СИСТЕМА
УПРАВЛЕНИЯ
БАЗАМИ
ДАННЫХ

ЛИНТЕР®

ЛИНТЕР БАСТИОН

ЛИНТЕР СТАНДАРТ

Запуск и останов СУБД ЛИНТЕР в среде ОС Linux

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
Дополнительные документы 2
Запуск СУБД ЛИНТЕР 3
Порядок запуска 3
Запуск СУБД с поддержкой Kerberos 5
Сетевой режим работы 5
Ключи доступа к БД 6
Ключи доступа к защищенной БД 7
Ключи управления оперативной памятью 8
Ключи управления функционированием 10
Ключи сетевых настроек 15
Ключи совместимости с SQL-стандартом и другими СУБД 15
Ключи оптимизации обработки SQL-запросов 19
Ключи протоколирования работы ядра СУБД 20
Ключи LDAP-аутентификации 27
Ключи KERBEROS-аутентификации 28
Ключи проверки соответствия мандатного доступа 28
Информационные ключи 28
Переменные среды окружения 28
Общие переменные 29
Переменные идентификации и аутентификации по LDAP-протоколу 30
Переменные идентификации и аутентификации по Kerberos-протоколу 34
Значения размеров очередей, задаваемые по умолчанию 34
Останов СУБД ЛИНТЕР 35
Пользовательский останов ядра СУБД ЛИНТЕР 35
Системный останов ядра СУБД ЛИНТЕР 36
Синхронизация завершения работы СУБД 37
Коды завершения 38
Коды завершения ядра СУБД 38
Код завершения 1 38
Код завершения 2 38
Код завершения 3 39
Код завершения 4 39
Код завершения 5 39
Код завершения 6 40
Код завершения 7 40
Код завершения 8 41
Код завершения 9 41
Код завершения 10 41
Код завершения 11 42
Код завершения 12 42
Код завершения 13 42
Код завершения 15 43
Коды завершения программы останова СУБД 43
Приложение 1. Пример настройки СУБД ЛИНТЕР для идентификации и аутентификации по Kerberos-протоколу 44
Приложение 2. Описание файла linter.out 47
Приложение 3. Описание файла LINTER.LOG 49
Указатель ключей 52

Предисловие

Назначение документа

Документ содержит описание процедуры запуска и останова СУБД ЛИНТЕР для ОС Linux и ЗОСРВ Нейтрино.

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

Для кого предназначен документ

Документ предназначен для системных администраторов и пользователей СУБД ЛИНТЕР.

Необходимые предварительные знания

- Ознакомиться с первой главой документации «Архитектура СУБД».
- Иметь навыки работы и администрирования в соответствующей ОС.

Дополнительные документы

- Архитектура СУБД
- Администрирование комплекса средств защиты данных
- Создание и конфигурирование базы данных
- Справочник по SQL
- Сетевые средства
- Процедурный язык
- Интерфейс нижнего уровня
- Миграция базы данных
- Конвертер баз данных
- Тестирование базы данных
- Архивирование и восстановление базы данных

Запуск СУБД ЛИНТЕР

Запуск СУБД ЛИНТЕР активизирует ядро СУБД, которое отвечает за управление данными во внешней памяти, управление буферами оперативной памяти, управление транзакциями и журнализацию. Кроме того, ядро включает в себя такие процессы, как SQL-транслятор, транслятор хранимых процедур и триггеров, процессор сортировки. Ядро является основной резидентной частью и основной составляющей серверной части СУБД. При запуске ядро СУБД ЛИНТЕР может использовать переменные среды окружения (см. раздел Переменные среды окружения).

Порядок запуска

Для запуска СУБД ЛИНТЕР:

1) указать путь к БД одним из следующих способов:
   - установить переменную окружения SY00 так, чтобы ее значение указывало на каталог БД;
   - указать путь при запуске ядра ЛИНТЕР в параметре /BASE;
   - сменить текущий каталог на каталог БД.
2) запустить исполняемый файл `linter` ядра СУБД ЛИНТЕР.

При этом каталог `/bin` должен быть включен в список каталогов переменной окружения `PATH`, или для запуска можно использовать полный абсолютный либо относительный путь к исполняемому файлу `linter`.

Примечания

1. Если при запуске ядра СУБД будет получено сообщение `error 5, can't lock sysrel index file`, значит, ядро СУБД уже запущено.
2. Для проверки лицензии используется реальная текущая дата. Если БД была запущена с будущей датой, срок лицензии считается истекшим, даже если вернуть дату назад.

При успешном запуске на консоль выводятся параметры и установленные режимы работы ядра:

```
Database creation time: 02.12.2025 08:37:22.48
Last database startup time:  09.12.2025 07:02:49.74
Last database shutdown time: 09.12.2025 07:02:52.77
Linter SQL Bastion v. 6.0.20.6 connected to data base "LINTER
 Database  "
64-bit build
 SY00 is "/usr/linter/db/"
 Database meets safety certification requirements
 Superuser mode is off
 Memory clearing is on
 POOL holds 207234 pages (203848 free pages, 0 pages for in-memory
 tables)
 Table    queue size  : 175 (0)
```

```
 Column   queue size  : 1046 (0)
 Channel  queue size  : 100
 File     queue size  : 350 (0)
 User     queue size  : 100
 Checking stored procedures
 Auditing disabled
 Transaction control is turned on
 Max concurrent sorting processes: 1 (sorting pool holds 25008
 blocks, 16 blocks per page)
 Record size limit: 65535 bytes
 Channel memory limit: 256 pages
 Phrase index pool size: 2000 pages
 Event queue size: 128 items
 Existing control point(s) in database:
 Control point list is empty.
 Kernel system parameters:  MBX - "20561", Pid - 770392
 INFO: JDBC listener started at 1070 port
 Copyright (C) 1990-2025 Relex, Inc. All rights reserved.
 JDBC server v. 2.0
 Bound to port 1070 (0x42E)
 Linter kernel command line: /usr/linter/bin/linter /BASE=/usr/
linter/db /POOL=100000 /SPOOL=25000 /TCP=1060 /JDBCS /JDBCP=1070
*** Linter OLTP v4.00  Copyright (C) 1990-2025 Relex, Inc. All
 rights reserved. ***
*** Linter transaction server (OLTP) started ***
*** RDBMS Linter is running
*** Press <ENTER> for shell prompt
```

При запуске СУБД ЛИНТЕР создаются пять задач:

1) `linter` – ядро СУБД;
2) `sql` – транслятор запросов;
3) `intsrt` – процессор сортировки;
4) `tsp` – процессор хранимых процедур;
5) `loltp` – процессор распределенных запросов (может отсутствовать в списке запущенных задач);
6) `dbs_tcp` – сетевой драйвер сервера (может отсутствовать в списке запущенных задач).

Чтобы разрешить удаленный доступ клиентских приложений к данной БД, необходимо выполнить команду:

```
grant access on unlisted station to all;
```

(разрешить доступ к этой БД со всех компьютеров), либо команду на создание станции (разрешить доступ к этой БД только с конкретных компьютеров) (см. документ «Администрирование комплекса средств защиты данных», раздел «Контроль доступа к БД с сетевых станций»).

Командная строка запуска СУБД ЛИНТЕР имеет следующий синтаксис:

```
linter [/<ключ> …]
```

Запуск СУБД с поддержкой Kerberos

Для запуска СУБД с поддержкой идентификации и аутентификации по Kerberos-протоколу необходимы следующие условия:

- ядро СУБД ЛИНТЕР должно поддерживать идентификацию и аутентификацию по Kerberos-протоколу;
- должна быть определена переменная окружения LINTER_KRB_SERVICE, задающая имя службы СУБД ЛИНТЕР. Значение переменной должно соответствовать имени сервиса СУБД ЛИНТЕР в терминах Kerberos-сервера. Это значение передается клиентскому приложению и используется как имя сервиса, для которого необходимо выполнить аутентификацию (приложение 1).

  Имя сервиса прописывается в виде `linter@server.domain` либо `linter/server.domain`, где `linter` – имя службы, `server.domain` – доменное имя сервера, на котором работает ядро СУБД ЛИНТЕР;
- перед первым запуском СУБД ЛИНТЕР с помощью средств администрирования Kerberos-сервера должен быть сгенерирован и сохранен ключ сервиса ЛИНТЕР либо в файле Kerberos-ключей по умолчанию, либо в любом другом файле. Если ключ был сохранен не в файле ключей по умолчанию, то местоположение этого файла должно быть задано перед запуском ядра СУБД ЛИНТЕР в переменной окружения KRB5_KTNAME. Kerberos-сервер должен быть настроен на аутентификацию службы, указанной в переменной окружения LINTER_KRB_SERVICE.

Пример настройки аутентификации по Kerberos-протоколу приведен в приложении 1.

Сетевой режим работы

Для настройки работы СУБД ЛИНТЕР в сетевом режиме кроме запуска ядра СУБД необходимо запустить сетевые средства: сетевой драйвер сервера и сетевой драйвер клиента. Порядок запуска компонентов не имеет значения. Можно запускать в последовательности

```
ядро СУБД->драйвер сервера->драйвер клиента
```

либо

```
драйвер клиента->драйвер сервера->ядро СУБД
```

Драйвера, так же как и ядро СУБД, могут быть запущены как приложение ОС или как сервисы ОС.

Чтобы разрешить доступ клиентских приложений к удаленной БД, необходимо на ЛИНТЕР-сервере, где размещается эта БД, в режиме локального доступа к ней выполнить команду:

```
grant access on unlisted station to all;
```

(разрешить доступ к этой БД со всех компьютеров), либо команду на создание станции (разрешить доступ к этой БД только с конкретных компьютеров) (см. документ «Администрирование комплекса средств защиты данных», раздел «Контроль доступа к БД с сетевых станций»).

Драйвер сервера

Драйвер сервера предназначен для обслуживания запросов удаленных клиентов на сервере базы данных.

Запустить сетевой драйвер сервера можно одним из способов:

- указать при запуске ядра СУБД ключ `/tcp[=<номер порта>]` (по умолчанию номер порта `1060`);
- выполнить из подкаталога `/bin` установочного каталога СУБД команду:

```
dbs_tcp -P=<номер порта>
```

По умолчанию номер порта `1060`.

При этом сетевой сервер будет запущен как приложение ОС.

Драйвер клиента

Драйвер клиента предназначен для обслуживания запросов локальных клиентов и серверов на локальном сервере БД. Он должен запускаться на том компьютере, на котором будет функционировать приложение. Драйвер можно запускать при активном или неактивном состоянии СУБД, но обязательно до запуска приложения. Перед запуском драйвера клиента необходимо внести в файл сетевой конфигурации `nodetab`, расположенном в подкаталоге bin установочного каталога СУБД данные о серверах с которыми планируется работа (см. «Сетевые средства», пункт «Файл сетевой конфигурации клиента»).

Запустить сетевой драйвер клиента можно одним из способов:

- создать собственный сервис ОС с командой запуска утилиты;
- выполнить из подкаталога `/bin` установочного каталога СУБД команду:

```
dbc_tcp -s=<имя ЛИНТЕР-сервера>
```

При этом сетевой драйвер клиента будет запущен как приложение ОС.

Ключи доступа к БД

```
/U=<имя>/<пароль>
```

Задает регистрационные данные пользователя БД. При старте ядра СУБД будет проверяться наличие данного пользователя. Если пользователь является создателем БД или зарегистрирован в БД с привилегией `DBA`, то ядро СУБД будет запущено. В противном случае происходит отказ в запуске ядра СУБД.

```
/BASE=<строка>
```

Задает полный путь к БД. Аналог переменной окружения SY00.

```
/CF=<спецификация файла>
```

Примечание
Поддерживается со сборки 6.0.17.92.

Задает имя и путь к текстовому файлу, содержащему ключи командной строки запуска ядра СУБД ЛИНТЕР.

Характеристика файла:

- файл может содержать несколько строк;
- строки, начинающиеся с символа #, считаются комментарием и не обрабатываются;
- считывание ключей из файла выполняется слева направо;
- если файл содержит одинаковые ключи или в командной строке заданы дополнительно отдельно от файла совпадающие ключи, то приоритет имеет самый последний обработанный ключ.

Возможные ошибочные ситуации при обработке ключа:

1) если файл не может быть прочитан, то ядро СУБД не запускается и выдается соответствующее диагностическое сообщение;
2) если строка в файле длиннее 4096 символов, то ядро СУБД не запускается и выдается соответствующее диагностическое сообщение;
3) если внутри файла встречается ключ /CF, то он игнорируется, ядро СУБД запускается с выдачей соответствующего предупреждения;
4) ключ не распознан (как в файле, так и в командной строке). Ядро СУБД запускается с распознанными ключами и выдается диагностическое сообщение длиной не более 256 символов со списком нераспознанных ключей. Если нераспознанных ключей на X больше, чем может уместиться в диагностическом сообщении, то добавляется текст «+X more». Если не умещается ни одного ключа, то выводится только часть первого ключа с символом ~ в конце;
5) ошибка при выделении памяти для загрузки файла.

С ошибками 1) и 2) выводится спецификация файла, вызвавшего ошибку (может быть выведено максимум 256 символов, остальные символы заменяются знаком ~).

Примеры

```
linter /CF=/home/user/start_lin60.txt
Содержимое файла start_lin60.txt:
/pool=20000 /nooutfile
/tcorrect
/procprint

linter /CF=/home/user/start_lin60.txt /tracelog
```

Ключи доступа к защищенной БД

/PASS=<пароль>

Задает пароль доступа к защищенной БД.

Примечание
В некоторых ОС для символа $ требуется экранирование при указании в командной строке.

Пример

```
linter /pass=$gost$123456
```

/SETPASS

Задает интерактивный ввод пароля доступа к защищенной БД. Пароль доступа надо будет ввести в предложенном диалоговом окне или в строке ввода.

Пример

```
linter /setpass
```

/PASSFILE=<спецификация файла>

Задает имя и путь к текстовому файлу, содержащему пароль доступа к защищенной БД.

Пример

```
linter /passfile=~/linter/pass.txt
```

Содержимое файла `pass.txt` – строка: `$gost$123456`

Ключи управления оперативной памятью

/POOL=<размер>

Задает размер пула памяти ядра СУБД в страницах по 4 Кбайт. В пуле размещаются все очереди ядра: очередь файлов, очередь таблиц, очередь столбцов и другие очереди.

По умолчанию установлен минимальный размер пула ядра: 20000 страниц.

Примечания

1. Если ядро СУБД не может зарезервировать память указанного размера, то оно не стартует и на консоль будет выдано сообщение типа «`ERROR: no memory for pool NNN`», где `NNN` – заданный в ключе размер пула в страницах. Дополнительно выводится всплывающее диалоговое окно с предложением изменить указанный параметр запуска ядра СУБД.
2. Значение ключа имеет больший приоритет по сравнению со значением параметра `AUTOCONFIG ON` (см. документ «Создание и конфигурирование базы данных», пункт «Конфигурирование БД»).

/SPOOL=<размер>

Задает размер пула памяти одного процесса сортировки в страницах. Размер страницы по умолчанию 65535 байт, но его можно изменить с помощью команды `ALTER DATABASE SET RECORD SIZE LIMIT` (см. документ «Справочник по SQL», пункт «Ограничение длины записи»). В пуле процесса сортировки хранятся промежуточные результаты сортировки выборок в случае отсутствующих индексов или сложных запросов.

По умолчанию установлен минимальный размер пула сортировки: 4000 страниц для максимально допустимой длины записей таблиц БД 4 Кбайт; 12800 страниц для максимально допустимой длины записей таблиц БД 64 Кбайт.

Примечания

1. Рекомендуемое соотношение между параметрами `/POOL` и `/SPOOL` - 4 к 1. Подробнее о значениях параметров, влияющих на эффективность работы ядра СУБД ЛИНТЕР, рассказано в документе «Архитектура СУБД», раздел «Распределение оперативной памяти».
2. Значение ключа имеет больший приоритет по сравнению со значением параметра `AUTOCONFIG ON` (см. документ «Создание и конфигурирование базы данных», пункт «Конфигурирование БД»).

`/PPOOL=<размер>`

Задает размер пула подсистемы фразового поиска СУБД в страницах по 4 Кбайт.

По умолчанию – `2000` страницы, минимум – `400` страницы. Максимальное значение – `524287` страниц.

`/PCONTCACHE=<размер>`

Задает размер кэша в страницах по 4 Кбайт для контейнера, используемого при создании/модификации фразового индекса (не влияет на скорость поисковых операций с использованием фразовых индексов).

По умолчанию `1000`.

Если задано `0`, используется значение по умолчанию.

`/PBVCACHE=<размер>`

Задает размер кэша в страницах по 4 Кбайт для бит-вектора, используемого при создании/модификации фразового индекса (не влияет на скорость поисковых операций с использованием фразовых индексов).

По умолчанию `1000`.

Если задано `0`, используется значение по умолчанию.

`/INMEMPOOL=<размер>`

Задает максимально допустимое количество страниц в пуле ядра СУБД ЛИНТЕР, выделяемых для размещения таблиц «в памяти».

Если этот ключ не задан, то для версий 6.0.20.3 и ниже использование таблиц «в памяти» запрещено.

Если этот ключ не задан, то для версий 6.0.20.4 и выше будет выполнена автоматическая конфигурация по следующему принципу: если не задано число страниц для таблиц «в памяти» (таблиц, столбцов, файлов для них), то в качестве значения для этих показателей берется 20% от общего для соответствующих элементов в их очереди (таблиц, столбцов, файлов, страниц). Если после вычитания числа элементов для таблиц «в памяти» (взятого по умолчанию или указанного явно) из общего числа элементов получается значение, меньшее минимума для числа соответствующих элементов (5 таблиц, 20 столбцов, 10 файлов, 50 страниц), то соответствующее число элементов для таблиц «в памяти» корректируется, чтобы этого избежать.

Примечания

1. Страницы для размещения таблиц «в памяти» выделяются из пула памяти ядра СУБД (см. описание ключа /POOL).
2. Если размер пула страниц минус число страниц «в памяти» меньше 20000 страниц, то размер пула страниц будет увеличен на требуемое значение с выдачей сообщения на консоль и в файл `linter.out`.

`/LOCK`

Задает блокирование выделенной для работы ядра СУБД оперативной памяти (под очереди системных объектов, пул страниц) на уровне ОС. В случае если память заблокировать невозможно, ядро СУБД не запускается. В этом режиме за счёт отсутствия вытеснения памяти ядра СУБД другими программами ускоряется его работа. Пользоваться этой возможностью нужно с осторожностью, чтобы не захватить практически всю физическую память и, таким образом, не повлиять отрицательно на общую производительность СУБД.

Ключи управления функционированием

`/NOTSP`

Запрещает запускать транслятор триггеров и хранимых процедур (предполагается, что работа с данными объектами БД не потребуется). Эта возможность позволяет использовать ЛИНТЕР в системах с ограниченными ресурсами.

`/NOSQL`

Запрещает запуск SQL-транслятора.

Эта возможность позволяет использовать ЛИНТЕР в системах с ограниченными ресурсами. При запуске с ключом `/NOSQL` ядро ЛИНТЕР может работать с уже скомпилированными приложениями, использующими только оттранслированные запросы.

`/KILL=<время>`

Задает временной промежуток (в секундах), через который проверяется «живучесть» программ-клиентов. Если через указанный промежуток времени обнаруживается, что какой-либо из клиентов закончил работу и не известил об этом ЛИНТЕР, то ядро автоматически освободит относящиеся к этому клиенту ресурсы.

По умолчанию 120 секунд, минимум 10 секунд.

`/WLHB`

Наличие этого ключа при запуске ядра СУБД ЛИНТЕР позволяет завершить сохранение системного журнала утилитой `lhb` при получении ядром команды останова.

Например, пусть ядро СУБД ЛИНТЕР запущено с ключом `/WLHB`. Если утилита архивирования баз данных (`lhb`) выполняет операцию сохранения БД в режиме `WAIT`, и в это время приходит команда на останов ядра, то ядро не завершит работу до тех пор:

1) пока утилита `lhb` не закончит сохранение системного журнала;
2) пока не наступит тайм-аут, определенный для утилиты `lhb` (то есть прошло более 10 секунд после предыдущей команды от `lhb`);

3) пока не прошло 3000 секунд с момента запроса на останов ядра.

Если ядро запущено без ключа /WLHB, то останов ядра произойдет сразу, и архив не будет сформирован полностью.

/DEFCOMM[={ACK|ALL}[, …]]

Примечание
Поддерживается со сборки 6.0.17.95.

Ключ используется для работы с системой резервирования (server) и системой архивирования БД (lhb).

Возможна ситуация, когда клиент уже получил ответ на операцию commit, но, поскольку lhb работает в асинхронном режиме, эти данные не были сохранены в архив. В этом случае при отказе ЛИНТЕР-сервера невозможно будет восстановить данные, которые уже зафиксированы в БД. Это замечание относится, как к lhb, так и к server.

Для исключения такой ситуации вводится особый режим отложенного commit. Он заключается в том, что ответ клиенту на commit посылается не сразу, а только после того, как данные будут отосланы lhb. Таким образом, гарантируется сохранение данных lhb до получения ответа на запрос изменения БД клиентом.

Такое поведение замедляет работу клиента, особенно в случае работы нескольких lhb одновременно. Поэтому в зависимости от требований к быстродействию и надежности приложения могут применяться следующие варианты отложенного commit:

- DEFCOMM: отсылка ответа клиенту осуществляется непосредственно после отсылки данных одному из lhb. Подтверждение приема данных lhb не осуществляется. Данный режим используется по умолчанию;
- DEFCOMM=ACK: отсылка ответа клиенту осуществляется по приему подтверждения, что данные приняты от одного из lhb;
- DEFCOMM=ALL: отсылка ответа клиенту осуществляется при отсылке данных всем lhb;
- DEFCOM=ACK,ALL: отсылка ответа осуществляется при получении подтверждения о сохранении данных от всех lhb.

Примечание
Данный режим распространяется только на работающие в wait-режиме lhb.

Режим отложенного ответа включается не сразу по запуску lhb, а только после того момента, когда будет скачана вся БД.

Режим отключается при отсоединении всех процессов lhb, работающих в wait-режиме.

В режиме TRUETYPECOMMIT OFF (см. документ «Создание и конфигурирование базы данных», пункт «Конфигурирование БД») ответ будет задержан только в случае сохранения системного журнала СУБД на диск. Если в результате запроса сохранение на диск журнала не производилось, то и задержки ответа не будет.

/IGNERROR

Заставляет ядро СУБД игнорировать ошибки восстановления по системному журналу БД.

По умолчанию ядро СУБД во время восстановления БД (например, после отказа компьютера или из-за ошибки в самом ядре СУБД) при обнаружении ошибки прекращает дальнейшее восстановление БД. При задании ключа процесс восстановления прерываться не будет. Это позволяет запустить потом утилиту `testdb` (см. документ «Тестирование базы данных», раздел «Выполнение программы»).

`/RAPID`

Задает режим «догона» системного журнала, развернутого из архивного файла БД. Используется при запуске СУБД ЛИНТЕР в системе резервирования.

`/TMPDIR=<строка>`

Определяет местоположение (каталог) для размещения временных файлов.

`/[NO]JEXIT`

Ключ `/JEXIT` задает режим автоматического завершения работы ядра СУБД ЛИНТЕР в случае невозможности продолжения записи в системный журнал. Если ядро запущено с ключом `/NOJEXIT`, то работа ядра будет продолжаться, но без ведения системного журнала.

Примечание
В системный журнал заносится информация обо всех изменениях в БД. В случае аварийного завершения работы СУБД ЛИНТЕР при следующем запуске ядро определит (по журналу) наличие прерванных транзакций и аннулирует все сделанные изменения. Если работа ведется без системного журнала, то СУБД ЛИНТЕР не сможет восстановить физическую и логическую целостность базы данных после аварии.

По умолчанию `/JEXIT`.

`/TCORRECT`

Заставляет ядро СУБД ЛИНТЕР игнорировать тот факт, что дата последнего запуска СУБД является «будущей» по сравнению с текущей датой ОС. Действует по умолчанию.

`/CHECK`

Задает режим периодической проверки живучести родительского процесса. Если родительский процесс завершился, ядро СУБД также завершает свою работу.

`/K=<значение>`

Заставляет посылать сигнал (`<значение>`) родительскому процессу при успешной инициализации ядра СУБД ЛИНТЕР.

`/NOLARGE`

При запуске ядра СУБД определяет, поддерживает ли операционная система длинные файлы (это файлы, размер которых больше 2 Гбайт). Если не поддерживает, то запрещается стартовать на БД, содержащей такие файлы. Для исключения возникновения подобной ситуации необходимо задавать параметр `/NOLARGE`, который запрещает создание таблиц размером больше 2 Гбайт.

Если ядро СУБД запускается без ключа /NOLARGE, а ОС не поддерживает длинные файлы, то ядро запустится, но не будет работать с таблицами, которые расположены в длинных файлах. При попытке обратиться к таким таблицам будет выдано соответствующее сообщение.

Если ядро СУБД запускается с ключом /NOLARGE, и в БД присутствуют файлы размером больше 2 Гбайт, то ядро не будет запущено. При попытке расширить файл до размеров больше 2 Гбайт будет возвращен код, сигнализирующий о невозможности это сделать.

Запуск с ключом /NOLARGE позволяет пользователю создавать БД и работать с ней какое-то время на ОС, поддерживающей длинные файлы, а затем перенести БД на ОС, не поддерживающую длинные файлы.

/RO

Задает работу с БД в режиме «только чтение» (модификация БД в этом режиме невозможна). При запуске СУБД с этим ключом доступны следующие команды:

SELECT	ALTER EVENT
EXECUTE PROCEDURE	SET EVENT
TEST TABLE	CLEAR EVENT
LOCK TABLE	SET LOG
UNLOCK TABLE	SET NAMES
CREATE EVENT	SET SESSION BLOB LOG
DROP EVENT	SET SESSION BLOB GEOMETRY STORAGE {OFF|ON}
GET EVENT	SET SESSION QUANTUM
WAIT EVENT	SET COMPATIBILITY

При работе в режиме read only в каталоге временных файлов создается ряд файлов (файлы логирования, трассировки, временные рабочие файлы): linter.out, LINTER.LOG, lintrace.log, phrase.idx, 1.31, 1.41, 1.51.

При завершении работы СУБД ЛИНТЕР перечисленные выше файлы удаляются.

Каталог временных файлов определяется следующим образом (по убыванию приоритета):

1) может быть задан в явном виде в параметрах запуска ядра с помощью ключа TMPDIR, например:

```
linter /base=/usr/linter/db /RO /TMPDIR=/tmp/linter_tmp
```

2) путь к каталогу временных файлов берется из переменной среды окружения TEMP.

/DEBUG

Запрещает процессу ядра СУБД переходить в фоновый режим.

/FIXCHAN[=<размер>]

Примечание
Поддерживается со сборки 6.0.17.92.

Если был задан этот ключ, то при очистке канала выделенная ему память не освобождается (как делается в обычном режиме работы), а впоследствии переиспользуется для вновь открытого канала с тем же номером. При указании параметра <размер> дополнительно при открытии канала с номером, который еще не использовался, для него резервируется указанный размер памяти в байтах. Максимальное значение параметра равно 65536. При указании ключа /FIXCHAN без параметра размера (или с параметром, равным 1) при открытии нового канала память изначально не резервируется.

/EVENTLIMIT=<размер очереди активных событий>

Примечание
Поддерживается со сборки 6.0.17.92.

Значение устанавливает максимальный размер очереди активных событий. При превышении значения будет выдан код завершения 90 «Нет памяти для размещения очереди сообщений». Значение по умолчанию и минимальное значение - 128, максимальное - 65535. При старте ядра на консоль и в linter.out будет выдано сообщение «Event queue size: NNN items».

/NOEXTFILE

Примечание
Поддерживается со сборки 6.0.17.92.

Запрещает все обращения к внешним файлам, то есть допустимо заносить имена файлов в столбцы типа EXTFILE, но нельзя выполнять любые операции, при которых формируются полные имена файлов для их поиска в файловой системе.

/RESTRICTEXTFILE

Примечание
Поддерживается со сборки 6.0.17.92.

Запрещает все обращения к внешним файлам, которые лежат вне каталога базы данных, то есть нельзя использовать абсолютные пути и относительные пути, содержащие ссылку на родительский каталог "..", в том числе в пути для файлов по умолчанию.

/SAFE_MODE

Примечание
Поддерживается со сборки 6.0.17.95.

Ядро СУБД запускается в безопасном режиме. В данном режиме запрещены команды: ALTER TABLE DROP COLUMN, PRESS TABLE, CORRECT, RENAME TABLE, RENAME COLUMN, RENAME INDEX, SET SESSION BLOB LOG OFF/ON, SQL BACKUP, TRUNCATE для циклических таблиц.

При подаче запрещенной команды в безопасном режиме будет выдан код завершения 97 «Операция запрещена в безопасном режиме».

/[NO]SYNC

При запуске ядра СУБД ЛИНТЕР с параметром /SYNC пул ядра начинает работать по механизму сквозной записи на диск. То есть результат операций сразу помещается на диск, минуя кэш.

Отключение данной опции осуществляется установкой параметра /NOSYNC. При этом сброс буферов происходит при завершении транзакции (применяется механизм отложенной записи на диск).

В синхронном режиме работа ядра СУБД ЛИНТЕР замедляется.

По умолчанию /NOSYNC.

Ключи сетевых настроек

/NAME=<строка>

Аналог переменной окружения LINTER_MBX.

/MBX=<параметр>

Аналог переменной окружения LINTER_MBX.

/JDBCP=<порт>

Задает запуск одновременно с запуском ядра СУБД сетевого драйвера сервера для JDBC-сервера с указанным номером <порта>.

/JDBCS

Задает запуск сетевого драйвера сервера для JDBC-сервера с номером порта по умолчанию 1070.

/TCP[=<порт>]

Задает режим запуска сетевого драйвера сервера, работающего по протоколу TCP/IP, одновременно с запуском ядра. <Порт> – номер порта, по которому будет осуществляться соединение клиента с сетевым сервером. Более подробно см. «Сетевые средства», раздел «Драйвер сервера», пункт «Запуск драйвера».

Значение по умолчанию 1060.

Ключи совместимости с SQL-стандартом и другими СУБД

```
/COMPATIBILITY=<опция>[,<опция> [, …]]
<опция>::=STANDARD
          |CASTNOLENCHECK
          |CASTNOLTRM
          |GEOPREFIX
          |PGGEO
```

```
|RESIGNAL_ALL
|ORACLE
|BROWSE_BLOB
|NOREC_EXCEPTION
|OPTIMISTIC
|NAMES_UPPERCASE
```

Опция STANDARD

При запуске с опцией STANDARD ядро СУБД ЛИНТЕР начинает придерживаться стандарта SQL92. Поведение ядра, запущенного без ключа с данной опцией, в некоторых случаях не совпадает со стандартом.

Отличия в работе ядра, запущенного с ключом /COMPATIBILITY=STANDARD:

1) позиционные DML-операции и работа транзакций.

   При запуске с опцией STANDARD команды UPDATE CURRENT и DELETE CURRENT могут выполняться по собственному каналу, а не по тому каналу, по которому был подан SELECT-запрос. Это важно для согласованной работы транзакций.

   При запуске без опции STANDARD команды UPDATE CURRENT и DELETE CURRENT выполняются по тому каналу, по которому был подан соответствующий SELECT;

2) очистка канала после завершения транзакции.

   При запуске с опцией STANDARD очистка канала по командам COMT/RBAC (завершение транзакции) закрывает выборку. Исключение составляет выполнение команды PUTM (в СУБД ЛИНТЕР можно подавать команду COMT внутри потока команд пакетного добавления без завершения транзакции), в этом случае выборка не будет закрыта.

   При запуске без опции STANDARD после команд COMT/ROLB выборка не будет закрыта, следовательно, можно подавать команды GET* по тому же каналу;

3) код завершения в случае обработки 0 записей.

   При запуске с опцией STANDARD всеми DML-операциями (SELECT, DELETE, UPDATE, INSERT FROM SELECT) в случае обработки 0 записей возвращается код завершения «Нет данных» (код 2).

   При запуске без опции STANDARD в случае обработки 0 записей операциями DELETE, UPDATE, INSERT FROM SELECT возвращается код успешного завершения (код 0), а код завершения «Нет данных» возвращается только операцией SELECT;

4) выполнение операций 'CAST <выражение> AS CHAR' в том случае, когда <выражение> имеет тип REAL или DOUBLE.

   При запуске с опцией STANDARD результат выполнения описанной операции выводится в экспоненциальной форме с усеченными нулями и без знака '+'.

   При запуске без опции STANDARD результат выполнения описанной операции всегда выводится в форме с десятичной точкой и без экспоненты (так же, как для значений DECIMAL);

5) выполнение операций 'CAST <выражение> AS CHAR', где <выражение> имеет тип DECIMAL.

При запуске с опцией STANDARD при преобразовании выражения типа DECIMAL с указанием точности в CHAR выводится столько символов после запятой, сколько указано точностью.

Без опции STANDARD при преобразовании DECIMAL в CHAR концевые нули всегда усекаются;

6) привилегии для выполнения операций DELETE/UPDATE.

При запуске с опцией STANDARD для выполнения операции DELETE/UPDATE над некоторой таблицей с отбором записей по WHERE необходима не только привилегия DELETE/UPDATE на эту таблицу, но и привилегия SELECT на нее.

При запуске без опции STANDARD для выполнения операции достаточно привилегии DELETE/UPDATE;

7) усечение лишних концевых пробелов в константах.

При запуске с опцией STANDARD перед проверкой совместимости типов происходит усечение лишних концевых пробелов в текстовых константах. Концевые пробелы в пределах длины типа не усекаются при операциях над значенями типов CHAR и NCHAR (как это делается в обычном режиме).

```
create or replace table tab2(c1 char(10), c2 char(10));
insert into tab2 values ('123', '456');
set compatibility 'STANDARD' on;
select c1 || c2 from tab2;
|123       456       |
set compatibility 'STANDARD' off;
select c1 || c2 from tab2;
|123456              |
```

При запуске без опции STANDARD перед проверкой совместимости типов усечение лишних концевых пробелов не происходит, например, не удастся занести в CHAR(5) константу "A " ("буква и 5 пробелов");

8) обязательность условия для JOIN.

При запуске с опцией STANDARD, если не задано условие соединения (ON или USING) для конструкции JOIN без конструкций NATURAL и UNION, выдается ошибка.

При запуске без опции STANDARD условие соединения воспринимается, как если бы соединяемые по JOIN таблицы были перечислены во FROM через запятую;

9) ESC-символ по умолчанию.

При запуске с опцией STANDARD ESC-символа по умолчанию нет.

При запуске без опции STANDARD ESC-символом по умолчанию является символ '\'.

10) сравнение NULL-значения <значимого выражения условия> в опции CASE с NULL-значением <значимого выражения условия> в опции WHEN.

При запуске с опцией STANDARD результат - FALSE.

При запуске без опции STANDARD результат - TRUE.

11) удаление таблицы (синонима), на которую ссылается представление.

    Запрещено обычное удаление таблицы, на которую ссылаются представления (`VIEW`). При этом на консоль ядра СУБД и в файл `linter.out` выдаётся список объектов, ссылающихся на удаляемый объект и не позволяющих его удалить.

12) доступ личной последовательности.

    При запуске с опцией `STANDARD` только создателю последовательности, при запуске без опции `STANDARD` - всем пользователям.

13) тип данных результата функция `SUM`.

    При запуске с опцией `STANDARD` функция `SUM` с аргументом одного из типов `SMALLINT`, `INT`, `BIGINT` возвращает результат типа `BIGINT`, при работе в обычном режиме она в этих же случаях возвращает результат типа `DECIMAL`.

Опция `CASTNOLENCHECK`

Задаёт подавление вывода кода завершения 1063 «`Попытка усечения непустых символов`» при преобразовании числового значения в строковое с явным указанием длины результата в конструкции выполнении `CAST`.

По умолчанию данный код завершения выдается.

Опция `CASTNOLTRM`

Запрещает усечение ведущих пробелов при преобразовании чисел с фиксированной точкой в строковое значение в конструкции `CAST`.

При задании ключа с данной опцией ведущие пробелы в 32-символьном представлении DECIMAL-значения не усекаются, по умолчанию – усекаются.

Опция `GEOPREFIX`

Задаёт распознавание ключевых слов подсистемы геометрических данных только при наличии префикса `LIN_` (например, `LIN_ASTEXT` вместо `ASTEXT`). По умолчанию воспринимаются ключевые слова без префикса `LIN_`, хотя некоторые из них распознаются только в определенном контексте (например, `ASTEXT`, если дальше следует признак функции (открывающая скобка "(")).

Опция `PGGEO`

В данном режиме координаты могут быть разделены пробелами и запятыми, в стандартном режиме они могут быть разделены только пробелами. При этом для каждой точки допустимо указание до 4-х координат из многомерного пространства, реально используются из которых только первые две.

Опция `RESIGNAL_ALL`

Заставляет некритичные исключения обрабатывать как критичные (при таком запуске ядра СУБД нет необходимости в каждом вложенном блоке явно выполнять оператор `RESIGNAL`).

Если опция `RESIGNAL_ALL` задана, то обработка некритичных исключения сразу передается в текущий <блок обработки исключений>, при его отсутствии – исключение будет передано на верхний уровень по иерархии.

Подробнее см. документ «Процедурный язык», раздел «Формат блока кода».

Опция ORACLE

Заставляет обеспечивать идентичность обработки данных СУБД ЛИНТЕР и СУБД ORACLE по умолчанию в случаях, когда их обработка различается.

При указании данной опции:

- разность дат возвращается в виде значения типа NUMERIC (в днях), а не значения типа DATE;
- выдается информация о том, что в SELECT-запросе делается попытка вызова хранимой процедуры, содержащей вызовы запросов, отличных от SELECT (выполнение которых может выдавать различные результаты при последовательных вызовах с одними и теми же аргументами). На консоль ядра СУБД и в файл linter.out будут выдаваться сообщения следующего вида:

```
WARNING: user function "имя_процедуры" (#ROWID) possibly contains
 non-SELECT query calls
```

- результатом конкатенации символьных значений с NULL-значением будет исходная символьная строка (а не NULL-значение).
- Наличие псевдостолбца с именем «LEVEL» в выборке вида SELECT * FROM... зависит от ключа /COMPATIBILITY=ORACLE в команде запуска ядра СУБД:
  1) псевдостолбец будет присутствовать, если ключ не задан;
  2) псевдостолбец будет отсутствовать, если ключ задан.

Опция BROWSE_BLOB

Блокирует текущую запись, если в ней есть BLOB-значение по умолчанию так, как если бы был задан модификатор for browse, то есть изменяет поведение при обнаружении конфликта доступа к BLOB-значению. В случае отсутствия ключа будет выдано диагностическое сообщение «Строка при работе с BLOB не заблокирована», а в случае его наличия будет выполнена блокировка записи.

Опция NOREC_EXCEPTION

При выполнении оператора FETCH процедурного языка при отсутствии обработанных строк в курсоре будет сгенерировано исключение 2 (NOREC).

Опция OPTIMISTIC

Разрешает использование режима OPTMISTIC.

Опция NAMES_UPPERCASE

Приводит все идентификаторы, в том числе указанные в двойных кавычках, к верхнему регистру.



Примечание
Поддерживается со сборки 6.0.17.97.

Ключи оптимизации обработки SQL-запросов

/ANALYZE

Оптимизатор СУБД ЛИНТЕР будет анализировать все SQL-запросы с точки зрения существования индексов, позволяющих ускорить их обработку. Список используемых и

рекомендуемых к созданию индексов будет выведен на экран консоли ядра и в файл `linter.out` (описание файла приведено в приложении 2). Допустимо указывать один из ключей анализа SQL-запросов.

`/ANALYZE_NEED`

Оптимизатор СУБД ЛИНТЕР будет анализировать все SQL-запросы с точки зрения существования индексов, позволяющих ускорить их обработку. Список рекомендуемых к созданию индексов будет выведен на экран консоли ядра и в файл `linter.out` (описание файла приведено в приложении 2). Допустимо указывать один из ключей анализа SQL-запросов.

Примечание
Поддерживается со сборки 6.0.20.2.

`/ANALYZE_USED`

Оптимизатор СУБД ЛИНТЕР будет анализировать все SQL-запросы с точки зрения существования индексов, позволяющих ускорить их обработку. Список используемых индексов будет выведен на экран консоли ядра и в файл `linter.out` (описание файла приведено в приложении 2). Допустимо указывать один из ключей анализа SQL-запросов.

Примечание
Поддерживается со сборки 6.0.20.2.

`/AUTOINDEX`

Оптимизатор СУБД ЛИНТЕР будет анализировать все SQL-запросы с точки зрения существования индексов, позволяющих ускорить их обработку и создавать необходимые индексы для ускорения обработки SQL-запросов.

Ключи протоколирования работы ядра СУБД

`/NOOUTFILE`

Запрещает печать сообщений в файл `linter.out` (протокол работы СУБД).

По умолчанию файл `linter.out` создается.

Примечание
Структура файла `linter.out` описана в приложении 2.

`/NOOUTPUT`

Запрещает вывод в консольное окно информации о работе ядра СУБД.

`/OUTFLIMIT=<размер>`

Задает размер файла сообщений ядра СУБД (`linter.out`) в страницах по 4 Кбайт. По умолчанию размер файла не ограничен.

Если в процессе работы размер файла linter.out превысит заданный размер, то он будет переименован в файл linter.ou1. После этого информация будет записываться в пустой файл linter.out. То есть в процессе работы СУБД могут существовать только два файла – linter.out и linter.ou1.

Примечание
Структура файла linter.out описана в приложении 2.

/LOGFLIMIT=<размер>

/TRACEFLIMIT=<размер>

Задают допустимые размеры файлов LINTER.LOG и lintrace.log соответственно. <Размер> задается в блоках по 4 Кб. Задание размеров файлов приводит к тому, что перед проверкой ограничений пользователей осуществляется проверка размера файлов LINTER.LOG и lintrace.log. Если их размер превышает заданные границы, вся информация из текущих файлов LINTER.LOG и lintrace.log копируется в файлы с маской LINTER_YYYYMMDDHH24MISS.LOG и lintrace_YYYYMMDDHH24MISS.log, где YYYY – год, MM – месяц, DD – день, HH24 – час, MI – минуты, SS – секунды.

Вслед за этим открываются пустые файлы с именами LINTER.LOG и lintrace.log, и работа продолжается в обычном режиме вплоть до очередной проверки.

Примечания
1. Указанные ключами максимальные размеры файлов LINTER.LOG и lintrace.log могут быть несколько превышены, так как проверки выполняются через определенные промежутки времени.
2. Если ОС не поддерживает размер файлов более 2 Гб, то максимальный размер файла LINTER.LOG, если он не задан, автоматически устанавливается равным 1 Гб.

/[NO]LOG

Ключ /NOLOG – запрещает, ключ /LOG – разрешает вести протокол обработки SQL-запросов клиентских приложений (файл LINTER.LOG).

Если при запуске ядра ЛИНТЕР задан ключ /LOG, то ведется «краткий» протокол обработки запросов: текст SQL-запроса, количество ответов, код возврата.

По умолчанию /NOLOG.

При перезапуске ядра с включенным протоколированием обработки SQL-запросов старый файл будет переименован в файл с маской LINTER_YYYYMMDDHH24MISS.LOG и будет создан новый файл LINTER.LOG.

/LOGQUERY

Задает режим протоколирования обработки SQL-запросов клиентских приложений в файле LINTER.LOG.

При перезапуске ядра с включенным протоколированием обработки SQL-запросов старый файл будет переименован в файл с маской LINTER_YYYYMMDDHH24MISS.LOG и будет создан новый файл LINTER.LOG.

Примечание
Структура файла LINTER.LOG описана в приложении 3.

/LOGALL

Задает режим полного протоколирования обработки SQL-запросов клиентских приложений в файле LINTER.LOG. В этом режиме в файл протокола заносится дополнительная информация: время выполнения команды, сетевой адрес, идентификаторы процесса и нити, пославшие запрос, и другая информация.

При перезапуске ядра с включенным протоколированием обработки SQL-запросов старый файл будет переименован в файл с маской LINTER_YYYYMMDDHH24MISS.LOG и будет создан новый файл LINTER.LOG.

```
/TRACE=DECOMP{[=FULL]|=DELAY[=<тики>]|=SPPAG}
| CHTRAN[=NOFLUSH]
| KRB
| LOCK[=LEVEL={0|1|2|3}|[TIME]]
|           LOGIO[=([DEF|COMMT|ABSADR|PREPADR|HEX|BLOCK|REC|STRUCT|
DATA=<значение>|LEVEL=<значение>][, …])]
| KANCHN
| SORT[=TIME]
| WRBL[=READ]
```

Задает различные режимы трассировки обрабатываемых SQL-запросов в трассировочный файл lintrace.log (размещается в каталоге БД).

Конструкция DECOMP задает режим трассировки общей информации об обрабатываемом запросе. По умолчанию выполняется краткая трассировка.

Для получения более полной информации надо указывать опцию FULL.

Трассировка содержит следующую информацию:

- текст SQL-запроса, переданный на обработку ядру СУБД ЛИНТЕР (после его оптимизации SQL-транслятором);
- какие массивы данных (бит-вектора) были задействованы при обработке SQL-запроса. Используемые массивы влияют на время выполнения запроса;
- количество считанных/записанных блоков данных (физических/логических);
- количество считанных/записанных блоков системного журнала.

Пример трассировочной информации

```
C#4 QUERY:
  SELECT
    T_0."MSG"
  FROM
    <TABLE "SYSTEM"."ERRORS" AS T_0>
  WHERE
    T_0."NMRERR" == 1503;
C#4 DECOMP.C (Start_Cur_Dec): Now computing derived set #0.
```

```
C#4 OBRSTRAT.C (OBRSTRAT): Start set: TABLE("SYSTEM"."ERRORS" AS
 T_0). Set included 1022 rows.
    List of predicates:
    Predicate [strategy #2(one index)]:
  T_0."NMRERR" == 1503
C#4 DECOMP.C (End_Dekart): Derived set #0 is computed, Rows count:
 1.
C#4 FORMOTW.C (FORMOTW): Read: 0 blocks, write: 0 blocks.
C#4 FORMOTW.C (FORMOTW): Read logical: 4 blocks, write logical: 0
 blocks.
C#4 FORMOTW.C (FORMOTW): Journal read: 0 blocks, written: 0
 blocks.
```

Конструкция DECOMP=DELAY=<тики> задает режим трассировки задержек длительности выполнения отдельных квантов ядра СУБД на время, не меньшее указанного в параметре <тики>. По умолчанию трассируются задержки не менее 100 тиков (0.1 секунды).

Пример трассировочной информации

```
C#3 OBRSTRAT.C (OBRSTRAT): Start set: TABLE("SYSTEM"."LINEITEM" AS
 T_1). Set included 6001215 rows.
    List of predicates:
    Predicate [strategy #1(full scan)]:
  T_1."L_SHIPDATE" > '15.03.2013:00:00:00:00'
C#3 PROZA.C (PROZA): Strategy: #1(full scan).
C#3 Delay for 1.87 sec
C#3 Delay for 0.35 sec
C#3 RIDSTRAT.C (RIDSTRAT): Snap_Bv: 3241776 rows.
```

Конструкция CHTRAN задает режим трассировки информации об обрабатываемых транзакциях. В этом режиме выполняется трассирование операций OPEN/OCUR/CLOS/KILL/CKILL/COMT/RBAC интерфейса нижнего уровня, а также SQL-запросов COMMIT и ROLLBACK.

Для повышения производительности СУБД предусмотрена опция NOFLUSH. При ее задании трассировочная информация предварительно накапливается в оперативной памяти и записывается в файл lintrace.log в соответствии с настройками ОС (по мере вытеснения буфера файла на диск).

Пример трассировочной информации

```
C#5 OPEN: E=0: M=MVCC_OO
C#6 OCUR: E=0: M=MVCC_RC|MVCC_AUTOCOMMIT: EX=5
C#6 CLOS: E=0
C#5 COMT: E=0: M=MVCC_OO: S=JOURNAL_OPER_EXIST
C#5 TRID: E=0: T=23.09.2014 15:13:51.44: ID=410: M=MVCC_OO:
 S=Open_exist|Tr_exist|WRT_OPTIMISTIC_EXIST
C#5 COMT: E=0: T=23.09.2014 15:13:51.44: ID=410: M=MVCC_OO:
 S=Open_exist|JOURNAL_OPER_EXIST
C#5 CLOS: E=0
```

Конструкция KRB задает режим трассировки информации, формируемой Kerberos-сервером.

Пример трассировочной информации

Данный пример приведен для имени сервиса linter@srv.example.net и имени аутентифицирующегося пользователя test. Имя библиотеки (в примере libgssapi_krb5.so) зависит от реализации Kerberos и может отличаться.

```
Use krb library libgssapi_krb5.so
load symbols from library libgssapi_krb5.so
Use service name linter@srv.example.net
gss_import_name, status MJ=0 MI=0
gss_acquire_cred status MJ=0 MI=0
gss_accept_sec_context status MJ=0 MI=0
gss_display_name status MJ=0 MI=0
KRB user name test
```

На клиенте был заранее получен тикет для пользователя test.

Mj и Mi – результат выполнения функции.

Конструкция LOCK добавляет вывод информации о заблокированных таблицах и запросах, вызвавших конфликты.

Опция LEVEL задает дополнительный уровень выдаваемой информации о блокировках. Поддерживаются следующие уровни выдаваемой информации о блокировках:

- LEVEL=0 – выдает в lintrace.log информацию об ошибках;
- LEVEL=1 – дополнительно выдает в lintrace.log информацию о блокировках таблиц (в том числе и тех, которые не вызывают конфликты блокировок);
- LEVEL=2 – дополнительно выдает в lintrace.log информацию о блокировках групп записей;
- LEVEL=3 – дополнительно выдает в lintrace.log информацию о блокировках отдельных записей.

Опция TIME выводит время для каждого сообщения (по умолчанию не выводится).

Пример трассировочной информации

```
TRACELOCK: PAUSE CHANNEL 6 by channel 8
Locked table: "T2"

C#6 QUERY:.
  UPDATE.
    <TABLE "SYSTEM"."T2" AS T_0>
  SET.
    T_0."I" = {(T_0."I")(1) <+> };
TRACELOCK: Found DEADLOCK. Channel 6 tries to pause channel 8.
Table to be locked: "T1"

C#8 QUERY:.
```

```
  UPDATE.
    <TABLE "SYSTEM"."T1" AS T_0>
  SET.
    T_0."I" = {(T_0."I")(1) <+> };
TRACELOCK: UNLOCK CHANNEL 6 paused by channel 8
```

Конструкция LOGIO (опции DEF, COMMT, ABSADR, PREPADR, HEX, BLOCK, REC, STRUCT, DATA, LEVEL) задает режим трассировки записи/чтения системного журнала.

Опция DEF задает установки по умолчанию (действует также, если задано /TRACE=LOGIO без опций).

Опция COMMT по умолчанию отключена. Устаревшая опция, в текущей версии не используется.

Опции ABSADR и PREPADR по умолчанию включены. Задают вывод журнальных адресов в дополнение к трассировке.

Опция HEX по умолчанию отключена. Задает вывод журнального адреса в шестнадцатеричном виде, иначе в десятичном.

Опция BLOCK по умолчанию включена. Задает вывод информации об операциях с журнальными блоками.

Опция REC по умолчанию включена. Задает вывод информации на уровне записей в журнале.

Опция STRUCT по умолчанию включена. Задает вывод информации о структуре журнальных записей.

Опция DATA по умолчанию имеет значение 24. Задает вывод буферов данных (выводится не более указанного числа байт).

Опция LEVEL по умолчанию имеет значение 3. Задает начальный уровень трассировки, который влияет на вывод или подавление вывода той или иной журнальной информации, этот уровень увеличивается и уменьшается в процессе работы.

Конструкция KANCHN задает режим трассировки изменения количества активных каналов.

Конструкция SORT задает режим трассировки обмена с процессами сортировки, опция TIME выводит время для каждого обмена (по умолчанию не выводится).

Конструкция WRBL задает режим трассировки записи на диск/чтения с диска страниц пула, страниц ядра.

Опция READ задает режим трассировки чтения страниц (по умолчанию только запись страниц).

Опция SPPAG предназначена для трассировки использования страниц файла 1.01. Это файл индексов таблицы $$$SYSRL, но в нем также хранятся и разные страницы специальных типов. Типы страниц этого файла такие: страницы индексов, страницы ограничений целостности (integrity), страницы CHECK, страницы диапазонов значений AUTOINC, страницы длинных VIEW, страницы описания составных ключей, страницы расширенной информации о файлах. В трассировке отображаются: выделение страницы с определенным номером в файле, занятие страницы с определенным номером под

определенный тип страницы, освобождение страницы с определенным номером из-под определенного типа страницы. Трассировка предназначена для поиска проблем, связанных со страницами файла 1.01.

/TRACELOG[=1]

Задает вывод в файл linter.out диагностических сообщений об открытии и закрытии соединения с БД (с полной информацией об источнике команды) и ошибок выполнения запросов по соединениям.

При задании параметра 1 информация о соединении выводится с каждым ошибочным запросом.

/PROCPRINT

Разрешает хранимым процедурам выводить:

- на консоль ядра СУБД ЛИНТЕР и в файл протоколирования linter.out сообщения процедурной функции PRINT. Максимальный размер выводимого сообщения 980 символов.

  Сообщение выводится в виде:

  «*** Message from Stored Procedure: <текст сообщения>»

  По умолчанию выполнение функции PRINT игнорируется.

Примечание
Вывод сообщений в файл протоколирования linter.out поддерживается со сборки 6.0.17.95.

- на консоль ядра СУБД ЛИНТЕР и в файл протоколирования linter.out информацию об исключениях. Если хранимая процедура оттранслирована с отладкой, то выдается номер ошибочной строки в процедуре.

  Примеры вывода.

```
Exception DIVZERO caught in procedure "TEST" line 2, processed
Exception 903 caught in procedure "AAA" line 8, processed
Exception CUSTOM (1) caught in procedure "TEST", processed
Exception 905 caught in procedure "Trigger #      47#" line 3,
 processed
Exception BADPARAM caught in procedure "TEST3" line 2, resignaled
Exception DIVZERO caught in procedure "TEST", ignored
```

  Не протоколируются те исключения, которые игнорируются автоматически (без явной конструкции IGNORE).

- Функционал аналогичен подаче SQL-команды SET PROCEDURE TRACE ON;, но не требуется перезапуск ядра.

/PIDFILE=<файл>

Задает имя файла, в который будет записан pid ядра СУБД. В случае корректного завершения работы СУБД этот файл удаляется.

Ключи LDAP-аутентификации

`/LDAPSUFFIX=<строка>`

Строка, добавляемая после имени пользователя, чтобы сформировать уникальное имя (`DN`) в БД LDAP-сервера.

Аналог переменной окружения LDAP_SUFFIX с более высоким приоритетом.

`/LDAPPREFIX=<строка>`

Строка, добавляемая перед именем пользователя, чтобы сформировать уникальное имя (`DN`) в БД LDAP-сервера.

Аналог переменной окружения LDAP_PREFIX с более высоким приоритетом.

`/LDAPTOUT=<целочисленное значение>`

Тайм-аут в секундах.

Аналог переменной окружения LDAP_TIMEOUT с более высоким приоритетом.

`/LDAPSRV=<адрес>`

Адрес LDAP-сервера в виде доменного имени.

Аналог переменной окружения LDAP_SERVER с более высоким приоритетом.

`/LDAPBASEDN=<строка>`

Корень поиска для LDAP-аутентификации с предварительным поиском.

Аналог переменной окружения LDAP_BASEDN с более высоким приоритетом.

`/LDAPFLTR=<строка>`

Фильтр поиска пользователя в БД LDAP.

Аналог переменной окружения LDAP_FILTER с более высоким приоритетом.

`/LDAPSRCHDN=<dn>`

Уникальное имя пользователя (`DN`), от которого будет производиться предварительный поиск в БД LDAP-сервера.

Аналог переменной окружения LDAP_SEARCHDN.

`/LDAPSRCHPW=<пароль>`

Пароль пользователя, от имени которого будет выполняться предварительный поиск в БД LDAP-сервера.

Аналог переменной окружения LDAP_SEARCHPW с более высоким приоритетом.

`/LDAPSRCHPWFILE=<спецификация файла>`

Путь к файлу, содержащему пароль `DN` для поиска в БД LDAP-сервера.

Аналог переменной окружения LDAP_SEARCHPW_FILE с более высоким приоритетом.

Ключи KERBEROS-аутентификации

/KRBSRVC=<строка>

Задает имя ЛИНТЕР-сервиса для Kerberos-сервера. Аналог переменной окружения LINTER_KRB_SERVICE, но с более высоким приоритетом.

Ключи проверки соответствия мандатного доступа

/CHECK_LEVELS

При задании ключа /CHECK_LEVELS выполняются дополнительные проверки соответствия мандатного доступа СУБД мандатному доступу операционной системы. При работе ядра, запущенного с указанным ключом выполняются дополнительные проверки при подключении пользователя к СУБД:

1) если пользователь СУБД не имеет уровней мандатного доступа, а пользователь с таким же именем есть в операционной системе и имеет отличный от 0 уровень, то подключение не проходит;
2) если пользователь СУБД имеет уровни мандатного доступа, и пользователя с таким же именем нет в операционной системе, то подключение не проходит;
3) если пользователь СУБД имеет уровни мандатного доступа, а пользователь с таким же именем есть в операционной системе и имеет отличный от RAL пользователя СУБД уровень, то подключение не проходит.
4) Для ЗОСРВ Нейтрино проверяется уровень доверия пользователя. Поддерживается в версиях 6.0.20.1 и выше.
5) Для ОС Astra Linux проверяются и RAL и WAL пользователя БД и они должны попадать в диапазон между минимальным и максимальным уровнями пользователя ОС. Поддерживается в версиях с 6.0.20.1 по 6.0.20.5.

При неуспешном подключении в журнал linter.out добавляется соответствующая запись с информацией о событии.

Информационные ключи

/BRIEFVERSION

Задает вывод в консольное окно краткой информации о версии ядра СУБД (если ядро запускается как приложение ОС).

/VERSION

Задает вывод в консольное окно подробной информации о версии ядра СУБД (если ядро запускается как приложение ОС).

Переменные среды окружения

В своей работе ядро и утилиты СУБД ЛИНТЕР используют переменные среды окружения, которые могут быть установлены средствами ОС до запуска ЛИНТЕР.

Для установки значения переменной среды окружения для сеанса консольного файлового менеджера необходимо выполнить в нем команду:

```
export <имя переменной>=<значение переменной>
```

Например:

```
export SY00=/linter/db
```

При этом для применения значений переменных для утилит СУБД они должны запускаться в командном режиме в текущем сеансе консольного файлового менеджера.

Общие переменные

В процессе работы СУБД ЛИНТЕР может использовать следующие переменные среды окружения:

1) **SY00** – значение переменной определяет путь до каталога основных файлов базы данных (файлов системных таблиц). По умолчанию в качестве значения переменной используется текущий каталог ОС. Этот путь может быть задан также из командной строки при запуске ядра;
2) **LINTER_EDIT** – переменная, определяющая местоположение редактора SQL-скриптов, используемого утилитой `inl`;

Примечание
Используется утилитами СУБД ЛИНТЕР.

3) **LINTER_MBX** – указывает идентификатор «почтового ящика» для обмена ядра с клиентскими приложениями. Под «почтовым ящиком» следует понимать межпроцессный механизм обмена между ядром ЛИНТЕР и приложением. Изменение значения этой переменной может быть использовано для запуска нескольких ядер СУБД ЛИНТЕР на одном компьютере.

   Значение этой переменной должно быть: для ОС Linux числовым в диапазоне от 1 до 65535 (значение по умолчанию 20561); для ЗОСРВ Нейтрино строковым (значение по умолчанию "`linter`").

   Оно не должно совпадать ни с одним из значений для существующих почтовых ящиков или других механизмов обмена в ОС – разделяемой памяти и семафора;

Примечание
Если для ОС Linux задано нечисловое значение, то используется значение 0.

Пример:

а) в `nodetab` добавить строку:

```
DemoDb LOCAL 1234
```

б) в файловом менеджере, в котором будем запускать утилиты, выполнить команду:

```
export LINTER_MBX=1234
```

в) выполнить запуск ядра СУБД и сетевого драйвера клиента в файловом менеджере:

```
/usr/linter/bin>linter /base=/linter/db
/usr/linter/bin>dbc_tcp
```

г) запустить новую сессию файлового менеджера, определить переменную LINTER_CP и запустить утилиту командного интерфейса с подключением к БД DemoDb:

```
export LINTER_MBX=1234
inl -u SYSTEM/MANAGER8 -n DemoDb
```

Успешный переход в интерактивный режим выполнения команд:

```
SQL>
```

является подтверждением успешного подключения к ядру СУБД;

д) завершить утилиту командного интерфейса командой `exit;`

е) завершить работу сетевого клиента: в сессии командного менеджера, в которой он запущен, нажать сочетание клавиш <Ctrl>+<C>;

ж) остановить ядро СУБД с помощью команды:

```
shut -u SYSTEM/MANAGER8
```

4) **LINTER_CP** – определяет кодировку данного клиентского приложения. Ядро СУБД ЛИНТЕР будет работать с этим приложением в кодировке, определяемой переменной LINTER_CP. Для разных приложений, даже на одной машине, могут быть заданы различные значения переменной LINTER_CP.

   СУБД ЛИНТЕР поддерживает однобайтовые, многобайтовые кодировки, UNICODE, UTF8, а также предоставляет возможность загрузить в БД кодировку, необходимую клиентской задаче. Переменная LINTER_CP может принимать значение любой кодировки, загруженной в БД (список кодировок хранится в системной таблице $$$CHARSET).

   Если значение переменной LINTER_CP не определено, то используется кодировка, соответствующая текущему значению locale.

   Для консольных утилит СУБД ЛИНТЕР используется кодировка по умолчанию 866, для графических утилит – 1251.

5) **NET_MBX** – переменная аналогична по своему назначению переменной LINTER_MBX, но предназначена для сетевого клиента. То есть NET_MBX определяет номер «почтового ящика» для обмена данными между приложением и сетевым драйвером клиента. Значение по умолчанию: 20562 для ОС Linux, linter_net для ЗОСРВ Нейтрино.

Примечание
Если для ОС Linux задано нечисловое значение или 0, то используется значение по умолчанию.

6) **LINTER_TMP** – определяет путь к каталогу временных файлов;

Переменные идентификации и аутентификации по LDAP-протоколу

Возможны два режима LDAP-аутентификации:

1) без предварительного поиска;

2) с предварительным поиском.

Для успешной аутентификации парольная аутентификация по LDAP-протоколу через LDAP-сервер требуется наличие пользователя в БД ЛИНТЕР и в БД LDAP-сервера, при этом имя пользователя БД ЛИНТЕР должно соответствовать одному из атрибутов пользователя в БД LDAP-сервера (для аутентификации без предварительного поиска данный атрибут должен входить в состав уникального имени пользователя в БД LDAP-сервера). Кроме того, чтобы для режима LDAP-аутентификации с предварительным поиском не требовалось задавать DN и пароль пользователя с правами поиска, необходимо разрешение на анонимный поиск в LDAP-базе.

В режиме без предварительного поиска сначала идет подключение к серверу LDAP_SERVER и попытка выполнить соединение с уникальным именем, сформированным как LDAP_PREFIX<имя>LDAP_SUFFIX, где <имя> и пароль определяются соответствующими параметрами команды OPEN (см. документ «Интерфейс нижнего уровня», пункт «Открыть канал (соединение)»). Удачное соединение означает удачную идентификацию и аутентификацию.

В режиме с предварительным поиском сначала идет подключение к серверу LDAP_SERVER и попытка выполнить соединение со специальными именем LDAP_SEARCHDN и паролем, который может быть задан одним из способов:

- напрямую через переменную LDAP_SEARCHPW;
- с помощью ключа LDAPSRCHPW;
- через файл, полный путь к которому передается в параметре LDAP_SEARCHPW_FILE;
- анонимно, если эти учетные данные не заданы.

После этого производится поиск пользователя в БД LDAP-сервера в каталоге LDAP_BASEDN с фильтром LDAP_FILTER. Найденное в результате поиска уникальное имя (если единственное) используется для попытки соединения с введенным пользователем паролем. Удачное соединение означает удачную идентификацию и аутентификацию.

Для настройки LDAP-аутентификации ядра СУБД ЛИНТЕР используются следующие переменные окружения:

- **LDAP_PREFIX** определяет строку, добавляемую в качестве префикса к имени пользователя БД ЛИНТЕР для формирования уникального имени (DN) для LDAP-аутентификации. Переменная служит для указания имени атрибута, соответствующего выбранной схеме, отличной от значения по умолчанию. Если переменная LDAP_PREFIX не задана, по умолчанию используется строка "cn=". Например, для пользователя SYSTEM БД ЛИНТЕР будет использован атрибут со значением cn=SYSTEM (если LDAP_PREFIX ="cn=", то есть стандартное имя пользователя). Если LDAP_PREFIX ="sn=", то имя пользователя ЛИНТЕР будет соответствовать атрибуту Surname LDAP-схемы.

  Переменная игнорируется в режиме LDAP-аутентификации с предварительным поиском.

  Значение переменной может задаваться с помощью ключа /LDAPPREFIX.
- **LDAP_SUFFIX** определяет строку, добавляемую в качестве суффикса к имени пользователя БД ЛИНТЕР для формирования уникального имени (DN) для LDAP-аутентификации. LDAP_SUFFIX выбирается при настройке LDAP-сервера и обычно

соответствует доменному имени организации. Значение строки должно состоять из предопределенных имен атрибутов схемы LDAP-сервера (`ou` – организационная единица, `c` – страна, `dc` – домен и т.п.) и их значений. Указание переменной `LDAP_SUFFIX` является обязательным для режима LDAP-аутентификации без предварительного поиска. Если эта переменная не задана, то любой запрос на аутентификацию пользователей с включенной LDAP-аутентификацией будет отвергнут.

Например,

```
LDAP_SUFFIX="ou=Users, dc=company, dc=com"
```

DN строка формируется как конкатенация строк `LDAP_PREFIX`, имени пользователя СУБД ЛИНТЕР, символа ',' и `LDAP_SUFFIX`.

Переменная игнорируется в режиме LDAP-аутентификации с предварительным поиском.

Значение переменной может задаваться с помощью ключа `/LDAPSUFFIX`.

- **LDAP_SERVER** задает адрес LDAP-сервера в виде доменного имени. Если переменная не задана, по умолчанию предполагается работа с локальным LDAP-сервером (`localhost`). В переменной `LDAP_SERVER` может быть указан номер порта, отделяемый от доменного имени двоеточием. Можно задавать несколько серверов через пробел, в этом случае попытка соединиться с очередным LDAP-сервером будет предприниматься после неудачного соединения с предшествующим LDAP-сервером.

  Например,

```
LDAP_SERVER="ldap.server.domain.org"
LDAP_SERVER="ldap.server.domain.org:636"
LDAP_SERVER="ldap.server.relex.org"
LDAP_SERVER="ldap1.server.domain.org ldap2.server.domain.org:636
ldap3.server.domain.org"
```

  Значение переменной может задаваться с помощью ключа `/LDAPSRV`.

- **LDAP_TIMEOUT** задает интервал в секундах, в течение которого ожидается ответ от LDAP-сервера. Если в течение указанного интервала ответ не поступил, возвращается код завершения «`Тайм-аут LDAP-сервера`». По умолчанию интервал тайм-аута принимается в 10 секунд.

  Ожидание ответа от LDAP-сервера выполняется в чередующихся фазах активного и пассивного тайм-аута:

Фаза активного тайм-аута	Фаза пассивного тайм-аута	…	Фаза активного тайм-аута	Фаза пассивного тайм-аута

  В фазе активного тайм-аута ЛИНТЕР-сервер ожидает ответ от LDAP-сервера в течение указанного в LDAP_TIMEOUT интервала времени. Если ответ не поступил, генерируется код завершения «`Тайм-аут LDAP-сервера`».

  Если пользователь продолжает попытки соединиться с ЛИНТЕР-сервером, то в течение следующей фазы пассивного тайм-аута (также равного значению LDAP_TIMEOUT) все попытки соединения с LDAP-сервером немедленно отвергаются с выдачей кода завершения «Тайм-аут LDAP-сервера» (хотя к этому моменту соединение с LDAP-сервером, возможно, уже восстановлено).

После пассивной фазы тайм-аута повторяется фаза активного тайм-аута и т.д. Это сделано с целью предотвращения накопления очереди пользователей, ожидающих аутентификации, при сбоях в работе LDAP-сервера.

Пример

```
LDAP_TIMEOUT=15
```

Значение переменной может задаваться с помощью ключа /LDAPTOUT.

- **LDAP_BASEDN** задает режим LDAP-аутентификации с предварительным поиском уникального имени пользователя в БД LDAP-сервера. Значением переменной является корень поиска, производимого через заданный фильтр.

  Задание переменной обязательно для LDAP-аутентификации с предварительным поиском.

  Значение переменной может задаваться с помощью ключа /LDAPBASEDN.

- **LDAP_FILTER** задает фильтр, используемый для поиска пользователя в БД LDAP-сервера. Строка фильтра может содержать служебное слово '%username%', которое при поиске заменяется на введенное при установлении соединения с СУБД ЛИНТЕР имя пользователя БД ЛИНТЕР. В найденной записи извлекается уникальное имя пользователя (DN), которое используется совместно с введенным пользователем паролем для попытки аутентификации. Если фильтр находит несколько записей (либо не находится ни одной), то аутентификация считается не пройденной.

  Переменная игнорируется в режиме LDAP-аутентификации без предварительным поиска.

  Значение по умолчанию "`(uid=%username%)`".

  Значение переменной может задаваться с помощью ключа /LDAPFLTR.

- **LDAP_SEARCHDN** задает уникальное имя (DN), с помощью которого будет производиться поиск в БД LDAP-сервера.

  Переменная игнорируется в режиме LDAP-аутентификации без предварительным поиска. Если значение переменной не задано или задана пустая строка, то будет использоваться анонимный поиск (без предоставления регистрационных данных пользователя БД LDAP-сервера).

  Значение переменной может задаваться с помощью ключа /LDAPSRCHDN.

- **LDAP_SEARCHPW** задает пароль для имени, от которого будет производиться поиск в БД LDAP-сервера.

  Переменная игнорируется в режиме LDAP-аутентификации без предварительным поиска или в случае анонимного поиска.

  Значение переменной может задаваться с помощью ключа /LDAPSRCHPW.

- **LDAP_SEARCHPW_FILE** задает путь к файлу, содержащему пароль LDAP_SEARCHPW. Переменная имеет более высокий приоритет, чем переменная LDAP_SEARCHPW и ключ /LDAPSRCHPW.

  Значение переменной может задаваться с помощью ключа /LDAPSRCHPWFILE.

Переменные идентификации и аутентификации по Kerberos-протоколу

Для идентификации и аутентификации по Kerberos-протоколу используются следующие переменные окружения:

- **LINTER_KRB_SERVICE** определяет имя службы ЛИНТЕР-сервера. Kerberos-сервер должен быть настроен на использование данной службы.

  Например,

```
LINTER_KRB_SERVICE="linter/linterserver.relex.ru"
LINTER_KRB_SERVICE="linter@linterserver.relex.ru"
```

- **KRB5_KTNAME** задает местоположение (полный путь и имя) файла с таблицей ключей. Может использоваться некоторыми реализациями Kerberos.

Примечание
Переменная окружения **KRB5_KTNAME** используется ядром СУБД ЛИНТЕР не напрямую, а опосредованно, через библиотеку Kerberos.

Значения размеров очередей, задаваемые по умолчанию

По умолчанию при запуске ядра СУБД ЛИНТЕР устанавливаются следующие значения:

- длина или размер очереди, содержащей описания таблиц – 100;
- длина или размер очереди, содержащей описания столбцов – 500;
- длина или размер очереди файлов – 30;
- количество каналов для связи приложений с ядром СУБД ЛИНТЕР – 100;
- квант обработки записей – число записей, просматриваемых системой без прерывания при обработке одного запроса – 10;
- квант обработки индексов – число индексов, просматриваемых без прерывания при обработке одного запроса (1 индекс).

Останов СУБД ЛИНТЕР

Пользовательский останов ядра СУБД ЛИНТЕР

Пользовательский останов ядра СУБД осуществляется специальной утилитой `shut`, которая подает команду останова ядру СУБД ЛИНТЕР и ожидает, пока оно завершится.

Синтаксис команды

```
shut [<командная строка>]

<командная строка>::=[<имя>] [<пароль>] [<ЛИНТЕР-сервер>]
```

или

```
<командная строка>::={[-u имя/пароль] [-n <ЛИНТЕР-сервер>]
                      [-r] [-ci <кодировка>] | -version | -h}
```

Реально завершение работы ядра СУБД ЛИНТЕР произойдет примерно через 100 секунд после начала выполнения утилиты shut.

Примечание

Если в процессе работы выполнялось расширение системных файлов (`SYSWRK`, `SYSSRT`, `SYSWBV`), то перед завершением работы ядро СУБД усекает их до размера, указанного при создании БД (или при её конфигурировании) и выдает об этом на консоль ядра СУБД и в файл `linter.out` информационное сообщение вида:

```
_Attention: file SYSWRK truncated from 16 to 4 pages

(Внимание: файл SYSWRK был усечен с 16 страниц до 4)
```

В связи с этим для исключения затрат СУБД на операции расширения рабочих файлов рекомендуется выполнить переконфигурирование БД, указав в качестве начальных размеров рабочих файлов те размеры, до которых они были автоматически расширены (см. документ «Создание и конфигурирование базы данных», пункт «Конфигурирование БД»).

Ключи управления утилитой

-u

Задает регистрационные данные (имя и пароль) создателя БД или пользователя БД с привилегией `DBA`.

-n

Задает имя узла удаленного ЛИНТЕР-сервера (должно присутствовать в файле `nodetab`). Если этот параметр не задан, то команда применяется к локальному узлу или к узлу по умолчанию.

-r

Заставляет выполнить завершение работы ядра СУБД, независимо от наличия активных транзакций в данный момент. В случае наличия таких транзакций, произойдет их откат, а пользователи будут извещены о принудительном останове ядра СУБД. По команде останова ядра СУБД без ключа `-r` при наличии активных транзакций ядро СУБД не будет остановлено, и утилита `shut` получит код возврата 1012.

-ci <кодировка>

Задает кодовую страницу для интерфейса утилиты.

Если ключ не задан, по умолчанию используется язык операционной системы.

Если кодовая страница задана неверно или не установлена в ОС, используется англоязычный интерфейс.

Примеры:

```
-ci cp866 (русскоязычный интерфейс)
```

```
-ci cp437 (англоязычный интерфейс)
```

-version

Выдает информацию о версии программы `shut`.

-h

Выдается справочная информация о программе.

Системный останов ядра СУБД ЛИНТЕР

Останов ядра СУБД возможно выполнить сигналом `SIGTERM`. В этом случае останов ядра СУБД будет выполняться аналогично останову ядра при подаче команды `shut -r`, то есть выполняется принудительный останов с откатом всех незаконченных транзакций.

Синхронизация завершения работы СУБД

Для синхронизации завершения работы СУБД ЛИНТЕР используется программа `lsyncd`. Данная программа передает управление ОС после полного завершения работы ядра СУБД. Обычно она применяется совместно с программой `shut` в скриптах на закрытие ОС.

Запуск программы:

```
lsyncd [-t <тайм-аут>]
```

`<Тайм-аут>` – тайм-аут (в секундах) ожидания завершения работы ядра СУБД ЛИНТЕР. Если параметр не задан, по умолчанию используется значение 30 секунд.

Программа `lsyncd` предназначена для определения момента фактического завершения работы ядра СУБД ЛИНТЕР. При подаче команды `shut` с использованием программы `shut` или из прикладной программы пользователем, имеющим привилегии на выполнение этой операции, происходит только инициирование начала процесса завершения работы. Ответ ядра на эту команду немедленно передается пославшей ее программе, и она имеет возможность сообщить об успешном начале процесса завершения работы ядра СУБД. Однако сам процесс останова СУБД занимает определенное время. Для определения реального момента завершения работы ядра СУБД ЛИНТЕР служит программа `lsyncd`.

После запуска данной программы она переходит в состояние ожидания (об этом свидетельствует выводимая ею строка `Wait for LINTER shutdown`) и находится в этом состоянии до тех пор, пока ядро СУБД ЛИНТЕР существует как задача ОС. По окончании процесса завершения работы ядра СУБД ЛИНТЕР программа выводит на экран строку complete и завершает свою работу.

Если при запуске `lsyncd` ядро СУБД ЛИНТЕР не активно, программа завершает свою работу немедленно с выдачей соответствующего сообщения.

Так же, как и для обычных клиентов, для перенаправления слежения за другим ядром можно использовать переменную окружения LINTER_MBX. Ее значение должно совпадать со значением этой переменной при запуске ядра СУБД ЛИНТЕР. Программа `lsyncd` может применяться только на том компьютере, на котором запущено выполнение ядра СУБД ЛИНТЕР. Работа по сети невозможна.

Программа `lsyncd` должна выполняться пользователем ОС, который запустил ядро СУБД ЛИНТЕР, или суперпользователем ОС.

Для ipc-версии СУБД ЛИНТЕР проверка активности ядра производится каждую секунду.

Для socket-версии завершение работы программы происходит немедленно после завершения работы ядра СУБД ЛИНТЕР.

Если по истечении тайм-аута ядро СУБД ЛИНТЕР еще активно, происходит окончание работы `lsyncd` с кодом завершения -1. Если же ядро СУБД ЛИНТЕР не загружено или в процессе работы `lsyncd` уже завершило свою работу, код завершения будет равен 0.

Коды завершения

Коды завершения ядра СУБД

Если во время запуска ядра СУБД ЛИНТЕР происходит сбой, то анализируется причина сбоя, и работа ядра завершается. При этом выдается соответствующий код завершения.

Код завершения 1

Диагностическое сообщение

```
Wrong database version. Check version for database and kernel.
```

Причина ошибки

Версия запускаемого ядра СУБД не соответствует версии БД, на которой его пытаются запустить.

Рекомендации по устранению

Варианты:

1) выполнить запуск ядра СУБД с БД соответствующей версии.
2) выполнить перенос БД на новую версию СУБД ЛИНТЕР с помощью утилиты `migration` или `datariv`.

Для определения версии СУБД ЛИНТЕР и БД выполнить команды:

```
linter /version
gendb -get "DATABASE VERSION"
```

См. документы:

- «Миграция базы данных»;
- «Конвертер баз данных»;
- «Создание и конфигурирование базы данных».

Код завершения 2

Диагностическое сообщение

```
Database is corrupt. Please repair.
```

Причина ошибки

Повреждение физической структуры БД.

Ответственный за устранение

Администратор БД.

Рекомендации по устранению

Выполнить рекомендованные действия в соответствии с документами:

- «Тестирование базы данных»;

- «Архивирование и восстановление базы данных».

Код завершения 3

Диагностическое сообщение

```
Not enough memory. Change kernel startup parameters.
```

Причина ошибки

Для запуска ядра СУБД ЛИНТЕР недостаточно доступной оперативной памяти.

Рекомендации по устранению

Варианты:

1) уменьшить размер пула ядра СУБД и/или уменьшить размер пула сортировки (см. описание ключей /POOL и /SPOOL в этом документе);
2) увеличить размер доступной ядру СУБД оперативной памяти либо за счет аппаратного увеличения оперативной памяти компьютера, либо уменьшить количество запущенных на компьютере программ.

Код завершения 4

Диагностическое сообщение

```
Low pool size. Change kernel startup parameters.
```

Причина ошибки

Заданный при запуске ядра СУБД размер пула ядра не достаточен для его работы.

Рекомендации по устранению

Увеличить размер пула ядра СУБД (см. описание ключа /POOL в этом документе).

Код завершения 5

Диагностическое сообщение

```
Can't initialize kernel. Syscall failed.
```

Причина ошибки

Возможные причины:

1) для БД, запущенной в режиме только для чтения (ключ /RO), длина пути к каталогу временных файлов превышает допустимую длину.

   Путь к каталогу временных файлов задается либо в ключе /DIR, либо в переменной среды окружения LINTER_TMP.

   Максимальная длина пути к каталогу задается в макросе ОС PATH_MAX (значение по умолчанию 32);

2) ошибка инициализации или установки дескриптора безопасности, создания мэйлслота, нити, объекта файлового отображения;

3) ошибка инициализации подсистемы репликации СУБД ЛИНТЕР.

Рекомендации по устранению

В первом случае создать каталог для временных файлов БД с допустимым по длине именем.

Во втором и третьем случае следует установить причину ошибки, изучив диагностические сообщения в файле `linter.out` каталога базы либо системный код ошибки из сообщения об ошибке (отображаемое либо в диалоговом окне, либо в событии в журнале событий). В дальнейшем действовать исходя из полученной информации об ошибке.

Код завершения 6

Диагностическое сообщение

```
Linter already running. Change kernel startup parameters.
```

Причина ошибки

Ядро СУБД на указанной БД уже запущено.

Рекомендации по устранению

Чтобы запустить несколько ядер СУБД ЛИНТЕР на одном компьютере, необходимо использовать ключ `/NAME` и переменную окружения `LINTER_MBX`.

См. документ:

- «Сетевые средства».

Код завершения 7

Диагностическое сообщение

```
Can't access database. Check database file presence and permissions.
```

Причина ошибки

Возможные причины:

1) местоположение БД неизвестно;

2) БД отсутствует в указанном каталоге;

3) у пользователя нет прав для запуска ядра СУБД на указанной БД.

Рекомендации по устранению

В первом случае указать местоположение БД с помощью ключа `/BASE` или переменной среды окружения `SY00`.

Во втором случае указать правильный каталог БД.

В третьем случае предоставить пользователю права DBA или предъявить пароль доступа к защищенной БД.

См. документ:

- «Справочник по SQL».

Код завершения 8

Диагностическое сообщение

```
Kernel abandon due to critical data write to disk fail.
```

Причина ошибки

Невозможно продолжать запись в системный журнал (ядро СУБД ЛИНТЕР запущено с ключом /JEXIT) из-за отсутствия свободного места на диске или исчерпания файлов системного журнала.

Рекомендации по устранению

1) Обеспечить наличие свободного пространства на диске.
2) Увеличить количество файлов системного журнала и/или их размер.

Количество файлов системного журнала и размер каждого из них задаются в свойствах БД с помощью утилиты linadm или gendb (параметры SYSLOG COUNT, SYSLOG SIZE).

См. документы:

- «Создание и конфигурирование базы данных».

Код завершения 9

Диагностическое сообщение

```
Conflict between system time and time of last db shutting.
```

Причина ошибки

Дата последнего запуска ядра СУБД (по временному поясу GMT) является «будущей» по сравнению с текущей датой ОС.

Рекомендации по устранению

Запускать ядро СУБД ЛИНТЕР с ключом /TCORRECT.

Код завершения 10

Диагностическое сообщение

```
Can't find temporary directory or environment variable 'TEMP'
is not found.
```

Причина ошибки

Возможные причины:

1) для БД, запущенной в режиме «только чтение» (ключ `/RO`), невозможно определить временный каталог;
2) задано неверное значение переменной среды окружения `TEMP`.

Рекомендации по устранению

Указать правильный каталог для размещения временных файлов БД с помощью ключа запуска `/TMPDIR` или в переменной среды окружения `TEMP`.

Код завершения 11

Диагностическое сообщение

```
Database files locked by another application. May be Linter
already running on this database.
```

Причина ошибки

Ядро СУБД не может заблокировать индексный файл системной таблицы `$$$SYSREL`. Указанная БД уже заблокирована другим приложением (возможно, на ней уже запущено ядро СУБД).

Рекомендации по устранению

Завершить работу приложения, блокирующего файлы БД, на которой запускается ядро СУБД, и повторить запуск ядра СУБД.

Код завершения 12

Диагностическое сообщение

```
Wrong username/password specified.
```

Причина ошибки

Неверное имя пользователя или пароль в ключе `/U` запуска ядра СУБД.

Рекомендации по устранению

Ввести правильные регистрационные данные пользователя, от имени которого выполнятся запуск ядра СУБД.

Список пользователей БД можно получить с помощью SQL-запроса:

```
select $$$s34 from $$$usr;
```

(для этого надо запустить ядро СУБД от имени пользователя с правильными регистрационными данными).

Код завершения 13

Диагностическое сообщение

```
Kernel arguments error.
```

Причина ошибки

Ошибка работы с файлом, содержащим ключи командной строки запуска ядра СУБД ЛИНТЕР (ключ `/CF`).

Возможные причины:

1) неверная спецификация (файл не найден);
2) длина записей файла больше 4096 байт;
3) значение ключа `/NAME` или `/MBX` превышает 32 символа;
4) значение ключа `/TMPDIR` превышает значение, установленное в макросе `PATH_MAX` в ОС (по умолчанию 32).

Рекомендации по устранению

Исправить файл с ключами командной строки запуска ядра СУБД ЛИНТЕР (см. ключ `/CF` в этом документе) и повторить запуск ядра СУБД.

Код завершения 15

Диагностическое сообщение

```
License check failed.
```

Причина ошибки

Истек срок использования демонстрационной версии СУБД ЛИНТЕР.

Рекомендации по устранению

Приобрести лицензионную версию СУБД ЛИНТЕР.

Коды завершения программы останова СУБД

В таблице приведен перечень кодов завершения программы останова СУБД.

Таблица. Перечень кодов завершения программы останова СУБД

Код завершения	Описание
0	Успешное завершение.
1	Неполные или неверные аргументы.
2	Неверные имя пользователя/пароль или нет привилегий для завершения работы ядра.
3	Ядро СУБД или сетевой драйвер не загружены, не найден указанный сервер.
4	Неизвестная ошибка при обращении к СУБД ЛИНТЕР.
5	Ядро СУБД выполняет другие запросы.

Приложение 1
Пример настройки СУБД ЛИНТЕР для идентификации и аутентификации по Kerberos-протоколу

Исходные данные:

- клиентские приложения – на компьютерах под управлением ОС Linux или ОС Windows;
- Kerberos-сервер установлен и сконфигурирован на компьютере с адресом `kdc.domain1.example.com` в ОС Linux;
- Linux-клиент и ядро СУБД ЛИНТЕР размещены на компьютере с ОС Linux и сетевым адресом `test.domain2.example.com`;
- Windows-клиент размещен на компьютере с ОС Windows и сетевым адресом `win7.domain1.example.com`;
- на компьютере с СУБД ЛИНТЕР (сетевой адрес `test.domain2.example.com`) средствами ОС созданы пользователи test и user;
- адреса всех компьютеров должны быть разрешимы через `DNS`.

Примечание
Для примера выбрано имя сервиса `server/test.domain2.example.com` (в общем случае имя сервиса состоит из следующих частей: произвольное имя, "/" и имя компьютера в сети, на котором запущен данный сервис).

Настройка компьютера с Kerberos-сервером (`kdc.domain1.example.com`):

1) Установить пакеты `krb5-admin-server` и `krb5-kdc`, указав в качестве запрашиваемых регистрационных данных следующие значения:

```
Default realm: DOMAIN1.EXAMPLE.COM
Kerberos server: kdc.domain1.example.com
Admin server: kdc.domain1.example.com
```

2) Запустить `krb5_newrealm` от имени суперпользователя.

3) Для поддержки Windows-клиента в файле `kdc.conf` (путь по умолчанию `/etc/krb5kdc/kdc.conf`) задать поддерживаемые типы кодирования:

```
supported_enctypes =
rc4-hmac:normal rc4-hmac-exp:normal des-cbc-crc:normal
```

4) Для корректного распознавания доменов в `realm` добавить в раздел `[domain_realm]` файла `/etc/krb5.conf`, в разделе `[domain_realm]` строки:

```
.domain1.example.com = DOMAIN1.EXAMPLE.COM
domain1.example.com = DOMAIN1.EXAMPLE.COM
.domain2.example.com = DOMAIN1.EXAMPLE.COM
domain2.example.com = DOMAIN1.EXAMPLE.COM
```

5) С помощью утилиты `kadmin.local` добавить пользователей в БД Kerberos-сервера:

```
sudo kadmin.local
> addprinc test
> addprinc user
```

Примечание
Запрашиваемые при создании пользователей пароли могут быть произвольными.

6) С помощью утилиты `kadmin.local` создать сервис со случайным паролем и файл ключа к нему, чтобы этот сервис мог аутентифицироваться по Kerberos-протоколу автоматически:

```
sudo kadmin.local
>addprinc -randkey server/test.domain2.example.com
>ktadd -k /tmp/server.keytab server/test.domain2.example.com
```

7) Скопировать файл `/tmp/server.keytab` на компьютер, где будет располагаться ядро СУБД ЛИНТЕР (`test.domain2.example.com`).

8) Для подключения Windows-клиента добавить его компьютер в БД Kerberos-сервера:

```
sudo kadmin.local
> addprinc -pw <password> host/win7.domain1.example.com
```

где:

`<password>` – произвольный пароль, который надо будет указывать в дальнейшем при аутентификации Windows-клиента;

`host` – ключевое слово для идентификации сетевого имени компьютера.

Настройка компьютера с ядром СУБД ЛИНТЕР (`test.domain2.example.com`):

1) Установить пакеты `krb5-config` и `krb5-user`.

2) Скопировать файл `/etc/krb5.conf` с компьютера с Kerberos-сервером в каталог `/etc`.

3) Скопировать ключ, созданный на Kerberos-сервере (см. **Настройка компьютера с Kerberos-сервером**, **действие** 6), либо в стандартный файл `/etc/krb5.keytab`, либо в любой другой, но тогда указать его местоположение в переменной окружения KRB5_KTNAME.

4) Указать имя ЛИНТЕР-сервиса в переменной окружения:

```
LINTER_KRB_SERVICE="server/test.domain2.example.com"
```

5) Запустить ядро СУБД ЛИНТЕР.

6) Создать в БД ЛИНТЕР пользователя с режимом идентификации и аутентификации по Kerberos-протоколу и тем же именем, что и в БД Kerberos-сервера, например, с помощью утилиты `inl`:

```
inl -u SYSTEM/MANAGER8
>CREATE USER "test" IDENTIFIED BY KRB;
>CREATE USER "user" IDENTIFIED BY KRB;
```

Настройка компьютера с Linux-клиентом (`test.domain2.example.com`):

1) Скопировать файл `/etc/krb5.conf` с компьютера с Kerberos-сервером в каталог `/etc`.

2) С помощью команды kinit получить Kerberos-тикет

```
kinit
```

при этом на запрашиваемый пароль необходимо ввести пароль соответствующего Kerberos-пользователя, который был создан на этапе настройки Kerberos-сервера (см. **Настройка компьютера с Kerberos-сервером**, **действие** 5).

3) Клиентское Linux-приложение готово к идентификации и аутентификации по Kerberos-протоколу. Для доступа к СУБД ЛИНТЕР по Kerberos-протоколу необходимо указать пустые имя и пароль пользователя. Доступ к СУБД ЛИНТЕР будет предоставлен тому пользователю, который указан в полученном от Kerberos-сервера Kerberos-тикете.

Настройка компьютера с Windows-клиентом (`win7.domain1.example.com`):

1) Подключиться к `Kerberos-realm` с помощью команд:

```
ksetup /setdomain DOMAIN1.EXAMPLE.COM
ksetup /addkdc DOMAIN1.EXAMPLE.COM kdc.domain1.example.com
```

   Для вступления изменений в силу перезагрузить компьютер.

2) Сделать привязку пользователей ОС Windows к пользователям БД Kerberos-сервера с помощью утилиты `ksetup` и опции `/mapuser`. Например, если имена пользователей в ОС Windows совпадают с именами пользователей в БД Kerberos-сервера, то для привязки друг к другу всех пользователей ОС Windows и БД Kerberos-сервера с одинаковыми именами выполнить:

```
ksetup /mapuser * *
```

3) Скопировать файл `krb5.conf` с Kerberos-сервера в файл конфигурации Kerberos на Windows. В роли файла конфигурации Kerberos-сервера в среде ОС Windows может выступать как файл `krb5.conf`, так и файл `krb5.ini`, в зависимости от версии ОС Windows и Kerberos (также могут варьироваться каталоги по умолчанию местонахождения этих файлов). Для подробностей необходимо обращаться к соответствующей документации либо явно указать их местоположение в переменной окружения `KRB5_CONFIG`.

4) создать Kerberos-пароль компьютеру с клиентским приложением с помощью команды:

```
ksetup /setmachpassword <password>
```

   Пароль <password> должен совпадать с тем, что был указан на этапе настройки Kerberos-сервера (см. **Настройка компьютера с Kerberos-сервером**, **действие** 8).

   Для вступления изменений в силу требуется перезагрузка компьютера.

5) После выполнения всех перечисленных пунктов пользователь может заходить в систему, используя имя и пароль Kerberos-аккаунта. При этом в качестве домена необходимо выбрать `Kerberos-realm` (`DOMAIN1.EXAMPLE.COM`).

6) Клиентские Windows-приложения готовы к идентификации и аутентификации по Kerberos-протоколу. Для доступа к СУБД ЛИНТЕР по Kerberos-протоколу необходимо ввести имя и пароль пользователя из БД Kerberos-сервера, в качестве ЛИНТЕР-сервера указать `Kerberos-realm` (`DOMAIN1.EXAMPLE.COM`).

   Доступ к СУБД ЛИНТЕР будет предоставлен тому пользователю, который вошел в ОС Windows.

Приложение 2
Описание файла linter.out

В приложении приводится пример информации, содержащейся в файле сообщений ядра СУБД ЛИНТЕР (`linter.out`).

В этот файл заносится информация о таких параметрах запуска ядра, как размер очередей (`Table`, `Column`, `Channel`, `File`, `User`), размер пула ядра (`POOL`), количество процессов сортировок; а также значение переменной `LINTER_MBX`, состояние системных таблиц БД (например: `Devices`, `Charset`, `Security`, `Procedures`). Сюда же вносятся предупреждения и сообщения об ошибках, возникающих в процессе работы СУБД.

Файл сообщений является дополняемым - при последующих запусках СУБД сообщения будут дописываться в конец файла `linter.out`.

При старте ядра СУБД ЛИНТЕР на уже созданной базе данных в файл `linter.out` заносятся сообщения о дате создания БД и датах последнего запуска и останова базы данных:

```
Database creation time:      02.12.2025 08:37:22.48
Last database startup time:  09.12.2025 06:25:34.49
Last database shutdown time: 09.12.2025 06:25:43.14
```

При завершении работы СУБД в файл `linter.out` заносится сообщение:

```
*** RDBMS Linter has been shut down ***
```

Пример файла `linter.out`.

```
Database creation time: 02.12.2025 08:37:22.48
Last database startup time:  09.12.2025 06:25:34.49
Last database shutdown time: 09.12.2025 06:25:43.14
09.12.2025 06:26:32.042 Linter SQL Bastion v. 6.0.20.6 connected
 to data base "LINTER Database  "
09.12.2025 06:26:32.042 64-bit build
09.12.2025 06:26:32.042  SY00 is "/usr/linter/db/"
09.12.2025 06:26:32.042  Database meets safety certification
 requirements
09.12.2025 06:26:32.042  Superuser mode is off
09.12.2025 06:26:32.042  Memory clearing is on
09.12.2025 06:26:32.042  POOL holds 100000 pages (96738 free
 pages, 19347 pages for in-memory tables)
09.12.2025 06:26:32.042  Table    queue size  : 165 (33)
09.12.2025 06:26:32.042  Column   queue size  : 994 (198)
09.12.2025 06:26:32.042  Channel  queue size  : 1000
09.12.2025 06:26:32.042  File     queue size  : 330 (66)
09.12.2025 06:26:32.042  User     queue size  : 100
09.12.2025 06:26:32.046  Checking stored procedures
09.12.2025 06:26:32.046  Auditing disabled
09.12.2025 06:26:32.047  Transaction control is turned on
09.12.2025 06:26:32.047  Max concurrent sorting processes: 1
 (sorting pool holds 25008 blocks, 16 blocks per page)
```

```
09.12.2025 06:26:32.047  Record size limit: 65535 bytes
09.12.2025 06:26:32.047  Channel memory limit: 256 pages
09.12.2025 06:26:32.047  Phrase index pool size: 2000 pages
09.12.2025 06:26:32.047  Event queue size: 128 items
 Existing control point(s) in database:
  Control point list is empty.
09.12.2025 06:26:32.048  Kernel system parameters:  MBX - "20561",
 Pid - 769688
09.12.2025 06:26:32.052 INFO: JDBC listener started at 1070 port
09.12.2025 06:26:32.052 INFO: TCP listener started at 1060 port
09.12.2025 06:26:32.052 Copyright (C) 1990-2025 Relex, Inc. All
 rights reserved.
09.12.2025 06:26:32.052  *** RDBMS Linter is running
09.12.2025 06:26:32.052  *** Press <ENTER> for shell prompt
09.12.2025 06:26:32.052 Linter kernel command line: /usr/linter/
bin/linter /BASE=/usr/linter/db /POOL=100000 /SPOOL=25000 /
TCP=1060 /JDBCS /JDBCP=1070
09.12.2025 06:26:32.052
09.12.2025 06:26:38.509  INFO: command 'SHUT', user 'SYSTEM', net
 protocol='', pid=769702, tid=769702.
09.12.2025 06:26:38.509  *** Linter is coming down ***
09.12.2025 06:26:38.509  *** Answer cache processing ***
09.12.2025 06:26:38.509      Free 0 answer elements
09.12.2025 06:26:38.509  *** Table queue processing ***
09.12.2025 06:26:38.509  *** Column queue processing ***
09.12.2025 06:26:38.510  *** User queue processing ***
09.12.2025 06:26:38.510  *** File queue processing ***
09.12.2025 06:26:38.619  *** RDBMS Linter has been shut down ***
```

Приложение 3
Описание файла LINTER.LOG

В приложении рассматриваются примеры информации, которая заносится в файл протоколирования обработки SQL-запросов пользователя (файл LINTER.LOG), и приводится расшифровка данной информации.

Файл LINTER.LOG является перезаписываемым - при запуске ядра (с параметром, разрешающим ведение протокола) все сообщения о предыдущем сеансе работы СУБД будут удалены из файла.

Пример 1. Пример файла LINTER.LOG в режиме краткого протоколирования.

```
?DESC:L=92:
!:E=0:C=0:
?OPEN:U="SYSTE<":P=16384:R=0:
!:E=1025 @&#:C=0:
?OPEN:U=H:P=16384:R=0:
!:E=1025 @&#:C=0:
?OPEN:U="SYSTEM":P=16384:R=0:
!:E=0:C=3:
?DESC:L=92:
!:E=0:C=3:
?DESC:L=92:
!:E=0:C=3:
?OCUR:C=3:P=16384:R=0:
!:E=0:C=4:
?SLCT:C=4:L=65535:P=0:
select * from "SYSTEM"."$$$AUDIT";;
!:E=0:C=4:A=1:
?GETA:C=4:L=0:
!:E=0:C=4:
?KILL:C=3:U=H:I=4:
!:E=0:C=3:
?CLOS:C=4:
!:E=1069 @&#:C=4:
OCUR:C=3:P=16384:R=0:
!:E=0:C=4:
?:C=4:L=65535:P=0:
execute "SYSTEM"."SAMPLE"(FALSE,FALSE,1);
#?OCUR:C=4:P=34:R=0:
#!:E=0:C=5:
#?:C=5:L=0:P=0:
drop table results;
#!:E=0:C=5:
#?:C=5:L=0:P=0:
create table results(lin char(100));
```

```
#!:E=0:C=5:
```

Пример 2. Пример файла LINTER.LOG в режиме полного протоколирования.

```
?SLCT:T=10:26:25.370:XPid=1732:XTid=1632:C=5:L=65535:P=0:
select * from auto;;;
!:E=0:T=10:26:25.450:XPid=1732:XTid=1632:C=5:A=1000:
?GETA:T=10:26:25.450:XPid=1732:XTid=1632:C=5:L=0:
!:E=0:T=10:26:25.450:XPid=1732:XTid=1632:C=5:
?GETA:T=10:26:25.450:XPid=1732:XTid=1632:C=5:L=3914:
!:E=0:T=10:26:25.460:XPid=1732:XTid=1632:C=5:
?OCUR:T=10:26:25.460:XPid=1732:XTid=1632:C=3:P=16384:R=0:
!:E=0:T=10:26:25.460:XPid=1732:XTid=1632:C=6:
?GETS:T=10:26:25.480:XPid=1732:XTid=1632:C=5:I=1:L=113:
!:E=0:T=10:26:25.480:XPid=1732:XTid=1632:C=5:
?GETS:T=10:26:25.480:XPid=1732:XTid=1632:C=5:I=1:L=113:
!:E=0:T=10:26:25.480:XPid=1732:XTid=1632:C=5:
```

Знак **?**, расположенный в начале строки, показывает, что информация относится к обработке запроса. Затем идёт команда из блока CBL интерфейса нижнего уровня (например, OCUR, SLCT, DESC). Если ведется полное протоколирование работы, то будет показана также дополнительная информация: время выполнения запроса, сетевой адрес, идентификаторы процесса и нити, пославшие запрос. Далее идут параметры команд из блока CBL, которые описаны ниже. Затем (на следующей строке) расположены данные, которые необходимы ядру СУБД для выполнения команды (если они требуются): текст запроса, данные для загрузки в BLOB и другая информация.

Знак **!**, расположенный в начале строки, указывает на то, что это информация об ответе ядра на запрос пользователя. Она включает в себя код возврата (:E=«код возврата») и признак ошибки (символы @&#, а затем параметры блока CBL), если была ошибка. Например, строка, сообщающая об ошибке при выполнении запроса, может выглядеть так !:E=1025 @&#:C=0: (см. пример 1).

Знак **#** в начале строки (перед **?** или **!**) – признак того, что запрос подан из триггера или хранимой процедуры.

Параметры блока CBL:

- C – номер канала;
- L – длина буфера ответа (поле LnBufRow);
- P – режим обработки команды интерфейса нижнего уровня;
- R – приоритет канала;
- U – имя пользователя/пароль;
- I – внутренний системный номер записи, которая была обработана последней (RowId);
- K – значение параметра, необходимого при обработке запроса. Например, номер BLOB-поля;
- S – размер запроса или обработанных данных;
- O – код ошибки, переданный СУБД операционной/сетевой средой при обработке запроса (SysErr);
- A – количество реально обработанных записей (RowCount).

Примечание
Подробно блок управления запросом CBL, описание его полей и параметров приводится в документе «Интерфейс нижнего уровня».

TRID – начало новой транзакции.

E – код завершения операции.

Обозначения для переменных канала:

- T – Tr_Time;
- ID – ChTransactionID;
- M – Tr_mode;
- S – Tr_Status.

Также используется дополнительное поле EX для расширенной информации.

Указатель ключей

A
ANALYZE, 19
ANALYZE_NEED, 20
ANALYZE_USED, 20
AUTOINDEX, 20

B
BASE, 6
BRIEFVERSION, 28

C
CF, 6
CHECK, 12
CHECK_LEVELS, 28
COMPATIBILITY, 15

D
DEBUG, 13
DEFCOMM, 11

E
EVENTLIMIT, 14

F
FIXCHAN, 13

I
IGNERROR, 11
INMEMPOOL, 9

J
JDBCP, 15
JDBCS, 15
JEXIT, 12

K
K, 12
KILL, 10
KRBSRVC, 28

L
LDAPBASEDN, 27
LDAPFLTR, 27
LDAPPREFIX, 27
LDAPSRCHDN, 27
LDAPSRCHPW, 27
LDAPSRCHPWFILE, 27
LDAPSRV, 27
LDAPSUFFIX, 27
LDAPTOUT, 27
LOCK, 10
LOG, 21
LOGALL, 22
LOGFLIMIT, 21
LOGQUERY, 21

M
MBX, 15

N
NAME, 15
NOEXTFILE, 14
NOJEXIT, 12
NOLARGE, 12
NOLOG, 21
NOOUTFILE, 20
NOOUTPUT, 20
NOSQL, 10
NOSYNC, 15
NOTSP, 10

O
OUTFLIMIT, 20

P
PASS, 7
PASSFILE, 8
PBVCACHE, 9
PCONTCACHE, 9
PIDFILE, 26
POOL, 8
PPOOL, 9
PROCPRINT, 26

R
RAPID, 12
RESTRICTEXTFILE, 14
RO, 13

S
SAFE_MODE, 14
SETPASS, 8
SPOOL, 8
SYNC, 15

T
TCORRECT, 12
TCP, 15
TMPDIR, 12
TRACE, 22
TRACEFLIMIT, 21
TRACELOG, 26

U
U, 6

V

VERSION, 28

W

WLHB, 10