create table tab (i int, j int);
insert into tab values (10,100);
insert into tab values (20,200);
```

6) остановить все сервисы БД с именем NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.NEWDB stop
```

Настройка параметров БД и сервисов

Для настройки параметров БД и сервисов необходимо выполнить операции:

1) вывести список параметров БД с именем NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 show dbparam LINTER.NEWDB
```

2) изменить параметр БД

```
linsnmp -v 1 -c private 127.0.0.1/1161 dbparam
 LINTER.NEWDB.LINTER_MBX 1234
```

3) вывести список параметров БД с именем NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 show dbparam LINTER.NEWDB
```

4) применить измененные значения переменных БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.NEWDB update
```

5) вывести список параметров сервиса listener для БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 show param
 LINTER.NEWDB.listener.0
```

6) установить значения параметров P и LINTER_MBX сервиса listener для БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 param
 LINTER.NEWDB.listener.0.P cmd 1061
linsnmp -v 1 -c private 127.0.0.1/1161 param
 LINTER.NEWDB.listener.0.LINTER_MBX env 1234
```

7) вывести список параметров сервиса listener для БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 show param
 LINTER.NEWDB.listener.0
```

Проверка настроек параметров

Для проверки настроек параметров необходимо выполнить операции:

1) открыть на редактирование файл `nodetab` из подкаталога `bin` установочного каталога и добавить строку

```
MY TCPIP 127.0.0.1 1061
```

2) запустить все сервисы БД с именем NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.NEWDB start
```

Убедиться, что запустились ядро и сетевой сервер (`dbs_tcp`) на БД NEWDB.

3) запустить сервис netclient

```
linsnmp -v 1 -c private 127.0.0.1/1161 service LINTER..netclient.0
 start
```

4) в командной строке запустить утилиту `inl` (из подкаталога `bin` установочного каталога)

```
inl -u SYSTEM/MANAGER8 -n MY
```

5) в `inl` выполнить запрос

```
select * from tab;
I J
 - -
| 10| 100|
```

```
| 20| 200|
 INL : выдано строк : 2
```

6) остановить все сервисы БД с именем NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.NEWDB stop
```

Убедиться, что оба процесса (ядро и сетевой сервер) остановлены.

7) остановить сервис netclient

```
linsnmp -v 1 -c private 127.0.0.1/1161 service LINTER..netclient.0
 stop
```

Работа с сервисами (создание, запуск, останов, удаление)

Для работы с сервисами (создание, запуск, останов, удаление) необходимо выполнить операции:

1) создать сервис для БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 service
 LINTER.NEWDB.listener.1
```

2) вывести список параметров сервиса listener для БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 show param
 LINTER.NEWDB.listener.1
```

3) запустить сервис listener для БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 service
 LINTER.NEWDB.listener.1 start
```

Убедиться, что запустился сетевой сервер (`dbs_tcp`) на БД NEWDB.

4) перезапустить сервис listener для БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 service
 LINTER.NEWDB.listener.1 restart
```

Убедиться, что процесс была сначала остановлен, а затем запущен снова.

5) остановить сервис listener

```
linsnmp -v 1 -c private 127.0.0.1/1161 service
 LINTER.NEWDB.listener.1 stop
```

6) удалить сервис listener для БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 no service
 LINTER.NEWDB.listener.1
```

Удаление и подключение БД

Для удаления и подключения БД необходимо выполнить операции:

1) удалить БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 no db LINTER.NEWDB
```

2) определить новую БД с именем NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.NEWDB
```

3) подключить существующую БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 db LINTER.NEWDB attach
```

Работа с пользователями (создание, добавление в группы, изменение паролей)

Для работы с пользователями (создание, добавление в группы, изменение паролей) необходимо выполнить операции:

1) создать пользователя newuser

```
snmpusm -v 3 -u admin -e 800057930430 127.0.0.1/1161 create
 newuser
```

2) клонировать пользователя

```
snmpusm -po -v 3 -u admin -e 800057930430 127.0.0.1/1161 clonefrom
 user3 templateMD5DES
```

3) изменить пароль аутентификации и личный пароль

```
snmpusm -po -v 3 -u admin -e 800057930430 127.0.0.1/1161
 privpasswd user3 "" "user3 PRIV"
snmpusm -po -v 3 -u admin -e 800057930430 127.0.0.1/1161
 authpasswd user3 "" "user3 AUTH"
```

4) добавить пользователя в группу "Read-Write group"

```
snmpvacm -v 3 -u admin -e 800057930430 127.0.0.1/1161 group
 RWGroup 3 user3
```

5) активировать пользователя

```
snmpusm -v 3 -u admin -e 800057930430 127.0.0.1/1161 activate
 user3
```

6) проверить корректность установки паролей

```
snmpget -v 3 -u user3 -e 800057930430 -l authPriv -a MD5 -x DES -A
 "user3 AUTH" -X "user3 PRIV" -M mibs 127.0.0.1/1161 sysName.0
```

Если все установлено корректно, будет выведена строка:

```
sysName.0 = (OCTET STRING) agent
```

7) деактивировать пользователя

```
snmpusm -v 3 -u admin -e 800057930430 127.0.0.1/1161 deactivate
 user3
```

8) удалить пользователя из группы "Read-Write group"

```
snmpvacm -v 3 -u admin -e 800057930430 127.0.0.1/1161 nogroup
 RWGroup 3 user3
```

9) удалить пользователя

```
snmpusm -v 3 -u admin -e 800057930430 127.0.0.1/1161 delete user3
```

Завершение проверки (очистка за собой, приведение в первоначальное состояние)

Для завершения проверки (очистка за собой, приведение в первоначальное состояние) необходимо выполнить операции:

1) удалить БД NEWDB

```
linsnmp -v 1 -c private 127.0.0.1/1161 no db LINTER.NEWDB
```

2) удалить сервис netclient (на ОС Linux, ЗОСРВ Нейтрино)

```
linsnmp -v 1 -c private 127.0.0.1/1161 no service
 LINTER..netclient.0
```

3) остановить SNMP-агент:

   на ОС Windows остановить службу LinSNMPAgt, на ОС Linux и ЗОСРВ Нейтрино остановить процесс `linsnmpd`:

```
ps -A | grep linsnmpd
28945 pts/1    S+     0:00 grep linsnmpd
kill 28945
```


© Удалённое управление компонентами СУБД. АО НПП «РЕЛЭКС», 1990-2026