СИСТЕМА
УПРАВЛЕНИЯ
БАЗАМИ
ДАННЫХ

ЛИНТЕР®

ЛИНТЕР БАСТИОН

ЛИНТЕР СТАНДАРТ

Архивирование и восстановление базы данных

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 4
Назначение документа 4
Для кого предназначен документ 4
Необходимые предварительные знания 4
Условия выполнения утилиты 6
Запуск 7
Команды утилиты 12
Архивирование БД 12
Полное сохранение БД 12
Нарастающее архивирование БД 13
Архивирование БД с восстановлением 14
Архивирование только различающихся данных 14
Фразовые индексы 15
Восстановление БД 16
Порядок поиска устройств с архивными файлами 16
Создание архива на магнитной ленте 17
Выполнение сценария архивирования 18
Управление контрольными точками 18
Общие сведения об архивном файле 18
Проверка архивного файла 19
Пакетный режим работы 20
Ключи утилиты 21
Ключи управления утилитой 21
Регистрационные данные пользователя 21
Спецификация файла с регистрационными данными пользователя 21
Спецификация файла сценария архивирования 21
Спецификация архивного файла 21
Спецификация файла протоколирования 22
Постраничный вывод информации 22
Спецификация многотомного архива 22
Спецификация комментария в архивном файле 22
Пароль доступа к архивному файлу 22
Пароль для защищенной БД 22
Ввод пароля из файла 23
Установка интерактивного ввода пароля 23
Спецификация каталога для восстановления БД 23
Спецификация удаленного ЛИНТЕР-сервера 23
Управление процессом архивирования/восстановления БД 23
Управление файлами системного журнала 24
Синхронизация данных в архивном файле и в файле БД 24
Архивирование непосредственно в файлы БД 24
Архивирование непосредственно в файлы БД различающихся данных 25
Управление сжатием различающихся архивируемых данных 25
Управление слежением работоспособности утилиты архивирования 25
Переход в фоновый режим 26
Установка русскоязычного интерфейса 26
Установка англоязычного интерфейса 26
Спецификация файла выходных сообщений утилиты 26
Спецификация дополнения файла выходных сообщений утилиты 26
Игнорирование целостности архивного файла 27
Сохранение pid значения запущенного процесса в файл 27
Сообщение об окончании скачивания журнала 27
Предоставление информации о версии утилиты 27

Получение справочной информации ... 27
Ключи описания объектов ... 28
Маска наименования объекта ... 28
Сохранение всех объектов БД ... 28
Сохранение информации о пользователях/схемах БД ... 29
Сохранение хранимых событий ... 29
Полное сохранение базовых таблиц ... 29
Сохранение метаданных базовых таблиц ... 30
Сохранение информации о внешних ключах ... 30
Сохранение представлений БД ... 30
Сохранение синонимов ... 30
Сохранение ролей ... 31
Сохранение информации о владельцах роли ... 31
Сохранение привилегий на базовые таблицы ... 31
Сохранение привилегий на хранимые процедуры ... 31
Сохранение зависимостей между объектами БД ... 31
Сохранение объектов конкретного пользователя БД ... 32
Сохранение триггеров ... 33
Сохранение хранимых процедур ... 33
Сохранение последовательностей ... 33
Сохранение информации о серверах репликации ... 33
Сохранение правил репликации ... 33
Сохранение чужих объектов БД ... 33
Восстановление чужих объектов БД ... 33
Ключи нарастающего архивирования ... 34
Начать нарастающее архивирование ... 34
Добавить порцию накопленных изменений ... 34
Закончить нарастающее архивирование ... 34
Создать новый архивный том ... 34
Спецификация тома инкрементного архива ... 34
Ключи фразового индекса ... 34
Сохранение фразовых индексов ... 34
Ключи поиска устройств при восстановлении БД ... 35
Порядок поиска устройств при восстановлении БД ... 35
Ключи приоритета ... 35
Ключи просмотра и управления контрольными точками ... 35
Ключи для создания архива на магнитной ленте ... 36
Ключи для просмотра сохраненных в архиве БД устройств ... 36
Ключи обработки ошибок ... 37
Автоматизация архивирования БД ... 38
Структура сценария ... 38
Раздел «Начальные установки» ... 38
Раздел «Переменные» ... 40
Раздел «График выполнения» ... 41
График выполнения ежечасных операций ... 43
График выполнения ежедневных операций ... 45
График выполнения еженедельных операций ... 45
График выполнения ежемесячных операций ... 46
Выполнение в заданное время ... 46
Раздел «Специальные действия» ... 46
Действия в начале работы ... 48
Действия в конце работы ... 48
Действия при ошибочной ситуации ... 49
Операторы языка сценария ... 50
Оператор выражения ... 50

Комментарий 51
Оператор цикла 51
Оператор условия 51
Оператор запуска утилиты lhb 52
Оператор копирования файла 53
Оператор перемещения файла 53
Оператор переименования файла 53
Оператор удаления файла 54
Оператор создания каталога 54
Оператор синхронного запуска программы 54
Оператор асинхронного запуска программы 54
Оператор окончания программы 55
Оператор прерывания обработки 55
Оператор продолжения обработки 55
Оператор объявления секции обработки ошибок 55
Оператор перехода на обработку ошибки 55
Предопределенные переменные и константы 57
Предопределенные функции 58
Текущее время 58
Номер года 58
Полный номер года 58
Номер месяца 58
Номер дня недели 59
Название дня недели 59
Номер дня месяца 59
Номер текущего часа 59
Номер текущей минуты 59
Информация о файле 60
Выдача сообщения на консоль 60
Запись сообщения в файл протокола 60
Преобразование числа в строку 60
Ввод строки с консоли 61
Ввод пароля 61
Значение переменной окружения 61
Пример сценария архивирования 61
Особенности архивирования при параллельной работе с пользователями БД 63
Сообщения утилиты lhb 64
Структура протокола выполнения 64
Протокольные сообщения 65
Оперативное архивирование 68
SQL-запросы оперативного архивирования 68
Начать оперативное архивирование 68
Примеры запуска оперативного архивирования 69
Остановить оперативное архивирование 70
Примеры останова оперативного архивирования 71
Механизм асинхронного архивирования 72
Мониторинг процессов асинхронного архивирования 73
Коды завершения СУБД ЛИНТЕР при работе резервного сохранения 74
Приложение 1. Коды завершения утилиты lhb 76
Приложение 2. Примеры применения утилиты lhb 79
Указатель ключей 85

Предисловие

Назначение документа

Документ содержит описание утилиты `lhb`, выполняющей резервное сохранение информации из БД ЛИНТЕР в файл архива и, при необходимости, восстановление БД из созданного архива. Эти возможности реализуются на всех программных платформах, на которых функционирует СУБД ЛИНТЕР.

Приводится описание команд, передаваемых через командную строку. Приложение 1 содержит коды завершения и соответствующие тексты сообщений утилиты `lhb`. В приложении 2 приведены примеры использования программы резервного сохранения всей БД или отдельных её объектов, примеры восстановления БД из файла архива.

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 лиц, ответственных за поддержание целостности информации в СУБД ЛИНТЕР.

Утилита может использоваться:

- для полного сохранения БД;
- для полного сохранения БД с возможностью инкрементного наращивания;
- для последующего полного восстановления БД из созданных файлов архива;
- для сохранения отдельных объектов БД;
- для восстановления отдельных объектов БД;
- для планирования по времени и другим параметрам резервного архивирования БД (т.к. поддерживает язык сценариев);
- для просмотра и удаления существующих в БД контрольных точек;
- для тестирования и листинга файла архива;
- для переноса БД между платформами с одинаковой архитектурой.

Необходимые предварительные знания

Для работы с командным интерфейсом необходимо:

- знать основы реляционных БД;
- обладать соответствующими правами в СУБД ЛИНТЕР;
- уметь работать в соответствующей операционной системе на уровне простого пользователя.

Дополнительные документы

- Справочник по SQL

- Справочник кодов завершения
- Интерфейс нижнего уровня
- Встроенный SQL

Условия выполнения утилиты

Для работы `lhb` необходимы следующие условия:

- СУБД ЛИНТЕР должна быть активна (кроме случаев полного восстановления БД);
- в момент запуска `lhb` СУБД ЛИНТЕР должна иметь в общем случае два свободных канала;
- свободное место на диске для размещения файла(ов) архива, равное примерно половине размера самой СУБД (если предполагается полное сохранение БД с компрессией);
- 250 Мбайт свободной оперативной памяти;
- пользователь, от чьего имени запускается `lhb`, должен быть зарегистрирован в БД, к которой осуществляется доступ, и иметь права DBA.

Запуск

Для запуска утилиты необходимо в интерпретаторе команд операционной системы подать командную строку:

[1] <синтаксис>::=
lhb [<команда>[<пробел><ключ>[<пробел><ключ>]]...]
[2] <команда>::=
<идентификатор команды>[<пробел><параметр команды>]
[3] <ключ>::= -<идентификатор ключа>[<пробел><значение ключа>]

Идентификаторы и параметры команд приведены в таблице 1.

Таблица 1. Идентификаторы и параметры команд

<идентификатор команды>	<параметр команды>	Описание
cp		Управление контрольными точками
ef	<спецификация файла>	Пакетный режим работы
l		Общие сведения об архивном файле
r		Восстановление БД
s		Архивирование БД
script		Выполнение сценария архивирования
t		Проверка архивного файла
tape		Создание архива на магнитной ленте

Идентификаторы и значения ключей командной строки приведены в таблице 2.

Таблица 2. Идентификаторы и значения ключей командной строки

<идентификатор ключа>	<значение ключа>	Комментарий
Ключи управления утилитой		
-bg		см. Переход в фоновый режим
-briefversion		см. Предоставление информации о версии утилиты
-c	"<комментарий>"	см. Спецификация комментария в архивном файле
-db		см. Синхронизация данных в архивном файле и в файле БД
-dbonly		см. Архивирование непосредственно в файлы БД
-diff		см. Архивирование непосредственно в файлы БД различающихся данных
-f	<имя файла>	см. Спецификация архивного файла

<идентификатор ключа>	<значение ключа>	Комментарий
-fl	<имя файла>	см. Спецификация файла протоколирования
-ft	<имя файла>	см. Спецификация файла сценария архивирования
-g	[<пароль>]	см. Пароль доступа к архивному файлу
-h | -?		см. Получение справочной информации
-le		см. Установка англоязычного интерфейса
-lr		см. Установка русскоязычного интерфейса
-movecp		см. Управление файлами системного журнала
-mypid	<спецификация файла>	см. Сохранение pid значения запущенного процесса в файл
-n	[<имя сервера>]	см. Спецификация удаленного ЛИНТЕР-сервера
-nocompress		см. Управление сжатием различающихся архивируемых данных
-notestcrc		см. Игнорирование целостности архивного файла
-out	<файл>	см. Спецификация файла выходных сообщений утилиты
-out+	<файл>	см. Спецификация дополнения файла выходных сообщений утилиты
-p	[<путь>]	см. Спецификация каталога для восстановления БД
-pf	<имя файла>	см. Спецификация файла с регистрационными данными пользователя
-pg		см. Постраничный вывод информации
-pid	<pid процесса>	см. Сообщение об окончании скачивания журнала
-u	<имя/пароль>| <имя/?> | <имя>	см. Регистрационные данные пользователя
-v	[<размер> [K|M]]	см. Спецификация многотомного архива
-version		см. Предоставление информации о версии утилиты

<идентификатор ключа>	<значение ключа>	Комментарий
-wait		см. Управление процессом архивирования/ восстановления БД
-wd	<файловый дескриптор>	см. Управление слежением работоспособности утилиты архивирования
Ключи описания объектов		
-d		см. Сохранение зависимостей между объектами БД
-oa	[<имя объекта 1>[,…]]	см. Сохранение привилегий на базовые таблицы
-oall		см. Сохранение всех объектов БД
-oap	[<имя объекта 1>[,…]]	см. Сохранение привилегий на хранимые процедуры
-ob	[<имя правила 1>[,…]]	см. Сохранение правил репликации
-oe	[<имя объекта 1>[,…]]	см. Сохранение хранимых событий
-on	[<имя сервера 1>[,…]]	см. Сохранение информации о серверах репликации
-only	<имя пользователя 1>	см. Сохранение объектов конкретного пользователя БД
-op	[<имя проц 1>[,…]]	см. Сохранение хранимых процедур
-oq	[<имя посл 1>[,…]]	см. Сохранение последовательностей
-or	[<имя объекта 1>[,…]]	см. Сохранение ролей
-oref		см. Сохранение информации о внешних ключах
-os	[<имя объекта 1>[,…]]	см. Сохранение синонимов
-osr	[<имя объекта 1>[,…]]	см. Сохранение информации о владельцах роли
-ot	[<имя объекта 1>[,…]]	см. Полное сохранение базовых таблиц
-otr	[<имя тригг 1>[,…]]	см. Сохранение триггеров
-otwd	[<имя объекта 1>[,…]]	см. Сохранение метаданных базовых таблиц
-ou	[<имя объекта 1>[,…]]	см. Сохранение информации о пользователях БД
-ov	[<имя объекта 1>[,…]]	см. Сохранение представлений БД
-own		см. Восстановление чужих объектов БД

<идентификатор ключа>	<значение ключа>	Комментарий
-takeforeign		см. Сохранение чужих объектов БД
Ключи нарастающего архивирования		
-fo	<спецификация файла>	см. Спецификация тома инкрементного архива
-inc		см. Добавить порцию накопленных изменений
-startinc		см. Начать нарастающее архивирование
-stopinc		см. Закончить нарастающее архивирование
-vi		см. Создать новый архивный том
Ключи фразового индекса		
-pi		см. Сохранение фразовых индексов
Ключи поиска устройств при восстановлении БД		
-dp	[0|1|2]	см. Порядок поиска устройств при восстановлении БД
Ключи приоритета		
-priority	<уровень>	см. Ключи приоритета
Ключи просмотра и управления контрольными точками		
-clear	{<номер> | ALL}	см. Ключи просмотра и управления контрольными точками
-list		см. Ключи просмотра и управления контрольными точками
Ключи для создания архива на магнитной ленте		
-crpart	<имя раздела>	см. Ключи для создания архива на магнитной ленте
-dev	<имя ленты>	см. Ключи для создания архива на магнитной ленте
-list		см. Ключи для создания архива на магнитной ленте
Ключи для просмотра сохраненных в архиве БД устройств		
-ldev		см. Ключи для просмотра сохраненных в архиве БД устройств
Ключи обработки ошибок		
-qc	[<событие 1>[,…]]	см. Ключи обработки ошибок
-qq	[<событие 1>[,…]]	см. Ключи обработки ошибок
-qx	[<событие 1>[,…]]	см. Ключи обработки ошибок

За один сеанс работы утилита обрабатывает только одну команду.

Примечания

1. Команды и ключи допускается вводить как малыми, так и большими буквами (не допускаются лишь разные регистры букв в одной команде).
2. При вводе значений ключей (имена, пароли, наименования таблиц и т.п.) малые и большие буквы различаются.
3. Если задан ключ, не относящийся к команде, то выводится ошибочный ключ, и выполнение утилиты завершается.
4. Если строка местоположения файла (пути к файлу) состоит из двух и более слов, то её надо заключать в кавычки.
5. При указании в значении ключа списка значений, элементы списка должны разделяться запятой «,».

Команды утилиты

Архивирование БД

Команда `s` используется при разных способах архивирования БД.

Под *архивированием БД* понимается выгрузка и сохранение в архивном файле выбранных объектов (возможно, всех, которые есть в БД). Для сохранения БД СУБД ЛИНТЕР должна быть запущена. После сохранения можно восстановить все имеющиеся в ней объекты или любые из них по выбору. Восстановление может быть произведено в уже существующую БД.

Полное сохранение БД

Полное сохранение – выгрузка всей БД в архивный файл. Если полное сохранение выполнено без ключа `-oall`, то БД может быть восстановлена только целиком (выделение и восстановление отдельных объектов не допускается). Восстанавливаемая БД будет сформирована заново в указанном каталоге в том виде, в каком была сохранена (запуск ядра СУБД ЛИНТЕР при этом не требуется).

Примечание
При полном архивировании удаленных (remote) таблиц сохраняется только их описание (без данных).

В случае полного сохранения с ключом `-oall` в последующем возможно выборочное восстановление отдельных объектов.

Полное или выборочное сохранение можно выполнять в режиме параллельной работы с другими пользователями БД. Некоторые особенности получаемого в этом случае архивного файла БД описаны в разделе «Особенности архивирования при параллельной работе с пользователями БД».

Сохранение синонимов возможно, если указывается <имя пользователя.имя синонима>.

Примечания

1. К команде выборочного сохранения относятся ключи: `-u`, `-f`, `-ft`, `-pg`, `-c`, `-g`, `-v`, `-n`, `-oall`, `-ou`, `-ot`, `-oref`, `-ov`, `-os`, `-or`, `-osr`, `-oa`, `-otr`, `-op`, `-d`, `-on`, `-ob`, `-takeforeign`, `-only`.
2. К команде полного сохранения относятся ключи: `-u`, `-f`, `-ft`, `-pg`, `-c`, `-g`, `-p`, `-v`, `-n`, `-startinc`, `-inc`, `-stopinc`.
3. Ключи выборочного сохранения устарели, использовать не рекомендуется.

Примеры

1) Сохранение БД в файле `base.lhb` с комментарием к файлу:

```
lhb s -u SYSTEM/MANAGER8 -f base.lhb -c "Простой комментарий"
```

2) Сохранение БД в автоматическом режиме по заданному сценарию из файла `time.BSL` с паролем ABRAKADABRA:

```
lhb script -u SYSTEM/MANAGER8 -g ABRAKADABRA -ft time.BSL
```

Нарастающее архивирование БД

Под *нарастающим (инкрементным) архивированием БД* понимается специальный режим ведения полного архива БД. В отличие от обычного сохранения, которое является разовой работой утилиты `архивирования`, нарастающее архивирование – синхронный с функционированием БД и продолжающийся во времени процесс, в ходе которого сделанные в БД изменения запоминаются в системном журнале для последующего внесения их в архив. Нарастающее архивирование целесообразно применять в тех случаях, когда регулярное полное сохранение БД является обременительным, а выборочное сохранение затруднено из-за непредсказуемости потока изменений (т.е. трудно или невозможно полностью задать с помощью шаблонов имен перечень изменяемых объектов БД).

Процесс нарастающего архивирования можно разделить на три стадии:

1) инициирование нарастающего архивирования.

   Задается ключом `-startinc`. По этому ключу утилита автоматически создает полный архив БД. Этот архив является базовым архивом (контрольной точкой БД, от которой отсчитываются все сделанные в БД изменения). После его создания он получает статус открытого архива, который можно дописывать по мере изменения БД. Открытый архив не должен удаляться или изменяться каким-либо другим способом, кроме как самой утилитой `lhb`. Кроме того, при инициировании нарастающего архивирования устанавливается режим запрета удаления данных из системного журнала БД. Это делается для того, чтобы исключить возможность потери произведенных в БД изменений.

2) внесение накопленных изменений в открытый архив.

   Задается ключом `-inc`. По этому ключу в открытый архив вносятся все изменения, накопленные в системном журнале (соответственно, в БД) после создания базового архива или предыдущего сеанса обновления открытого архива. По завершении этой операции открытый архив будет представлять полный и реальный архив БД. После успешного обновления открытого архива системный журнал очищается.

3) окончание нарастающего архивирования.

   Задается ключом `-stopinc`. По этому ключу никакого сохранения данных не происходит, просто восстанавливается возможность удаления записей системного журнала.

Для системы контрольных точек при указании ключа `-stopinc` стирается контрольная точка, соответствующая заданному архивному файлу (необходимо указывать его имя).

Примечания

1. При восстановлении БД из нарастающего архива предварительно необходимо внести в открытый архив все накопленные изменения и только потом начать процесс восстановления БД.
2. Так как при нарастающем архивировании устанавливается запрет на удаление из системного журнала, важно правильно определить периодичность обновления открытого архива. При интенсивном изменении БД и редком обновлении открытого архива системный журнал может принять слишком большие размеры (вплоть до исчерпания свободных ресурсов на диске) со всеми вытекающими отсюда последствиями.

3. К команде сохранения в нарастающем режиме относятся следующие ключи: `-u`, `-f`, `-ft`, `-pg`, `-v`, `-c`, `-g`, `-p`, `-n`, `-startinc`, `-inc`, `-stopinc`.

Примеры

1) Установить режим нарастающего архивирования:

```
lhb s -u SYSTEM/MANAGER8 -startinc -f base.lhb -c "Нарастающий
 архив"
```

2) Внести накопленные изменения:

```
lhb s -u SYSTEM/MANAGER8 -inc -f base.lhb
```

3) Закончить режим нарастающего архивирования:

```
lhb s -u SYSTEM/MANAGER8 -stopinc -f base.lhb
```

Архивирование БД с восстановлением

Под сохранением с восстановлением понимается специальный режим полного архивирования БД (команда `s`). В этом режиме кроме создания lhb-архива БД выполняется одновременное восстановление полученной информации в БД (ключ `-db`). При этом создается и lhb-архив БД, и копия БД в отдельном каталоге, как если бы она была развернута из этого архива. Это уменьшает время восстановления БД из архива практически до 0.

Режим имеет модификацию (ключ `-dbonly`), при которой данные только разворачиваются в каталог БД, но не сохраняются в архивном файле. Архивный файл создается, но содержит только служебную информацию. В этом режиме `lhb` работает в системе горячего резервирования.

При сохранении с восстановлением можно использовать инкрементное архивирование (`-startinc`, `-inc`), а также режим ожидания (`-wait`).

Архивирование только различающихся данных

Это специальный режим (задается ключом `-diff`) полного архивирования с одновременным восстановлением данных только в БД. Выполняется архивирование только различий оригинальной и архивной БД. Предполагается, что разархивирование производится в каталог, который уже содержит копию файлов БД, созданную `lhb`, или выполняется пофайловое копирование БД. После создания этой копии некоторые блоки БД могут поменяться, а некоторые будут неизменными. Вместо повторного архивирования всех данных БД `lhb` получает от ядра СУБД ЛИНТЕР только контрольные суммы блоков данных архивной БД. Она также высчитывает контрольные суммы блоков файлов архивируемой БД. При несовпадении контрольных сумм изменившиеся блоки архивируются. Таким образом может быть значительно снижен трафик данных, при некотором увеличении нагрузки на процессор и файловую систему. После архивирования всех различающихся файлов таблиц, системный журнал архивируется обычным образом. Это позволяет одновременно применять данный режим с инкрементным режимом и режимом ожидания.

Поскольку в данном режиме архивная БД уже существует, то в интервале от начала архивирования до его окончания она может прийти в некорректное состояние. Для исключения работы ядра СУБД ЛИНТЕР с такой БД, перед началом архивирования она приводится в непригодное для запуска СУБД ЛИНТЕР состояние – удаляются

файлы системного журнала, рабочие файлы и файлы сортировки. После окончания архивирования эти файлы создаются вновь.

В случае если размер файла уменьшился или файл был удален в результате удаления таблицы, соответствующие файлы архивной БД также будут усечены или удалены.

Если БД архивируется в пустой каталог, то данный режим аналогичен режиму обычного архивирования, за исключением дополнительных накладных расходов на передачу контрольных сумм.

Необходимо различать данный режим с режимом инкрементного архивирования. При инкрементном архивировании необходимо предварительно создать полную копию архивируемой БД с контрольной точкой в БД (`-startinc`), после чего в архивируемую БД будет добавляться только системный журнал с накопленными изменениями. В режиме отличий контрольная точка необязательна, т.к. производится сверка не файлов системного журнала, а файлов таблиц БД.

Режим отличий может применяться совместно с инкрементным режимом. В этом случае вместо первоначального архивирования всей БД (`-startinc`) будет производиться архивирование только различающихся файлов таблиц. Работа с системным журналом в этих режимах осуществляется одинаково.

В режиме сохранения отличий никогда не учитываются устройства, хранимые в БД. Можно использовать только те устройства, которые указаны в переменных окружения. Это сделано с целью исключения потери файлов архивируемой БД при локальной работе.

В связи с этим необходимо быть особенно внимательным при назначении устройств при работе в режиме сохранения отличий. При неправильном указании каталога устройства размещенная в нем БД будет замещена, причем, в отличие от остальных режимов, без предварительного предупреждения и подтверждения данной операции. Это относится и к каталогам архивируемой БД (может произойти потеря информации).

В режиме сохранения отличий можно отключить компрессию ключом `-nocompress`. На быстрых сетях это может дать некоторое увеличение производительности, особенно при архивировании в пустой каталог.

Фразовые индексы

Архивирование информации о фразовых индексах (список таблиц и проиндексированных столбцов) выполняется только при полном сохранения БД (без нарастающего архивирования).

Примечание
Если ключ архивирования фразовых индексов `-pi` задан для любого вида архивирования, кроме полного архивирования БД, он игнорируется.

Восстановление информации о фразовых индексах происходит в конце процесса полного восстановления БД из архива. В результате восстановления информации о фразовых индексах создается файл `phrase.rbd` в том же каталоге, что и файлы БД. Наличие данного файла является сигналом для ядра СУБД на создание фразовых индексов для таблиц и их столбцов, перечисленных в этом файле.

Непосредственное создание фразовых индексов происходит при первом запуске ядра СУБД для работы с восстановленной БД. Во время восстановления фразовых индексов

активная работа с ядром невозможна, т.к. механизм квантования еще не запущен (вся работа по восстановлению фразовых индексов производится из функции, отвечающей за инициализацию ядра). О том, что ядро занято построением фразовых индексов, на консоль выдается соответствующее сообщение, содержащее имя таблицы, её владельца и столбец, для которого создается фразовый индекс.

После успешного создания фразового индекса файл `phrase.rbd` удаляется (при неудачном восстановлении фразовых индексов он сохраняется для возможного последующего применения).

По окончании восстановления фразовых индексов ядро СУБД готово к работе.

Восстановление БД

Команда `r` используется для восстановления БД из архива.

Под *восстановлением БД* понимается процесс создания объектов БД, идентичных сохраненным в архивном файле (в случае, если было проведено выборочное сохранение), и создание новой БД, если было проведено полное сохранение. При выборочном восстановлении СУБД ЛИНТЕР должна быть активна. Процесс полного восстановления БД выполняется автономно, и функционирование СУБД ЛИНТЕР не требуется. Выборочное восстановление БД возможно при параллельной работе с другими пользователями БД. Однако в этом случае возможны взаимные помехи в работе, например, при восстановлении данных таблицы, с которой в этот момент работает пользователь БД (добавляет в нее записи). Особенности восстановления из архива, полученного при параллельной работе с другими пользователями БД, приведены в разделе «Особенности архивирования при параллельной работе с пользователями БД».

Примечания

1. Для удаленных (remote) таблиц восстанавливается только их описание (без данных).
2. Хотя процесс выборочного восстановления возможен при параллельной работе с другими пользователями, во избежание нежелательных последствий рекомендуется осуществлять его только в чрезвычайных ситуациях с соблюдением необходимых мер предосторожности.
3. К команде восстановления относятся ключи: `-u`, `-f`, `-pg`, `-p`, `-oall`, `-ou`, `-otwd`, `-ot`, `-oref`, `-ov`, `-os`, `-or`, `-osr`, `-oa`, `-d`, `-otr`, `-op`, `-on`, `-ob`, `-own`.

Примеры

1) Полное восстановление БД из архива `base.lhb` в текущем каталоге:

```
lhb r -f base.lhb
```

2) Полное восстановление БД из архива `base.lhb` в каталоге `C:\DB`:

```
lhb r -f base.lhb -p C:\DB
```

Порядок поиска устройств с архивными файлами

При архивировании БД вместе с сохранением собственно данных сохраняется и информация об их местоположении в файловой структуре операционной системы (например, для повышения производительности файлы данных и индексов могут быть размещены на разных внешних устройствах).

В ряде случаев при восстановлении БД из архива может оказаться, что для некоторых файлов требуемых устройств (или каталогов) в файловой структуре операционной системы обнаружить не удалось (например, восстановление БД выполняется на новой аппаратной конфигурации). В такой ситуации утилите необходимы указания о порядке размещения восстанавливаемых файлов. Для этого используется ключ `-dp`.

Поиск устройства для размещения файла восстанавливаемой БД выполняется по следующим правилам:

1) если местоположение восстанавливаемого файла соответствует переменной окружения `SY00`, то он размещается в каталоге, определяемом ключом `-p` командной строки или, если этот ключ не задан, в текущем каталоге;
2) если при восстановлении встречается файл, относящийся к устройству, отличному от переменной окружения `SY00`, то поиск пути для определения местоположения файла выполняется следующим образом:
   - если ключ `-dp` не указан или указан без параметра, либо указан с параметром `0`, то производится поиск имени устройства среди переменных окружения;
   - если переменная окружения существует, то считывается значение этой переменной, и на основании его формируется полный путь восстанавливаемого файла. Файл будет восстановлен не в жестко заданном по ключу `-p` пути, а в сформированном;
   - если же переменная окружения, соответствующая имени устройства, не обнаруживается среди переменных окружения, то производится поиск имени устройства в списке устройств из файла архива. Список устройств формируется в самом начале процедуры создания файла архива и содержит информацию об именах устройств и соответствующих им каталогах на дисках;
   - если имя устройства нигде не найдено, то на консоль выдается запрос на размещение восстанавливаемого файла в каталог по умолчанию, т.е. в каталог, из которого была запущена утилита `lhb`, или в каталог, указанный с помощью ключа `-p`;
   - если ключ `-dp` задан с параметром `1`, то порядок поиска устройства будет обратным по сравнению с параметром `0`, т.е. сначала поиск будет осуществляться в списке устройств из файла архива, а затем, если устройство не будет найдено, среди переменных окружения;
   - если ключ `-dp` задан с параметром `2`, то поиск устройств для размещения файлов выполняться не будет, и по умолчанию все файлы восстанавливаемой БД будут размещаться в каталог по умолчанию. Дополнительные вопросы для подтверждения такого размещения файлов выдаваться не будут.

Создание архива на магнитной ленте

Команда `tape` позволяет создавать архивный файл на магнитной ленте.

В некоторых случаях (повышенные требования к сохранности и надежности архивных носителей, необходимость ограничения доступа к архивным файлам, технические и экономические ограничения и др.) предпочтительнее создавать и/или хранить архивные файлы БД на магнитных лентах.

Создание архива на магнитной ленте выполняется в следующей последовательности:

1) монтирование лентопротяжного устройства. Для этого используется команда `tape` с ключом `-dev`;

2) создание раздела на ленте (метки ленты) при первоначальном использовании ленты или при её переиспользовании. Для этого применяется команда `tape` с ключом `-crpart`;

3) создание архивного файла и запись его на ленту. Для этого используется команда сохранения `s` с ключом `-dev`.

Примеры

1) Создание нового архива:

```
lhb tape -dev MT1: -crpart ARCHIVES  -c "Для служебного
 пользования";
lhb s -u SYSTEM/MANAGER8 -dev MT1: -f DB_BANK.lhb -c "Банковские
 расчеты"
```

2) Добавление архива на магнитную ленту:

```
lhb tape -dev MT1:
lhb s -u SYSTEM/MANAGER8 -dev MT1: -f DB_STORE.lhb -c "Склад"
```

3) Просмотр содержимого ленты:

```
lhb tape -dev MT1: -list
```

Выполнение сценария архивирования

Команда `script` запускает на выполнение файл сценария архивирования БД. Описание языка сценария архивирования приведено в разделе «Автоматизация архивирования БД».

Управление контрольными точками

Команда `cp` позволяет управлять контрольными точками архивного файла. Ключи команды описаны в разделе «Ключи просмотра и управления контрольными точками».

Пример

См. приложение 2.

Общие сведения об архивном файле

Команда `l` позволяет получить общую информацию об архивном файле.

Примечание
К команде относятся ключи: `-f`, `-pg`, `-g`.

Пример

Получить общие сведения об архиве.

```
lhb l -f c:\linter\base.lhb

Linter On-line BACKUP utility v.4.1.0.893
```

```
===Файл c:\linter\base.lhb===
СОДЕРЖИМОЕ ФАЙЛА АРХИВА
----------------------------------------
ПАРОЛЬ АРХИВА:     НЕ ЗАДАН
МЕТОД СЖАТИЯ:      STANDARD
ПОЛЬЗОВАТЕЛЬ:      SYSTEM
ИМЯ БД:            DEMO DATABASE
ВЕРСИЯ БД:         6.1
ВЕРСИЯ АРХИВА:     4.1.0
ВРЕМЯ:             10.09.2007 12:33:43 GMT
КОНТРОЛЬНАЯ ТОЧКА:NONE
КОДИРОВКА:         WINDOWS
КОММЕНТАРИЙ:       Дубликат архива БД
```

Проверка архивного файла

Команда `t` позволяет проверить архивный файл.

Под *проверкой (тестированием) архивного файла* понимается контроль целостности и сохранности архива БД. В процессе архивирования утилита, помимо записи сохраняемых объектов в архивный файл, подсчитывает и записывает контрольные суммы этих объектов. При тестировании архива выполняется сверка текущей контрольной суммы каждого заархивированного объекта с эталонной контрольной суммой. Результаты тестирования выдаются в виде протокола, где напротив каждого архивного объекта проставлен результат проверки.

Примечание
К команде тестирования относятся ключи: `-f`, `-pg`, `-g`.

Пример

Проверка архивного файла `base.lhb` (выборочное архивирование БД).

```
lhb t -f c:\linter\base.lhb
Linter On-line BACKUP utility v.4.1.0.893
===Файл c:\linter\base.lhb===
ПРОВЕРКА АРХИВА
----------------------------
ПАРОЛЬ АРХИВА:     НЕ ЗАДАН
МЕТОД СЖАТИЯ:      STANDARD
ПОЛЬЗОВАТЕЛЬ:      SYSTEM
ИМЯ БД:            TRIGG
ВЕРСИЯ БД:          6.1
ВЕРСИЯ АРХИВА:     4.1.0
ВРЕМЯ:             10.09.2007 13:45:43 GMT
КОНТРОЛЬНАЯ ТОЧКА:NONE
КОДИРОВКА:         WINDOWS
КОММЕНТАРИЙ:
```

```
Тип            Имя                Пользователь     Прогресс
--------- ------------------- ----------------- -------
(-)Table..... AUTO                SYSTEM         100% OK
(-)Table..... PERSON              SYSTEM         100% OK
(-)Table..... $$$COMMENTS         SYSTEM         100% OK
(-)Table..... ALL_SEQUENCES       SYSTEM         100% OK
(-)Table..... ALL_PRILTREE        SYSTEM         100% OK
(-)Table..... M_COMPLECTATION     SYSTEM         100% OK
(-)Table..... M_CUSTOMERS         SYSTEM         100% OK
(-)Table..... M_TYPE_DOC          SYSTEM         100% OK
(-)Table..... M_ED_IZM            SYSTEM         100% OK
(-)Table..... TYPEINFO            SYSTEM         100% OK
(-)Table..... PRIV_TYPES          SYSTEM         100% OK
(-)Table..... $$$PRCD             SYSTEM         100% OK
(-)Table..... $$$PROC             SYSTEM         100% OK
(-)Table..... $$$TRIG             SYSTEM         100% OK
(-)Table..... $$$DEVICE           SYSTEM         100% OK
(-)Table..... $$$STATION          SYSTEM         100% OK
(-)Table..... $$$RELATION         SYSTEM         100% OK
(-)Table..... $$$AUDIT            SYSTEM         100% OK
(-)Table..... $$$GROUP            SYSTEM         100% OK
(-)Table..... $$$LEVEL            SYSTEM         100% OK
```

Атрибуты архивного файла:

1) `МЕТОД СЖАТИЯ` – указывает на то, что при создании архива выполняется уплотнение данных;
2) `ВЕРСИЯ АРХИВА` – версия утилиты `lhb`, с помощью которой создан архив;
3) столбец `Тип` – название объекта. Знак «-» перед названием указывает, что объект был сохранен по ключу, знак «+» соответствует зависимому объекту, сохраненному автоматически;
4) столбец `Прогресс` показывает динамику тестирования каждого объекта архива. В процессе тестирования он изменяется от 0 до 100 и информирует пользователя о степени завершенности процесса;
5) дополнительная информация – при проверке полного архива БД дополнительно выводятся информационные сообщения, относящиеся к архиву в целом и к контрольным точкам, например:

```
Тип                          Дата                        Прогресс
--------------------------- ------------------------ --------
Конец основной секции данных 10.09.2007 13:46:04 GMT  100% OK
Конец добавленной секции     10.09.2007 13:46:04 GMT  100% OK
Конец архива                 10.09.2007 13:46:04 GMT  100% OK
```

Пакетный режим работы

Команда `ef <имя файла>` позволяет запускать команды утилиты из текстового файла. В файле допустимы переводы строки – они будут восприниматься как пробелы.

Ключи утилиты

Ключи управления утилитой

Ключи управления предназначены для обеспечения утилиты `lhb` дополнительной информацией, необходимой при выполнении операций архивирования (восстановления) БД.

Регистрационные данные пользователя

`-u <имя/пароль>| <имя/?>| <имя>`

Задает имя и пароль пользователя БД. Если <пароль> пропущен или задан в виде «?», то он будет запрошен интерактивно. В этом случае вводимое значение пароля на экране будет представлено знаками «****». Когда имя и пароль задаются в виде <имя/пароль>, значение <пароля> при вводе не скрывается и может быть доступно любому заинтересованному лицу.

Примечание
Запрос на интерактивный ввод пароля предваряется подсказкой: «`Введите ваш пароль:`».

Спецификация файла с регистрационными данными пользователя

`-pf <имя файла>`

Задает спецификацию (имя и путь) текстового файла с регистрационными данными пользователя.

Парольный файл должен иметь кодировку CP1251 при работе в ОС типа Windows и KOI8-R в ОС типа Linux, ЗОСРВ Нейтрино.

Формат файла:

```
USER=<имя пользователя>
PASSWORD=<пароль пользователя>
```

Спецификация файла сценария архивирования

`-ft <имя файла>`

Задает спецификацию (имя и путь) текстового файла с описанием сценария архивирования БД (раздел «Автоматизация архивирования БД»).

Спецификация архивного файла

`-f <имя файла>`

Задает спецификацию (имя и путь) архивного файла. Если ключ не задан, по умолчанию архивный файл будет иметь имя `db.lhb` и размещаться в текущем каталоге.

Спецификация файла протоколирования

`-fl <имя файла>`

Создает файл с расширением `.log` для записи script-операций. Имеет значение только при задании ключа `-ft`.

Постраничный вывод информации

`-pg`

Задает постраничный вывод информации на монитор.

Спецификация многотомного архива

`-v [<размер> [K|M]]`

Разрешает создание многотомного архива. Параметр <размер> – целое положительное число, задающее размер архивного тома. Параметр [K|M] указывает единицу измерения объема архивного тома: K – в килобайтах, M – в мегабайтах. Если единица измерения не задана, по умолчанию принимается K. Если <размер> не задан, то размер архивного тома определяется автоматически. В случае автоматического создания архивных томов файлы архива будут размещаться на магнитном носителе до исчерпания свободного пространства, и только после этого пользователю будет предложено сменить или указать новый магнитный носитель архива. Если ключ не задан, по умолчанию действует режим автоматического создания нового тома.

Спецификация комментария в архивном файле

`-c "<комментарий>"`

Задает комментарий пользователя, сохраняемый в архивном файле.

Пароль доступа к архивному файлу

`-g [<пароль>]`

Задает пароль для последующего доступа к архиву. Если ключ задан, а <пароль> не указан, то он будет запрошен интерактивно. Если архив БД создан с паролем, то восстановление БД из этого архива возможно только при предъявлении пароля. Если пароль утерян, то восстановление БД из такого архива невозможно. При вводе <пароля> регистр клавиатуры не имеет значения.

Пароль для защищенной БД

Ключ

```
-pass=<пароль>
<пароль>::=пароль доступа к защищенной БД
```

Описание

Задает пароль для защищенной БД.

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

Ввод пароля из файла

Ключ

```
-passfile=<файл>
<файл>::=спецификация текстового файла
```

Описание

Задает ввод пароля доступа к защищенной БД из первой строки указанного текстового файла. Файл должен содержать строку пароля доступа.

Пример

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

Установка интерактивного ввода пароля

```
-setpass
```

Задает чтение пароля БД с терминала.

Спецификация каталога для восстановления БД

```
-p [<путь>]
```

Задает путь к каталогу, в котором должна быть восстановлена БД. Если <путь> не указан или ключ не задан, то БД создается в текущем каталоге. Если БД восстанавливается в каталоге, где уже размещена некоторая БД, то при наличии в нем дубликатов восстанавливаемых файлов будет выдаваться запрос на перезапись этих файлов.

Спецификация удаленного ЛИНТЕР-сервера

```
-n [<имя сервера>]
```

Задает имя удаленного ЛИНТЕР-сервера. Ключ необходим, если выполняется архивирование (восстановление) удаленной БД.

Управление процессом архивирования/восстановления БД

```
-wait
```

Ожидать последние изменения в БД (при использовании с командой `s`) или в архивном файле (при использовании с командой `r`). Ключ заставляет файл архива находиться в открытом состоянии в ожидании новой архивной информации. Если пользователь нажимает клавиши **<Ctrl>+<Break>** для прекращения работы, то утилита сохраняет (восстанавливает) последний блок файла архива и завершает работу, не дожидаясь поступления изменений от ядра СУБД.

Ключ применим как к простому архивированию, так и к инкрементному (нарастающему). В последнем случае он может быть использован как при старте инкрементного архива (`-startinc`), так и при его продолжении (`-inc`).

Примеры

1) непрерывное архивирование БД

```
lhb s -wait -f arc.lhb ...
```

Файл архива будет закрыт только после принудительного прерывания процесса архивации с помощью клавиш **<Ctrl>+<Break>**.

2) восстановление БД без закрытия файла архива

```
lhb r -wait -f arc.lhb ...
```

БД будет восстановлена, но файл архива останется открытым до принудительного прерывания процесса архивации с помощью клавиш **<Ctrl>+<Break>**.

Если одновременно используется один и тот же архивный файл для создания архива в режиме ожидания изменений (в одном каталоге/узле) и восстановления архива (в том же режиме, но в другом каталоге/узле), например, для создания быстро доступной резервной копии БД:

```
lhb s -wait -f arc.lhb ...
lhb r -wait -f arc.lhb ...
```

то процесс, выполняющий восстановление БД, перестанет обрабатывать вновь добавленные в архивный файл записи при достижении конца архивного файла (несмотря на ключ `-wait`). Это позволяет с помощью клавиш **<Ctrl>+<Break>** завершить в подходящий момент процесс восстановления резервной БД и быстро запустить соответствующий ей ЛИНТЕР-сервер.

Управление файлами системного журнала

`-movecp`

При архивации системного журнала выполнять перенос контрольной точки на текущий файл журнала. Это предотвращает накопление файлов системного журнала при длительной работе программы в режиме ожидания изменений. Ключ используется при полном сохранении БД совместно с ключом `-wait`.

Синхронизация данных в архивном файле и в файле БД

`-db`

Выполнять синхронизацию данных в архивном файле и в файле БД. Полученный при сохранении блок данных записывается в архивный файл и одновременно разворачивается на диск в виде файлов БД.

Ключ используется при полном сохранении БД, фактически эквивалентен совместному использованию режимов сохранения (команда `s`) и восстановления (команда `r`) БД.

Архивирование непосредственно в файлы БД

`-dbonly`

Выполнять архивирование непосредственно в файлы БД. В процессе архивирования в файл архива записывается только служебная информация, а архивируемые данные сохраняются в БД. Средний объем служебной информации равен примерно 340 байт и может увеличиваться на несколько десятков байт при работе в режиме нарастающего архивирования. Ключ используется при полном сохранении БД.

Архивирование непосредственно в файлы БД различающихся данных

`-diff`

Выполнять архивирование непосредственно в файлы БД только различающихся данных. Работает аналогично ключу `-dbonly`, но загружает не все данные, а производит предварительное сравнение контрольных сумм источника и приемника данных. Архивируются только отличающиеся блоки данных (см. пункт «Архивирование только различающихся данных»).

Управление сжатием различающихся архивируемых данных

`-nocompress`

Отключение сжатия архивируемых данных в режиме архивирования только различающихся данных.

При задании данного ключа блоки данных не будут подвергаться сжатию при архивировании различающихся данных. Это может дать некоторое увеличение производительности, особенно при архивировании в пустой каталог.

Управление слежением работоспособности утилиты архивирования

`-wd <файловый дескриптор>`

Выполнять периодическую запись символа в файл, заданный <файловым дескриптором>. В этом режиме при каждой операции обращения к ядру за новой порцией данных производится вывод одного символа 'B' при работе в wait-режиме после получения части журнала, необходимой для корректного восстановления БД из архива. После получения этой минимальной необходимой части журнала будет послан сигнал процессу:

```
lhb -pid <дескриптор процесса> -dsig <номер сигнала>
```

Аналогично в wait-режиме, когда `lhb` «догоняет» ядро, т.е. получает весь имеющийся на тот момент журнал, посылается сигнал процессу:

```
lhb -pid <дескриптор процесса> -sig <номер сигнала>
```

и в <файловый дескриптор> пишется символ 'C'.

Этот режим может применяться для слежения за работоспособностью `lhb` (watchdog). В нормальном режиме обращение за новой порцией данных происходит примерно один раз в 30 секунд. Если за 40-60 секунд не было выведено ни одного символа, необходимо

предпринять действия по рестарту lhb, а возможно, и ядра СУБД ЛИНТЕР после дополнительных проверок.

Необходимо также учитывать, что при активном поступлении данных программа слежения должна успевать читать записываемые символы, или переданный lhb файловый дескриптор должен быть переведен в асинхронный режим перед запуском lhb.

Недопустимо значение <файлового дескриптора> равное 1 при использовании опций -out, -out+.

Примечание
Ключ используется при полном сохранении БД.

Переход в фоновый режим

-bg

Запускает функционирование утилиты в фоновом режиме.

Используется только в ОС типа Linux, ЗОСРВ Нейтрино.

Установка русскоязычного интерфейса

-lr

Выводить сообщения по-русски (значение по умолчанию).

Установка англоязычного интерфейса

-le

Выводить сообщения по-английски.

Примечание
Ключи -lr, -le оказывают действие только на текущий сеанс работы утилиты.

Спецификация файла выходных сообщений утилиты

-out <файл>

Задает спецификацию файла, в который будут записываться все сообщения утилиты lhb. Если указанный файл уже существует, то вся информация из него предварительно удаляется.

Спецификация дополнения файла выходных сообщений утилиты

-out+ <файл>

Задает спецификацию существующего файла (обычно созданного ранее с помощью ключа -out), в конец которого будут дописываться все сообщения утилиты lhb. Если указанный файл не существует, то он будет создан.

Игнорирование целостности архивного файла

`-notestcrc`

Игнорировать несовпадающую контрольную сумму целостности архивного файла (CRC).

Сохранение pid значения запущенного процесса в файл

`-mypid <спецификация файла>`

Задаёт местоположение и имя текстового файла, в который будет записан pid запущенного процесса архивирования.

Примечание
Поддерживается в ОС Linux, ЗОСРВ Нейтрино.

Пример

```
lhb s -wait -u SYSTEM/MANAGER8 -mypid lhbpidfile.pid
```

Сообщение об окончании скачивания журнала

`-pid <pid процесса>`

Задаёт pid процесса, которому будет послан сигнал по окончании частичного или полного скачивания системного журнала СУБД. Номер сигнала задается опцией dsig или sig. См. также описание опции wd.

Примечание
Поддерживается в ОС Linux, ЗОСРВ Нейтрино.

Предоставление информации о версии утилиты

`-version`

Предоставить полную информацию о версии утилиты.

`-briefversion`

Предоставить краткую информацию о версии утилиты.

Получение справочной информации

`-h |-?`

Для получения справочной информации об утилите (список команд и ключей с краткими пояснениями) следует запустить утилиту без командной строки или с ключом `-h` (`-?`).

Примечание
Местоположение ключа в командной строке имеет значение: например, в случае "./lhb -? -le" первым обнаруживается и обрабатывается ключ `-?`, остальные ключи игнорируются.

В случае "./lhb -le -?" первой обрабатывается команда такого `-le`, а поскольку команда отсутствует, то выдается ошибка.

Ключи описания объектов

Ключи описания объектов предназначены для обеспечения утилиты `lhb` информацией об объектах БД, которые должны быть обработаны при выполнении операций архивирования (восстановления) БД.

Маска наименования объекта

При описании ключей этой группы параметр <имя объекта> определяет маску объекта, к которому относится ключ. Маска может задавать как конкретное имя объекта, (например, таблицу PERSON), так и правило выбора множества объектов аналогично конструкции LIKE в языке SQL (см. документ «Справочник по SQL»). Знак «*» в маске соответствует не менее одному любому символу, знак «?» указывает на один любой символ. Например, «S*» задает имена всех объектов, начинающиеся на «S*», «*doc*» – имена всех объектов, у которых в имени в любом месте присутствует набор знаков «*doc*», маска «tab?» определяет имена объектов, имеющих одинаковую длину имени и отличающихся только одним последним знаком – «tab1», «tab2». Допускается задавать несколько масок, разделенных запятой.

Пример

PERSON, AUTO, BANK – конкретные имена таблиц БД, TST* – маска имен таблиц, название которых начинается с «TST».

```
-ot PERSON,AUTO,TST*,BANK
```

Сохранение всех объектов БД

```
-oall
```

Примечание
Ключ устарел, использовать не рекомендуется.

Сохранение всех следующих объектов БД:

- базовые таблицы;
- представления;
- синонимы;
- последовательности;
- хранимые процедуры и триггеры;
- события;
- серверы;
- правила репликации;
- пользователи, схемы и роли;
- элементы дискреционного доступа;

- элементы ссылочной целостности.

Примечание
В текущей версии утилиты при сохранении БД по ключу `-oall` не сохраняются кодировки, трансляции, уровни доступа, станции и устройства.

Примечание
Восстановленные установки аудита будут автоматически активированы только после перезапуска ядра.

Ключ используется для полного логического сохранения БД (т.е. сначала определяются все объекты БД, затем они поочередно архивируются). При полном сохранении без использования этого ключа выполняется физическое (по файлам) сохранение БД. Кроме того, при полном сохранении с ключом `-oall` возможна потеря данных из архива, полученного при параллельной работе с другими пользователями БД, что исключено при физическом сохранении БД (раздел «Особенности архивирования при параллельной работе с пользователями БД»).

Сохранение информации о пользователях/схемах БД

`-ou  [<имя объекта 1>[,…]]`

Задает объект БД «Пользователь» или «Схема». Значение <имя объекта…> должно представлять имя пользователя или имя схемы или маску их имен.

Примечание
При восстановлении схемы обязательно требуется, чтобы пользователь-владелец схемы был восстановлен раньше самой схемы, в противном случае `lhb` вернет код завершения 30126 – («`Ошибка восстановления пользователя`») и код завершения СУБД ЛИНТЕР 2203 («`Несуществующий пользователь`»).

Сохранение хранимых событий

`-oe  [<имя объекта 1>[,…]]`

Задает объект БД «Хранимые события». Значение <имя объекта…> должно представлять имя хранимого события или маску имен хранимых событий.

Полное сохранение базовых таблиц

`-ot  [<имя объекта 1>[,…]]`

Примечание
Ключ устарел, использовать не рекомендуется.

Задает объект БД «Таблицы с данными». Значение <имя объекта…> должно представлять имя таблицы или маску имен таблиц БД. Существует возможность указывать владельца объекта или имя схемы (например, `-ot SYSTEM.AUTO`). При использовании этого ключа выполняется полное архивирование таблицы и, соответственно, возможно её полное восстановление.

Маска объекта допускает 2 представления:

- маска заключается в двойные кавычки, не может содержать символа двойной кавычки, а символы '*' и '?' всегда трактуются как метасимволы;
- любой символ маски, следующий за экранирующим символом '\' теряет своё специальное значение. Это позволяет задавать маски, содержащие символы '*', '?', '"', ' ', '.'. Также следует экранировать символы ',' и ';', чтобы отменить их специальное значение разделителя масок.

Примечание
Таблицы всегда восстанавливаются для текущего пользователя. Чтобы восстановить таблицы для схемы, требуется предварительно сделать текущей эту схему с помощью команды «SET SCHEMA».

Пример

См. описание ключа -only.

Сохранение метаданных базовых таблиц

`-otwd [<имя объекта 1>[,…]]`

Задает объект БД «Описание таблицы». Значение <имя объекта…> должно представлять имя таблицы или маску имен таблиц БД. При использовании этого ключа выполняется архивирование только структуры таблицы.

Сохранение информации о внешних ключах

`-oref`

Задает объект БД «Вторичные ключи» (имеется в виду foreign keys). Ключ используется для сохранения ссылочной целостности БД из архива. Он заставляет утилиту сохранять, помимо собственно данных архивируемой таблицы, и информацию обо всех ссылках этой таблицы на другие таблицы БД. Ключ имеет смысл только при использовании совместно либо с ключом `-ot`, либо `-otwd`.

Сохранение представлений БД

`-ov [<имя объекта 1>[,…]]`

Задает объект БД «Представления». Значение <имя объекта…> должно задавать имя представления или маску имен представлений БД.

Сохранение синонимов

`-os  [<имя объекта 1>[,…]]`

Задает объект БД «Синонимы». Значение <имя объекта…> должно представлять имя синонима или маску имен синонимов БД.

PUBLIC-синонимы нужно сохранять и восстанавливать отдельно. Для сохранения надо использовать ключ `-takeforeign`, при восстановлении необходимо указать, что восстанавливаться должны объекты пользователя PUBLIC.

Например:

```
lhb r -u SYSTEM/MANAGER8 -os MMM -only PUBLIC
```

Сохранение ролей

```
-or  [<имя объекта 1>[,…]]
```

Задает объект БД «Роли». Значение <имя объекта…> должно представлять имя роли или маску имен ролей БД.

Сохранение информации о владельцах роли

```
-osr  [<имя объекта 1>[,…]]
```

Задает объект БД «Назначения ролей». Значение <имя объекта…> должно представлять имя пользователя или маску имен пользователей БД, кому назначены роли.

Сохранение привилегий на базовые таблицы

```
-oa  [<имя объекта 1>[,…]]
```

Задает объект БД «Привилегии». Значение <имя объекта…> представляет имя таблицы или маску имен таблиц БД, для которых должны быть сохранены (восстановлены) привилегии доступа.

Сохранение привилегий на хранимые процедуры

```
-oap  [<имя объекта 1>[,…]]
```

Задает объект БД «Привилегии на хранимые процедуры». Значение <имя объекта… > представляет имя хранимой процедуры или маску имен хранимых процедур, для которых должны быть сохранены (восстановлены) привилегии доступа.

Сохранение зависимостей между объектами БД

```
-d
```

Задает режим сохранения:

- владельца сохраняемой таблицы;
- внешних ссылок (от сохраняемой таблицы);
- таблиц и их владельцев, на которые ссылается сохраняемая таблица;
- имя сервера в случае сохранения таблицы, расположенной на удаленном сервере.

Однако объекты, связанные по правилам ссылочной целостности, автоматически архивироваться не будут. Для этого надо использовать ключ `-oref`.

Примечания

1. Ключ `-d` совместно с ключом `-oall` игнорируется.
2. Данная версия утилиты не определяет объекты, от которых зависит представление (VIEW). В следующих версиях это ограничение будет устранено.

Сохранение объектов конкретного пользователя БД

`-only <имя пользователя>`

Сохранять объекты заданного пользователя. Значение <имя пользователя> должно представлять имя конкретного пользователя БД или маску имен пользователей БД.

Маска допускает 2 представления:

- маска заключается в двойные кавычки, не может содержать символа двойной кавычки, а символы '*' и '?' всегда трактуются как метасимволы;
- любой символ маски, следующий за экранирующим символом '\' теряет своё специальное значение. Это позволяет задавать маски, содержащие символы '*', '?', '"', ' ', '.'. Также следует экранировать символы ',' и ';', чтобы отменить их специальное значение разделителя масок.

Примечание
Ключ `-only` не действует на маску прав доступа пользователя, т.к. в маске доступа <Пользователь>.<Объект> поле <Пользователь> имеет значение получателя привилегий, а не владельца объекта. Владельцем объекта всегда является пользователь, выполнявший сохранение этого объекта.

Пример

```
SQL>username SYSTEM/MANAGER8
SQL>create table T (i int);
SQL>username ABC/12345678
SQL>create table T (i int);
SQL>grant select on T to SYSTEM;

lhb s -u SYSTEM/MANAGER8 -ot T
!сохранит SYSTEM.T

lhb s -u SYSTEM/MANAGER8 -ot T -takeforeign
или
lhb s -u SYSTEM/MANAGER8 -ot T -only
или
lhb s -u SYSTEM/MANAGER8 -ot T -only *
!сохранит SYSTEM.T, ABC.T

lhb s -u SYSTEM/MANAGER8 -ot T -only ABC
!сохранит ABC.T

lhb s -u SYSTEM/MANAGER8 -ot T -only ABC -takeforeign
!сохранит ABC.T

lhb s -u SYSTEM/MANAGER8 -ot SYSTEM.T,T -only ABC
!сохранит SYSTEM.T, ABC.T
```

 © Архивирование и восстановление базы данных. АО НПП «РЕЛЭКС», 1990-2026

```
lhb s -u SYSTEM/MANAGER8 -ot T -only SYSTEM,A*
!сохранит SYSTEM.T, ABC.T
```

Примечание
Из примеров видно, что во всех случаях вместо ключа `-takeforeign` можно использовать ключ `-only` (это справедливо и для ключа `-own` при восстановлении объектов БД).

Сохранение триггеров

`-otr [<имя тригг 1>[,…]]`

Задает объект БД «Триггеры». Значение <имя тригг…> должно представлять имя триггера или маску имен триггеров.

Сохранение хранимых процедур

`-op [<имя проц 1>[,…]]`

Задает объект БД «Хранимые процедуры». Значение <имя проц…> должно представлять имя хранимой процедуры или маску имен хранимых процедур.

Сохранение последовательностей

`-oq [<имя посл 1>[,…]]`

Задает объект БД «Последовательности». Значение <имя посл…> должно представлять имя последовательности или маску имен последовательностей.

Сохранение информации о серверах репликации

`-on [<имя сервера 1>[,…]]`

Задает объект БД «Сервер репликации». Значение <имя сервера…> должно представлять имя сервера репликации или маску имен серверов репликации.

Сохранение правил репликации

`-ob [<имя правила 1>[,…]]`

Задает объект БД «Правила репликации». Значение <имя правила…> должно представлять имя правила репликации или маску имен правил репликации.

Сохранение чужих объектов БД

`-takeforeign`

Позволяет сохранять чужие объекты (т.е. принадлежащие другим пользователям), к которым разрешен доступ на чтение.

Восстановление чужих объектов БД

`-own`

Позволяет восстанавливать как свои объекты (таблицы, представления, процедуры и триггеры), сохраненные другим пользователем. Например, если были предварительно сохранены чужие объекты по ключу `-takeforeign`, то использование ключа `-own` позволяет восстанавливать их в БД как свои. Следует с осторожностью пользоваться данным ключом для таких объектов, как представления, процедуры и триггеры, т.к. при восстановлении не производится проверки на существование зависящих от них объектов.

Ключи нарастающего архивирования

Ключи этой группы предназначены для управления нарастающим архивированием БД.

Начать нарастающее архивирование

`-startinc`

Начать режим нарастающего (инкрементного) архивирования.

Добавить порцию накопленных изменений

`-inc`

Добавить накопленные изменения.

Закончить нарастающее архивирование

`-stopinc`

Закончить режим нарастающего (инкрементного) архивирования в указанном архиве.

Создать новый архивный том

`-vi`

Создать новый архивный том.

Спецификация тома инкрементного архива

`-fo <спецификация файла>`

Указывает местоположение последнего тома инкрементного архива. Новый архивный том будет создаваться в каталоге, указанном в <спецификации файла>.

Например, первые тома инкрементного архива были скопированы на отдельный носитель и затем удалены с диска. В этом случае использование данного ключа позволяет продолжить инкрементное архивирование.

Ключи фразового индекса

Ключи этой группы предназначены для сохранения/восстановления фразовых индексов подсистемы полнотекстового поиска СУБД ЛИНТЕР.

Сохранение фразовых индексов

`-pi`

Сохранение в файле архива информации о фразовых индексах для возможности их последующего восстановления. Ключ применим только для полного сохранения БД (без инкрементного).

Примечание
Ключ действует по умолчанию.

Ключи поиска устройств при восстановлении БД

Ключи этой группы устанавливают порядок поиска устройства, если при восстановлении БД встречается файл, относящийся к устройству, отличному от заданного в переменной окружения SY00.

Примечание
Если устройство задано в переменной SY00, то оно игнорируется, и архив восстанавливается в текущий каталог.

Порядок поиска устройств при восстановлении БД

`-dp [0|1|2]`

Задает порядок поиска пути к устройству при восстановлении БД:

- `0` – искать сначала в переменных окружения, затем в списке устройств, сохраненных в файле архива (значение по умолчанию);
- `1` – искать сначала в списке устройств, сохраненных в файле архива, затем среди переменных окружения;
- `2` – восстанавливать все файлы БД в каталог по умолчанию.

Ключи приоритета

Ключи этой группы предназначены для определения приоритета работы `lhb`.

`-priority <уровень>`

Параметр <уровень> может принимать значения:

- `HIGH` – высокий приоритет (по умолчанию);
- `NORMAL` – средний приоритет;
- `LOW` – низкий приоритет.

Ключи просмотра и управления контрольными точками

При создании инкрементных архивов утилита использует механизм контрольных точек, суть которого в следующем:

- при запуске нарастающего архивирования с ключом `-startinc` утилита записывает в БД контрольную точку, содержащую необходимую для нарастающего

архивирования системную информацию (в частности, дату установки контрольной точки и параметры файлов системного журнала БД, от которых следует производить следующее инкрементное архивирование);

- после установки контрольной точки все файлы системного журнала БД от этой точки навсегда остаются в БД (если же контрольные точки не устанавливаются, то файлы системного журнала, содержащие зафиксированные в БД изменения, удаляются самой СУБД ЛИНТЕР). В связи с этим целесообразно производить удаление уже неиспользуемых старых контрольных точек (например, если начали еще один инкрементный архив и не хотим продолжать старый).

`-list`

Показать существующие в БД контрольные точки.

`-clear {<номер> | ALL}`

Очистить все контрольные точки (параметр ALL) или указанную в (<номере>) контрольную точку. Значение ключа является обязательным.

Примечание
Ключи контрольных точек используются совместно с командой `cp`.

Ключи для создания архива на магнитной ленте

Ключи этой группы предназначены для работы с архивами, хранящимися на магнитных лентах:

`-dev <имя ленты>`

Монтирование лентопротяжного устройства с системным именем <имя ленты>. Допускается одновременное монтирование нескольких устройств.

`-list`

Показать список файлов на ленте (ключ команды `tape`).

`-crpart <имя раздела>`

Создание нового раздела на магнитной ленте (ключ команды `tape`). Удаляет всю имеющуюся на ней информацию (при переиспользовании ленты) и создает новую метку ленты.

Ключи для просмотра сохраненных в архиве БД устройств

Ключи этой группы предназначены для работы с сохраненными в архиве устройствами.

`-ldev`

Показать список сохраненных в полном архиве БД устройств (это позволяет заранее определить необходимые переменные окружения для восстановления БД).

Ключ используется в команде `l`.

```
lhb l -ldev
```

Ключи обработки ошибок

Ключи этой группы управляют реакцией утилиты `lhb` на ситуации, которые могут возникнуть при её выполнении. Одновременно можно задавать несколько ключей. Самый высокий приоритет имеет ключ `-qq`, далее `-qc`, затем `-qx`. Если для однотипного события задано несколько ключей, то обработка события выполняется в соответствии со спецификацией высокоприоритетного ключа, остальные ключи в этом случае игнорируются.



Примечание
Если ключ задан без параметров, он игнорируется.

Утилита контролирует следующие события:

Событие	Описание
`NV`	Необходимо создание нового тома
`DF`	Создаваемый файл уже существует
`CB`	Поступил сигнал **<Ctrl>+<Break>**
`OB`	При восстановлении не совпала версия БД
`NB`	Слишком новая версия СУБД ЛИНТЕР
`UD`	Неизвестное устройство
`ALL`	В любом из вышеперечисленных случаев

`-qx [<событие 1> …]`

Прекратить работу при возникновении указанного события.

`-qc [<событие 1> …]`

Продолжать работу при возникновении указанного события. Нажатие комбинации клавиш **<Ctrl>+<Break>** игнорируется.

`-qq [<событие 1> …]`

Запросить пользователя о дальнейших действиях при возникновении указанного события.

Автоматизация архивирования БД

Архивирование БД является важной и наиболее часто выполняемой операцией администратора СУБД ЛИНТЕР. Для автоматизации этой работы утилита `lhb` имеет специальное средство – язык сценария архивирования, на котором можно написать командный файл (сценарий) для выполнения всех операций, необходимых при архивировании (восстановлении) конкретной БД. После того, как сценарий написан, отлажен и передан утилите `lhb` для исполнения, администратор БД может снять с себя все заботы о сохранности БД.

Сценарий архивирования содержит расписание и порядок выполнения утилитой `lhb` всего комплекса операций, связанных с ведением архива сложной информационной системы: архивирование БД, создание резервных копий, перенос архива или резервных копий на другие носители данных, удаление устаревших архивов и их копий и др.

При описании языка сценария используется синтаксическая нотация, основанная на формах Бэкуса-Наура (БНФ), со следующими расширениями:

- угловые скобки < > определяют строку, которая именует элемент командной строки (не терминальный символ);
- оператор определения ::= разделяет определяемый элемент (слева от оператора) и собственно его определение (справа от оператора);
- квадратные скобки [ ] указывают необязательный элемент командной строки или ключа. Часть определения, заключенная в эти скобки, может использоваться, как описано в конструкции, либо пропускаться;
- альтернативный оператор | указывает, что синтаксический элемент, следующий за этим символом, является одним из возможных вариантов конструкции;
- многоточие «…» указывает на допустимость повторения синтаксического элемента один или несколько раз внутри синтаксической конструкции;
- ключи командной строки выделены жирным шрифтом;
- пробелы используются для разделения синтаксических элементов.

Запуск сценария архивирования выполняется по команде `script`.

Примечание
Время старта автоматического архивирования в сценариях утилиты `lhb` задается по Гринвичу.

Структура сценария

Описание сценария может включать следующие разделы:

1) начальные установки;
2) описание переменных;
3) график выполнения;
4) специальные действия.

Раздел «Начальные установки»

Раздел «Начальные установки» является необязательным разделом сценария. Он должен включаться в том случае, если в сценарии предполагается использовать данные типа дата/время в формате, отличном от принятого по умолчанию в языке.

Установленный разработчиком сценария собственный формат входных данных типа «дата/время» запоминается в предопределенной переменной `tsformat` исходного текста сценария.

Примечание
Выходной формат данных типа «дата/время» устанавливается в предопределенной переменной `DATEFORMAT`.

Описание раздела

```
set:
[tsformat = "<входной формат даты/времени>"];
<входной формат даты/времени> ::= <строка формата>
<строка формата>::=
  <элемент формата> [<разделитель><элемент формата>…]
<разделитель>::=
  / | - | . | : | , | <символ пробела> | "<символьная константа>"
<элемент формата>::=
                <день месяца>
                | <день года>
                | <месяц>
                | <год>
                | <часы>
                | <минуты>
<день месяца>::= DD | dd
<день года>::= DDD | ddd
<месяц>::= MM | mm | MONTH | Month | month | MON | Mon | mon |
 MONR
<год>::= Y [Y [Y [Y]]] | y [y [y [y]]]
<часы>::= { HH | hh } [ {MID | mid } ]
      | { HH12 | hh12 }[ {MID | mid] }
      | { HH24 | hh24 }
<минуты>::= MI | mi
```

Описание шаблонов

Шаблоны `DD`, `dd` задают представление дня месяца арабскими цифрами. Диапазон значений этих шаблонов от 1 до 31.

Шаблоны `DDD`, `ddd` задают представление дня года арабскими цифрами. Диапазон значений этих шаблонов от 1 до 366.

Шаблоны `MM`, `mm` задают представление месяца арабскими цифрами. Диапазон значений этих шаблонов от 1 до 12.

Шаблоны `MONTH`, `Month`, `month` задают полное название месяца.

Шаблоны `MON`, `Mon`, `mon` задают сокращенное (три символа) название месяца.

Шаблон `MONR` задает представление месяца римскими цифрами. Допустимые значения этого шаблона: I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII.

Шаблоны `Y`, `y` задают представление цифр года арабскими цифрами. Диапазон значений этих шаблонов от 0 до 9.

Шаблоны `HH`, `hh`, `HH12`, `hh12` задают представление часа половины суток (до обеда или после обеда) арабскими цифрами. Диапазон значений этих шаблонов от 0 до 11.

Шаблоны `HH24`, `hh24` задают представление часа суток арабскими цифрами. Диапазон значений этих шаблонов от 0 до 23.

Шаблоны `MID`, `mid` задают представление символов am, pm.

Шаблоны `MI`, `mi` задают представление минут арабскими цифрами. Диапазон значений этих шаблонов от 0 до 59.

По умолчанию используется следующий формат:

```
"DD/MM/YYYY HH24:MI";
```

Примеры форматов и представляемых ими значений

Символьное представление	Формат
16.02.1998:10:35	DD.MM.YYYY:HH:MI
16.02.98:10:35	dd.mm.yy:HH:MI
16-02-98 10:35	DD-mm-YY hh:mi
10 час 30 мин 16/02/1998	HH "час" MI "мин" DD/MM/YYYY
В год 1998 месяца 02 дня 16	"В год" YYYY "месяца" MM "дня" DD
10.35	hh.mi
До конца года осталось дней: 264	"До конца года осталось дней: " DDD
10.VII.2001	DD.MONR.YYYY
15/07/1999 07pm:30	DD/MM/YYYY hh12mid:mi
Последняя архивация: март, 23.1998	"Последняя архивация: " MONTH, DD.YYYY

Раздел «Переменные»

Раздел «Переменные» является необязательным разделом сценария. Он должен включаться в том случае, если в исходном тексте сценария предполагается использовать собственные (отличные от предопределенных в языке сценария утилиты) переменные.

Язык сценария допускает два типа переменных – символьные и числовые. Тип переменной явно не объявляется и определяется по типу данных начального значения.

Описание раздела

```
variables:
[<определение переменной>];
[<определение переменной>];
……
[<определение переменной>];

<определение переменной>::=
<имя переменной> = <начальное значение>
```

 © Архивирование и восстановление базы данных. АО НПП «РЕЛЭКС», 1990-2026

```
<имя переменной>::= <идентификатор>
<идентификатор>::= алфавитно-цифровая строка длиной не более 18
 знаков
<начальное значение>::=
   "<символьная константа>"
   | <числовая константа>

<числовая константа>::=[ + | - ] <целое число>
```

Примечание
Символьная константа может включать коды \n, \t, \\, \" для представления символа «возврат каретки», «табуляция», непосредственно самого символа \ и символа ".

Примеры

```
FILE_DIR = "c:\linterSQL\db";

Name_File="";

Count = 25;
```

Раздел «График выполнения»

Раздел «График выполнения» содержит описание календарных дат и моментов времени, при наступлении которых утилита lhb должна выполнить действия, описанные в сценарии.

Описание раздела

```
rights:
[<ежечасная операция> [<ежечасная операция> … ]]
[<ежедневная операция> [<ежедневная операция> … ]]
[<еженедельная операция> [<еженедельная операция> … ]]
[<ежемесячная операция> [<ежемесячная операция> … ]]
[<в заданное время> [<в заданное время> … ]]
```

Примечание
Порядок описания операций может быть произвольным.

Для всех графиков выполнения действуют следующие правила:

1) если на одно и то же время запланировано несколько событий, то инициируется более редкое событие;
2) если для определённого правила время работы не позволяет запустить другие правила, то они приостанавливаются (с последующим запуском в порядке поступления), а то правило, которое не успело вовремя отработать, игнорируется (глубина игнорирования не превышает 1). Для этого ведется очередь отсортированных по времени выполнения операторов. Всегда выполняется первый элемент из этой очереди (при этом не важно, насколько давно прошло время его запуска – главное, чтобы момент запуска уже наступил). После выполнения оператора EVERYXXX он повторно ставится в очередь (при этом генерируется новое время запуска оператора) и т.д.

Пример

```
variables:
NUM = 1;
rights:
  everyweek (day=Tue,time='07:45')
  {
    NUM = NUM + 1;
    print ("Everyweek (200X)");
    backup ("s -u " + NAME + "/" + PASSWORD + " -f "+ FILENAME +
 TOSTR(NUM +
 2000) + ".lhb");
exception: /* Обработка ошибок оператора 'everyweek' */
    print ("Error=" + TOSTR(CERROR) + ",LinError=" +
 TOSTR(LINERROR) +
 ",SysError=" + TOSTR(SYSERROR));
    stop;
  }
  direct (timestamp='01.12.2016 07:45')
  {
    NUM = NUM + 1;
    print ("Direct (400X)");
    backup ("s -u " + NAME + "/" + PASSWORD + " -f "+ FILENAME +
 TOSTR(NUM +
 4000) + ".lhb");
exception: /* Обработка ошибок оператора 'direct' */
    print ("Error=" + TOSTR(CERROR) + ",LinError=" +
 TOSTR(LINERROR) +
 ",SysError=" + TOSTR(SYSERROR));
    stop;
  }
  everyday (time = '07:45')
  {
    NUM = NUM + 1;
    print ("Everyday (100X)");
    backup ("s -u " + NAME + "/" + PASSWORD + " -f "+ FILENAME +
 TOSTR(NUM +
 1000) + ".lhb");
exception: /* Обработка ошибок оператора 'everyday' */
    print ("Error=" + TOSTR(CERROR) + ",LinError=" +
 TOSTR(LINERROR) +
 ",SysError=" + TOSTR(SYSERROR));
    stop;
  }
  everyhour (min=45)
  {
    NUM = NUM + 1;
```

```
    print ("Everyhour (X)");
    backup ("s -u " + NAME + "/" + PASSWORD + " -f "+ FILENAME +
 TOSTR(NUM) +
 ".lhb");
exception: /* Обработка ошибок оператора 'everyhour' */
    print ("Error=" + TOSTR(CERROR) + ",LinError=" +
 TOSTR(LINERROR) +
 ",SysError=" + TOSTR(SYSERROR));
    stop;
  }
  everymonth (date=1, time='07:45')
  {
    NUM = NUM + 1;
    print ("Everymonth (300X)");
    backup ("s -u " + NAME + "/" + PASSWORD + " -f "+ FILENAME +
 TOSTR(NUM +
 3000) + ".lhb");
exception: /* Обработка ошибок оператора 'everymonth' */
    print ("Error=" + TOSTR(CERROR) + ",LinError=" +
 TOSTR(LINERROR) +
 ",SysError=" + TOSTR(SYSERROR));
    stop;
  }
special:
before /* just after the start */
{
  print ("Start backup system");
}
after /* after 'stop' or Ctrl-C */
{
  print ("Stop backup system");
  if (CERROR != 0)
    logprint ("Error present:" + TOSTR(CERROR));
}
iferr /* global */
{
  print ("Error=" + TOSTR(CERROR) + ",LinError=" + TOSTR(LINERROR)
 +
 ",SysError=" + TOSTR(SYSERROR));
  stop;
}
```

График выполнения ежечасных операций

Синтаксис

```
<ежечасная операция>::=
```

```
everyhour (min=<минуты часа>) <действия>
<минуты часа>::= <дата в формате MI>
<действия>::= {<блок операторов>}
```

Описание

Указанные <действия> будут выполняться на заданной минуте каждого часа суток. Если к указанному моменту времени утилита выполняет <действия>, заданные каким-либо другим оператором, то выполнение наступивших <действий> будет задержано до полного окончания предыдущих <действий>. В связи с этим при разработке сценария следует так планировать ежечасные <действия>, чтобы утилита успевала выполнять требуемую работу в заданных промежутках времени.

Отрицательное значение аргумента <минуты часа> определяет периодичность повторения заданного действия от начала часа, например,

`everyhour (min=-5)`
означает, что действие будет выполнено в .., 13:00, 13:05, 13:10, 13:15, .., 13:55, 14:00, ..

Примечание
Отрицательное значение аргумента <минуты часа> можно задавать только в виде литерала, а не переменной.

Если значение <минуты часа> не является делителем 60, то промежуток между последним вызовом в предыдущем часе и в начале следующего часа будет меньше этого значения, например, если задано `everyhour (min=-7)`, то событие вызовется в 13:56, а затем в 14:00.

Примечание (общее для всех операторов данного раздела)
При разработке сценария необходимо учитывать следующий важный момент. Если на некоторую календарно-временную дату совпало несколько действий (например, совпали по дате и времени ежедневные и еженедельные операции), то утилита выполнит запланированные действия только одного оператора. Другой при этом будет проигнорирован, причем нельзя определенно сказать, какой именно оператор из нескольких будет выполнен. Поэтому, чтобы избежать таких коллизий, графики выполнения планируемых событий должны отличаться друг от друга хотя бы на 1 минуту.

Пример

```
everyhour(min=59)
{
   /* Сохранение быстроменяющейся  таблицы */
   backup ("s -u SYSTEM/MANAGER8 -f c:\arc_db -ot Заказы");
exception:
   logprint(ctimestamp()+":\n");
   logprint("\tError=" + TOSTR(CERROR));
   logprint("\tLinError=" + TOSTR(LINERROR));
   logprint("\tSysError=" + TOSTR(SYSERROR));
   stop;
}
```

График выполнения ежедневных операций

Синтаксис

```
<ежедневная операция>::=
   everyday (time=<время суток>) <действия>
   <время суток>::= '< дата в формате HH24:MI>'
   <действия>::= {<блок операторов>}
```

Описание

Указанные <действия> будут выполняться каждый день в заданное время суток.

Пример

```
everyday(time='22:00')
{
   /* нарастающее архивирование каждый день */
   backup ("s -inc -u SYSTEM/MANAGER8 -f " + lastfilename);
}
```

График выполнения еженедельных операций

Синтаксис

```
<еженедельная операция>::=
   everyweek (day=<день недели>, time=<время суток>)<действия>
   <день недели>::= Mon | Tue | Wed | Thu | Fri | Sat | Sun
   <время суток>::= '< дата в формате HH24:MI>'
   <действия>::= {<блок операторов>}
```



Примечание
Mon – понедельник, Tue – вторник, Wed – среда, Thu – четверг, Fri – пятница, Sat – суббота, Sun – воскресение.

Описание

Указанные <действия> будут выполняться каждую неделю в заданный день недели и время суток.

Пример

```
everyweek(day=Mon,time='08:00')
{
   /* Создание нового архива */
   backup ("s -startinc -u SYSTEM/MANAGER8 -f c:\lhb\"+
           tostr(cdate())  + "-" +
           tostr(cmonth()) + "-" +
           tostr(cyear2()) + "\"");
exception:
   logprint(ctimestamp()+":\n");
```

```
   logprint("\tError="    + TOSTR(CERROR));
   logprint("\tLinError=" + TOSTR(LINERROR));
   logprint("\tSysError=" + TOSTR(SYSERROR));
   stop;
}
```

График выполнения ежемесячных операций

Синтаксис

```
<ежемесячная операция>::=
  everymonth (date=<день месяца>, time=<время суток>) <действия>
  <день месяца>::= <дата в формате DD > | lastdate
  <время суток>::= '<дата в формате HH24:MI>'
  <действия>::= {<блок операторов>}
     lastdate – последний день текущего месяца
```

Описание

Указанные <действия> будут выполняться каждый месяц в заданный день месяца и время суток.

Примечание
Для работы конструкции everymonth (date=..., time=...) необходимо присутствие в скрипте одной из конструкций вида `everyhour`, `everyday`, `everyweek`.

Примеры

```
everymonth (date=4, time='03:00')
```

```
everymonth (date=lastdate, time='23:59')
```

Выполнение в заданное время

Синтаксис

```
<в заданное время>::=
direct (timestamp=<дата>) <действия>
<дата> – строка даты/времени в соответствии с установленным в
 разделе set форматом (или с форматом по умолчанию – в случае
 отсутствия в сценарии этого раздела)
<действия>::= {<блок операторов>}
```

Описание

Указанные <действия> будут выполнены один раз при наступлении заданной даты.

Раздел «Специальные действия»

Раздел «Специальные действия» содержит описание действий, которые утилита `lhb` должна выполнять в определенные моменты своей работы и/или при возникновении

ошибочных ситуаций. Раздел не является обязательным, но если задается, то каждый оператор в нем должен быть представлен в единственном экземпляре.

Описание раздела

```
special:
   [<действия в начале работы>]
   [<действия в конце работы>]
   [<действия при ошибочной ситуации>]
```

Указанные <действия в начале работы> выполняются программой только в случае успешного запуска, т.е. при условии отсутствия синтаксических ошибок в тексте сценария или в командной строке.

Возможны два варианта выполнения <действий в начале работы> в зависимости от используемого сценария:

1) сценарий содержит раздел «График выполнения» с указанием циклически осуществляемых операций (ежечасных, ежедневных и т.п.). В этом случае утилита `lhb` должна функционировать в режиме 24 часа в сутки, 7 дней в неделю, т.е. теоретически «вечно», поэтому <действия в начале работы> (опять-таки теоретически) будут выполнены всего лишь один раз в жизни утилиты. Если это так, то необходимость в программировании и отладке <действий в начале работы> отпадает (один раз эти действия администратор БД может выполнить и вручную). Однако, как показывает практика, реальная жизнь утилиты весьма далека от теоретической модели. Выполнение утилиты может быть прервано по разным причинам: сбой в оборудовании, аварийное завершение утилиты по собственной инициативе из-за неустранимых ошибок (оператор `STOP`), прекращение работы по инициативе администратора БД (например, в связи с необходимостью изменения файла сценария) и др. Таким образом, в реальной жизни повторные запуски утилиты `lhb` являются скорее не исключением, а правилом. В этой ситуации необходимость в разработке <действий в начале работы> становится актуальной, и степень актуальности зависит от частоты перезапуска утилиты `lhb`.

   В <действиях в начале работы> можно выделить следующие обычно задаваемые функции:

   - подготовительные операции (например, проверка наличия достаточного дискового пространства для размещения архивного файла, создание необходимых каталогов, возможность монтирования лентопротяжного устройства в случае создания архива на магнитной ленте, инициирование нарастающего архивирования – создание открытого архива и т.п.);
   - проверка результатов предыдущего запуска утилиты `lhb` и устранение выявленных нарушений в архиве (тестирование, по крайней мере, последней копии архивного файла и повторное создание архива).

2) сценарий содержит раздел «График выполнения», но в разделе отсутствует описание циклически выполняемых операций (ежечасных, ежедневных и т.п.). В этом случае утилита `lhb` будет запускаться на выполнение всегда по инициативе администратора БД, и в <действия в начале работы> можно включить операции, которые обычно планируются в циклических операциях, например, удаление архивных файлов с истекшим сроком хранения.

Указанные <действия в конце работы> выполняются утилитой только в случае её корректного завершения – по команде **<CTRL>+<C>** или с помощью оператора `STOP` (инициатива самой утилиты `lhb`). Если работа утилиты прекращена из-за

отказа оборудования или операционной системы, то <действия в конце работы> не выполняются. Это свойство утилиты `lhb` может быть использовано, в частности, для определения способа завершения предыдущего запуска утилиты. Например, если в <действиях в начале работы> создать метку некоторого файла нулевой длины, а в <действиях в конце работы> удалить этот файл, то присутствие на диске указанного файла при запуске утилиты будет свидетельствовать о её предыдущем некорректном завершении.

Указанные <действия при ошибочной ситуации> определяют реакцию утилиты в случаях ошибочных ситуаций на глобальном уровне. На данный уровень обработки ошибок утилита выходит, если нет ни одного нижнего уровня, где выявленная ошибка могла быть обработка. Язык сценария предоставляет два способа обработки ошибок: с помощью конструкции `iferr`, относящейся непосредственно к оператору языка (раздел «Операторы языка сценария») и конструкции `exception`, объявляющей секцию обработки всех ошибок блока операторов, внутри которого расположена эта секция.

В утилите `lhb` предусмотрена следующая иерархия обработки ошибок:

1) если синтаксис оператора предусматривает конструкцию `iferr`, и эта конструкция задана, то в случае выявления ошибки при выполнении оператора она обрабатывается так, как это предусмотрено в конструкции `iferr` данного оператора;
2) если конструкция `iferr` не задана или не предусмотрена синтаксисом оператора, ошибка обрабатывается в секции `exception` текущего блока операторов;
3) если текущий блок операторов не содержит секцию обработки ошибок, утилита пытается найти эту секцию на более высоком уровне вложенности блоков операторов. Если на каком-нибудь уровне вложенности секция обработки ошибок будет найдена, то ошибка обрабатывается в соответствии с правилами этой секции;
4) если ни одной секции внутри блоков операторов не задано, ошибка будет обработана в глобальной секции (оператор `iferr` в разделе `special`);
5) если глобальная секция обработки ошибок не определена, то ошибка будет проигнорирована. Последствия распространения этой ошибки на процесс архивирования (восстановления) БД полностью перекладываются на разработчика сценария.

Действия в начале работы

Синтаксис

```
<действия в начале работы>::= before <действия>
<действия>::= { <блок операторов> }
```

Описание

Указанные <действия> выполняются в момент запуска утилиты `lhb` при условии отсутствия синтаксических ошибок в тексте сценария или в командной строке.

Действия в конце работы

Синтаксис

```
<действия в конце работы>::= after <действия>
<действия>::= { <блок операторов> }
```

Описание

Указанные <действия> выполняются при корректном завершении работы утилиты `lhb` (в том числе и при завершении по инициативе пользователя по команде **<Ctrl>+<C>**).

Действия при ошибочной ситуации

Синтаксис

```
<действия при ошибочной ситуации>::= iferr <действия>
<действия>::= { <блок операторов> }
```

Описание

Указанные <действия> выполняются в случае выявления ошибки, не обработанной в блоках операторов сценария.

Операторы языка сценария

Оператор выражения

Оператор выражения задает выражение строкового или числового типа. В таблице 3 представлены операции, допустимые для строковых операндов.

Таблица 3. Операции строковых операндов выражения

Операция	Название операции	Тип результата	Примечание
+	Конкатенация	Символьный	
=	Присваивание	Символьный	
==	Сравнение на равенство	Числовой	0 – операнды не равны 1 – операнды равны
!=	Сравнение на неравенство	Числовой	0 – операнды равны 1 – операнды не равны

В таблице 4 указаны допустимые операции для числовых операндов.

Таблица 4. Операции числовых операндов выражения

Операция	Название операции	Примечание
+	Сложение	
-	Вычитание	Бинарная операция
-	Изменение знака	Унарная операция
=	Присваивание	
==	Сравнение на равенство	0 – условие не выполнено 1 – выполнено
!=	Сравнение на неравенство	0 – условие не выполнено 1 – выполнено
>	Сравнение на больше	0 – условие не выполнено 1 – выполнено
<	Сравнение на меньше	0 – условие не выполнено 1 – выполнено
>=	Сравнение на больше или равно	0 – условие не выполнено 1 – выполнено
<=	Сравнение на меньше или равно	0 – условие не выполнено 1 – выполнено
AND	Логическое И	0 – условие не выполнено 1 – выполнено
OR	Логическое ИЛИ	0 – условие не выполнено 1 – выполнено
NOT	Логическое отрицание	0 – условие не выполнено 1 – выполнено

Комментарий

Комментарий может начинаться в любой позиции строки между синтаксическими элементами языка и должен располагаться в одной строке исходного текста сценария. Признаком начала и конца комментария является сочетание символов /* и */ соответственно.

Оператор цикла

Синтаксис

```
while (<условие цикла>)
{ <блок операторов> }
<условие цикла>::= выражение, результат вычисления которого имеет
 числовой тип.
```

Примечание
Если <блок операторов> состоит из одного оператора, скобки { , } можно не указывать.

Описание

В качестве <условия цикла> может использоваться любое выражение, результатом вычисления которого является значение числового типа.

<блок операторов> выполняется циклически до тех пор, пока значение <условия цикла> не равно нулю.

Пример

```
while(I)
  {
    I=I-1;
    /* нарастающее архивирование */
    backup ("s -inc -u" + myname + "/" + mypass + " -f " +
      "\"" + lastfilename + "\"");
    if(CERROR==0)
      logprint(ctimestamp() + ":" + ">>" + lastfilename);
  }
```

Оператор условия

Синтаксис

```
if (<условие выбора>)
{ <блок операторов 1> }
[ else
{ <блок операторов 2> } ]
<условие выбора>::= выражение, результат вычисления которого имеет
 числовой тип
```

Описание

Если значение <условия выбора> не равно нулю, выполняется <блок операторов 1>, в противном случае (если задана конструкция `else`) – <блок операторов 2>.

Примеры

1)

```
if (name == "") /* если пользователь не указал имя и пароль */
    {
    MYNAME = GETSTR("Введите имя:");
    MYPASS = GETPASS("Введите пароль:");
    }
  else
    {
    MYNAME = NAME;
    MYPASS = PASSWORD;
    }
```

2)

```
if (not exist (FILES_DIR))
  crdir (FILES_DIR); /* создать директорий для архива */
```

Оператор запуска утилиты lhb

Синтаксис

```
backup (<командная строка>);
[iferr{ <обработка ошибки> }
{ <блок операторов 2> } ]
<командная строка>::= <символьное выражение>
```

Описание

Оператор выполняет запуск утилиты `lhb` с командами и ключами, заданными в <командной строке>.

Конструкция `iferr` будет выполнена в случае возникновения ошибки. Если она не задана, то ошибка, выявленная при выполнении оператора, будет обработана в соответствии с иерархией обработки ошибок (см. пункт «Действия при ошибочной ситуации»).

Пример

```
backup ("s -startinc -u " + myname + "/" + mypass + " -f " +
  "\"" + FILES_DIR + "\\" +
  tostr(cdate()) + "-" + tostr(cmonth()) + "-" +
  tostr(cyear2()) + "\"");
```

 © Архивирование и восстановление базы данных. АО НПП «РЕЛЭКС», 1990-2026

Оператор копирования файла

Синтаксис

```
copy (<спецификация копируемого файла>, <спецификация копии>);
[ iferr{ < блок операторов > }]
<спецификация копируемого файла>::= <символьное выражение>
<спецификация копии>::= <символьное выражение>
```

Описание

Оператор выполняет копирование заданного файла. <спецификация копии> задает новое местоположение и имя файла. Имя файла в обеих спецификациях должно быть определено явно, использование знака «*» не допускается. Исходный файл остается без изменений.

Пример

```
copy("25-12-98.lhb","c:\arc_db\25-12-98.lhb");
```

Оператор перемещения файла

Синтаксис

```
move (<спецификация перемещаемого файла>,<спецификация
 перемещения>);
[iferr { <блок операторов> } ]
<спецификация перемещаемого файла>::= <символьное выражение>
<спецификация перемещения>::= <символьное выражение>
```

Описание

Оператор выполняет перемещение заданного файла. <спецификация перемещения> определяет новое местоположение и имя файла. Имя файла в обеих спецификациях должно быть задано явно, использование знака «*» не допускается.

Пример

```
move("25-12-98.lhb","../old/25-12-98.lhb");
```

Оператор переименования файла

Синтаксис

```
rename (<спецификация переименуемого файла>,<спецификация
 перенаименования>);
[iferr { <блок операторов> } ]
<спецификация переименуемого файла>::= <символьное выражение>
<спецификация перенаименования>::= <символьное выражение>
```

Описание

Оператор выполняет переименование заданного файла. <спецификация перенаименования> вводит новое имя и тип файла. Местоположение исходного файла не изменяется.

Пример

```
rename("c:\arc_db\25-12-98.lhb","25-12-98.arc");
```

Оператор удаления файла

Синтаксис

```
delete (<спецификация файла>);
[ iferr { <блок операторов> } ]
<спецификация файла>::= <символьное выражение>
```

Описание

Оператор выполняет удаление указанного файла. Имя файла в <спецификации файла> должно быть задано явно, использование знака «*» не допускается.

Пример

```
delete("../old/__24-12-98.lhb");
```

Оператор создания каталога

Синтаксис

```
crdir (<спецификация каталога>);
[ iferr { <блок операторов> } ]
<спецификация каталога>::= <символьное выражение>
```

Описание

Оператор выполняет создание указанного каталога.

Оператор синхронного запуска программы

Синтаксис

```
run (<спецификация программы>);
[ iferr { <блок операторов> } ]
<спецификация программы>::= <символьное выражение>
```

Описание

Оператор запускает на выполнение указанную программу, и утилита `lhb` переходит в ожидание завершения работы этой программы. <Спецификация программы> должна однозначно определять имя программы и её местонахождение. Если указано только имя программы, она будет искаться в соответствии с правилами ОС. Результат выполнения программы возвращается в предопределенной переменной `CERROR`. Если результат выполнения программы отличен от 0, то происходит генерация `exception` или выполнение секции `iferr` оператора.

Оператор асинхронного запуска программы

Синтаксис

```
exec (<спецификация программы>);
```

 © Архивирование и восстановление базы данных. АО НПП «РЕЛЭКС», 1990-2026

```
[ iferr { <блок операторов> } ]
<спецификация программы>::= <символьное выражение>
```

Описание

Оператор запускает на выполнение указанную программу, и утилита `lhb` продолжает свою работу. <Спецификация программы> должна однозначно определять имя программы и её местонахождение. Если путь к программе не прописан, она будет искаться в текущем каталоге. Результат выполнения программы не контролируется.

Оператор окончания программы

Синтаксис

```
stop
```

Описание

Оператор прекращает выполнение утилиты `lhb`.

Оператор прерывания обработки

Синтаксис

```
break
```

Описание

Оператор прекращает работу текущего оператора `if` или выполняет преждевременный выход из цикла `while`.

Оператор продолжения обработки

Синтаксис

```
continue
```

Описание

Оператор вызывает выполнение следующей итерации охватывающего цикла (`while`, `if`).

Оператор объявления секции обработки ошибок

Синтаксис

```
exception:
```

Описание

Оператор объявляет начало секции обработки ошибок текущего уровня вложенности.

Оператор перехода на обработку ошибки

Синтаксис

```
error
```

Описание

Оператор выполняет переход в секцию `exception` или в блок операторов `iferr`, соответствующий текущему оператору.

 © Архивирование и восстановление базы данных. АО НПП «РЕЛЭКС», 1990-2026

Предопределенные переменные и константы

В языке сценария реализованы следующие предопределенные переменные (переменные, использовать которые можно без их объявления и инициализации):

Имя переменной	Значение переменной
LASTFILENAME	Спецификация пути и имени (без расширения) последнего сохраненного архивного файла
LASTDATE	Дата последнего сохранения
CERROR	Код завершения утилиты `lhb` или программы, выполненной по оператору `run`. Этот код будет проверяться для вызова `iferr` (приложение может присваивать её собственное значение)
LINERROR	Код завершения, возвращенный СУБД ЛИНТЕР
SYSERROR	Код завершения, возвращенный операционной системой
NAME	Имя пользователя из командной строки
PASSWORD	Пароль пользователя из командной строки
FILENAME	Имя файла из командной строки
DATEFORMAT	Выходной формат данных типа «дата/время». Элементы формата аналогичны элементам формата в операторе `set: tsformat`

Предопределенные функции

Текущее время

Синтаксис

```
CTIMESTAMP()
Тип возвращаемого значения: символьный.
```

Описание

Функция возвращает текущую календарную дату в формате предопределенной переменной DATEFORMAT.

Для установки формата, отличного от используемого по умолчанию, необходимо присвоить переменной DATEFORMAT соответствующее значение, например:

```
DATEFORMAT="YYMMDD-HH24MI";
```

Номер года

Синтаксис

```
CYEAR2()
Тип возвращаемого значения: числовой.
```

Описание

Функция возвращает две последние цифры текущего года.

Полный номер года

Синтаксис

```
CYEAR4()
Тип возвращаемого значения: числовой.
```

Описание

Функция возвращает полный номер текущего года.

Номер месяца

Синтаксис

```
CMONTH ()
Тип возвращаемого значения: числовой.
```

Описание

Функция возвращает номер текущего месяца (число в диапазоне от 1 до 12).

Номер дня недели

Синтаксис

```
CDAY()
Тип возвращаемого значения: числовой.
```

Описание

Функция возвращает номер текущего дня недели (число в диапазоне от 1 до 7, нумерация начинается с понедельника).

Название дня недели

Синтаксис

```
CWEEKDAY ()
Тип возвращаемого значения: символьный.
```

Описание

Функция возвращает первые три символа названия текущего дня недели. Названия выдаются в англоязычном виде, например:

```
Mon, Tue, …
```

Номер дня месяца

Синтаксис

```
CDATE()
Тип возвращаемого значения: числовой.
```

Описание

Функция возвращает номер текущего дня месяца (число в диапазоне от 1 до 31).

Номер текущего часа

Синтаксис

```
CHOUR ()
Тип возвращаемого значения: числовой.
```

Описание

Функция возвращает номер текущего часа (число в диапазоне от 0 до 23).

Номер текущей минуты

Синтаксис

```
CMIN ()
Тип возвращаемого значения: числовой.
```

Описание

Функция возвращает номер минуты текущего часа (число в диапазоне от 0 до 59).

Информация о файле

Синтаксис

```
EXIST(<спецификация файла>)
<спецификация файла>::= <символьное выражение>
Тип возвращаемого значения: числовой.
```

Описание

Функция возвращает информацию о существовании указанного файла: `0` – файл не существует, `1` – файл существует.

Выдача сообщения на консоль

Синтаксис

```
PRINT (<сообщение >)
<сообщение>::= <символьное выражение>
Тип возвращаемого значения: числовой.
```

Описание

Функция посылает на консоль (устройство, с которого запущена утилита `lhb`) текст указанного сообщения. Всегда возвращается значение `1`.

Запись сообщения в файл протокола

Синтаксис

```
LOGPRINT(<сообщение >)
<сообщение>::= <символьное выражение>
Тип возвращаемого значения: числовой.
```

Описание

Функция помещает в файл протокола утилиты указанное сообщение. Если файл протокола не ведется, функция игнорируется. В любом случае возвращается значение `1`.

Преобразование числа в строку

Синтаксис

```
TOSTR(<числовое выражение>)
Тип возвращаемого значения: символьный.
```

Описание

Функция преобразует числовое выражение в символьное представление.

Ввод строки с консоли

Синтаксис

```
GETSTR(<символьная переменная>)
Тип возвращаемого значения: символьный.
```

Описание

Функция осуществляет ввод с консоли символьного текста. Перед вводом данных на консоль выдается подсказка, текст которой содержится в параметре <символьная переменная> вызова функции. Максимальная длина вводимого текста равна 200 символам, признаком конца ввода является переход на новую строку.

Пример

```
GETSTR("lhb>Имя файла архива: ")
```

Ввод пароля

Синтаксис

```
GETPASS (<символьная переменная>)
Тип возвращаемого значения: символьный.
```

Описание

Функция осуществляет ввод с консоли символьного текста, представляющего пароль. Вводимые символы визуально отображаются в виде «****». Перед вводом данных на консоль выдается подсказка, текст которой содержится в параметре <символьная переменная> вызова функции. Максимальная длина вводимого текста равна 200 символам, признаком конца ввода является переход на новую строку.

Пример

```
GETPASS("lhb>Пароль ?: ")
```

Значение переменной окружения

Синтаксис

```
GETENV (<имя переменной>)
<имя переменной>::= <символьное выражение>
Тип возвращаемого значения: символьный.
```

Описание

Функция возвращает значение заданной переменной окружения, если она существует, и пустую строку – в противном случае.

Пример

```
GETENV ("SY00")
```

Пример сценария архивирования

```
//
```

```
// Пример сценария архивации
// -------------------------
variables:
NUM = 1;
rights:
everyday (time = '12:00')
   {
   if (CWEEKDAY() =="Sun") /* Новый архив создавать в воскресенье
 */
      {
      move (FILENAME + TOSTR(NUM) + ".lhb", "c:\arc");
      NUM = NUM + 1;
      backup ("s -u " + NAME + "/" + PASSWORD + " -f " +
        FILENAME + TOSTR(NUM) + ".lhb -startinc");
      }
   else { /* В остальные дни ведется нарастающее архивирование */
   backup ("s -u " + NAME + "/" + PASSWORD + " -f " +
     FILENAME + TOSTR(NUM) + ".lhb -inc");
   } /* end of 'if' */
   exception: /* Обработка ошибок оператора 'everyday' */
print ("Error=" + TOSTR(CERROR) +
      ",LinError=" + TOSTR(LINERROR) +
      ",SysError=" + TOSTR(SYSERROR));
stop;
   }

special:
before /* just after the start */
   {
   print ("Start backup system");
backup ("s -u " + NAME + "/" + PASSWORD + " -f " +
  FILENAME + TOSTR(NUM) + ".lhb -startinc");
}
after /* after 'stop' or Ctrl-C */
   {
   print ("Stop backup system");
   if (ERROR != 0)
      logprint ("Error present:" + TOSTR(CERROR));
}
iferr /* global */
{
print ("Error=" + TOSTR(CERROR) +
      ",LinError=" + TOSTR(LINERROR) +
      ",SysError=" + TOSTR(SYSERROR));
stop;
}
```

Особенности архивирования при параллельной работе с пользователями БД

Утилита `lhb` позволяет выполнять процесс архивирования (восстановления) данных параллельно с работой пользователей БД. Однако при этом следует учитывать следующие важные моменты, которые влияют на степень полноты получаемого архива и, следовательно, на целесообразность использования этого способа архивирования. В каждом конкретном случае вопрос должен рассматриваться администратором БД индивидуально.

В процессе выборочного архивирования утилита не возвращается к повторному архивированию уже сохраненного и впоследствии измененного до завершения процесса архивирования объекта (то есть режим «догона» не выполняется). Поэтому, если после архивирования некоторого объекта и до окончания выполнения утилиты этот объект был изменен другими пользователями БД в процессе параллельной работы, новые данные в архив не попадут и, следовательно, при последующем восстановлении будут потеряны.

При полном архивировании БД в параллельном режиме данные будут сохранены полностью.

Если после окончания выборочного или полного архивирования в БД остались некоторые незавершенные транзакции, то при восстановлении из такого архива данные незавершенных транзакций будут потеряны.

Если в процессе выборочного архивирования операция `COMMIT` для некоторой транзакции была выполнена при активной утилите `lhb` после архивирования объектов этой транзакции и после архивирования системного журнала, то при восстановлении из такого архива изменения, выполненные этой транзакцией, не попадут в файл архива.

Сообщения утилиты lhb

Структура протокола выполнения

Структура диагностического сообщения утилиты:

```
LINTER HOT BACKUP PROGRAM – v.4.1.0 build 893 -- Relex,Inc.
 1990-2013[1]
<строка протокола выполнения>
…
<строка протокола выполнения>[2]
----------------------------------------
lhb return code (<код завершения lhb>) <текст сообщения lhb>[3]
Linter return code (<код завершения СУБД ЛИНТЕР>) <текст
 сообщения СУБД ЛИНТЕР>[4]
System return code (код завершения ОС) <текст сообщения ОС>[5]
```

Коды завершения и соответствующие тексты сообщений утилиты `lhb` приведены в приложении 1 данного документа.

Коды завершения и соответствующие тексты сообщений СУБД ЛИНТЕР приведены в документе «Справочник кодов завершения».

Коды завершения и соответствующие тексты сообщений операционной системы приведены в эксплуатационной документации на операционную систему. Если операционная система русифицирована, текст сообщения будет на русском языке, в противном случае – на языке операционной системы.

Примеры

1)

```
LINTER HOT BACKUP PROGRAM-v.4.1.0 build 893 -Relex,Inc. 1990-2013
Ошибка соединения. (1001)
-------------------------------------------
lhb return code (30111) - Ошибка соединения с БД
Linter return code ( 1001)
```

2)

```
LINTER HOT BACKUP PROGRAM-v.4.1.0 build 893 -Relex,Inc. 1990-2013
Файл qqq.lhb
Не могу открыть файл 'qqq'
-----------------------------------------------
```

[1]Строка 1 содержит идентификационные сведения об утилите `lhb`.

[2]Строки 2 содержат протокольные сообщения утилиты `lhb`.

[3]Строка 3 содержит код завершения и соответствующий диагностический текст последней операции, которая выполнялась `lhb` и закончилась неудачно. Описание самой операции находится в последней строке протокола.

[4]Строка 4 содержит код завершения, сформированный СУБД ЛИНТЕР.

[5]Строка 5 содержит код завершения, сформированный операционной системой.

```
lhb return code (30105) - Ошибка открытия файла
System return code (    1) - No such file or directory
```

Протокольные сообщения

Утилита `lhb` выдает следующие протокольные сообщения:

1) сообщение «`Ошибка соединения`»

   Причина.

   Утилита архивирования не смогла установить доступ к БД. Возможны следующие причины:
   - БД (локальная или удаленная), которая должна архивироваться (или частично восстанавливаться), не активна;
   - было задано неправильное имя удаленного ЛИНТЕР-сервера;
   - ошибка сетевого доступа к удаленной БД (например, не загружен один из сетевых драйверов).

   Устранение:
   - активизировать необходимую БД и повторить выполнение утилиты;
   - проверить (уточнить у администратора локальной сети) правильность заданного имени ЛИНТЕР-сервера;
   - обратиться к администратору БД или локальной сети для устранения причины отказа сетевого доступа.

2) сообщение «`Проверка пароля: строки не идентичны!`»

   Причина.

   В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация закодированного архивного файла, при этом введенный пароль и его подтверждение не совпали.

   Устранение:
   - указать правильно пароль и его подтверждение (учитывая различие больших и малых букв).

3) сообщение «`Не могу открыть файл <имя файла>`»

   Причина.

   В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация несуществующего файла, или пользователь утилиты не обладает необходимыми полномочиями для доступа к архивному файлу.

   Устранение:
   - задать правильную спецификацию (путь и имя) архивного файла, установить необходимые полномочия пользователю утилиты или изменить атрибуты доступа (в операционной системе) к архивному файлу.

4) сообщение «`Неверный архив <имя файла>`»

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация архивного файла, который в действительности не является архивом БД.

Устранение:

- указать правильную спецификацию архивного файла.

5) сообщение «`Слишком старая версия файла <имя файла>`»

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация архивного файла, который создан более ранней версией утилиты `lhb`.

Устранение:

- для работы с этим архивным файлом использовать соответствующую ему версию утилиты.

6) сообщение «`Слишком старая версия lhb`»

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация архивного файла, который создан более поздней версией утилиты `lhb`.

Устранение:

- для работы с этим архивным файлом использовать соответствующую ему версию утилиты.

7) сообщение «`Неверный пароль`»

Причина.

В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация закодированного архивного файла, и при запросе пароля был введен неверный пароль.

Устранение:

- попытаться вспомнить (восстановить) пароль. Если восстановить пароль невозможно, архив потерян навсегда.

8) сообщение «`Начать следующий том?`»

Причина.

Для команды сохранения был задан ключ «Запрашивать разрешение на создание тома» (ключ `-qq`, значение ключа «Новый том» (NV)).

Устранение:

- ответить **ДА**, если очередной том многотомного архива нужно создать. При ответе **НЕТ** процесс архивирования закончится.

9) сообщение «`Недопустимый формат даты <введенная дата>, должен быть 'DD.MM.[YY]YY.HH:MI'`»

   Причина.

   В файле сценария строка, задающая дату и время очередного сохранения БД, имеет неправильный формат.

   Устранение:

   - откорректировать файл сценария сохранения БД.

10) сообщение «`Ошибка сравнения контрольной суммы!`»

    Причина.

    В командах восстановления, тестирования или получения информации об архивном файле была задана спецификация архивного файла, целостность которого нарушена.

    Устранение:

    - ситуация критическая – восстановление из испорченного архива невозможно. Если имеется дубликат архива, использовать его.

11) сообщение «`Файл уже существует. Удалить его?`»

    Причины:

    - в команде сохранения БД была задана спецификация существующего архивного файла;
    - в команде восстановления БД была задана спецификация существующей БД.

    Устранение:

    - при ответе **ДА** существующий файл будет заменен новым, при ответе **НЕТ** выполнение утилиты прекратится.

Оперативное архивирование

Полное или нарастающее архивирование БД можно выполнять в оперативном режиме, параллельно с обслуживанием ядром СУБД запросов пользователей. Управление оперативным архивированием осуществляется с помощью соответствующих SQL-запросов. Такое архивирование очень эффективно с точки зрения удаленного администрирования СУБД ЛИНТЕР или при отсутствии возможности использовать утилиту `lhb`. Оперативное восстановление БД ядром СУБД невозможно – для этого необходимо использовать утилиту `lhb`. Файлы архива, создаваемые утилитой `lhb` и ядром СУБД при оперативном архивировании, полностью совместимы между собой.

SQL-запросы оперативного архивирования

Начать оперативное архивирование

Функция

Определение оператора запуска процесса архивирования БД в оперативном режиме.

Спецификация

```
<начать архивирование>::=
  BACKUP DATABASE [[START | STOP] INCREMENT]
  [DEVICE <имя устройства>]
  [FILE <спецификация файла> [REWRITE]]
  [COMMENT <комментарий>]
  [PASSWORD <пароль>]
  [VOLUMES <размер тома> [K|M]]
  [ASYNC];

<имя устройства>::= 4-х символьный литерал
<спецификация файла>::= символьный литерал
<размер тома>::= целочисленный литерал
```

Синтаксические правила

1) Опция `DATABASE` задает полное сохранение БД без возможности нарастающего архивирования;
2) <спецификация файла> задает полную спецификацию (устройство, путь к каталогу и имя) архивного файла. Если тип файла не указан, по умолчанию используется `.lhb`.

   Пример для ОС типа Linux, ЗОСРВ Нейтрино:

```
BACKUP DATABASE FILE 'SY00/ARCHIVES/database.lhb';
BACKUP DATABASE FILE '/../db.lhb';
```

3) Фраза `REWRITE` разрешает удалять существующий архивный файл и создавать новый с тем же именем;
4) Конструкция `DATABASE START INCREMENT` задает полное сохранение БД с возможностью нарастающего архивирования;
5) Конструкция `DATABASE INCREMENT` – сохранение всех изменений в БД со времени последнего нарастающего сохранения (от последней контрольной точки, созданной, например, с помощью `DATABASE START INCREMENT`);

6) Конструкция DATABASE STOP INCREMENT прекращает ведение нарастающего архивирования: последняя контрольная точка удаляется, а архивный файл закрывается. Так как по этому SQL-запросу сохранение в архивном файле последней порции накопленных в БД изменений (то есть начиная с последней контрольной точки) не осуществляется, то перед его выполнением необходимо подать SQL-запрос BACKUP DATABASE INCREMENT для исключения потери архивных данных;
7) Опция DEVICE задает логическое имя устройства, на котором должен создаваться файл архива. Устройство должно быть описано в системной таблице $$$DEVICE. Если имя устройства не задано, по умолчанию используется переменная окружения SY00. Если SY00 не определена, архив создается в каталоге запуска ядра СУБД ЛИНТЕР.

Пример:

```
BACKUP DATABASE DEVICE 'SY02' FILE 'h:\linter\arhiv\sale2804.lhb';
```

8) <комментарий> задает текст комментария к архивному файлу;
9) <пароль> задает пароль архивного файла;
10) Конструкция VOLUMES <размер тома> [K|M] разрешает разбивать архивный файл на отдельные тома указанного <размера тома>: K – в Кбайтах, M – Мбайтах;
11) Опция ASYNC задает асинхронное выполнение процесса архивирования.

Общие правила

Примечания

1. По возможности использовать для архивирования БД утилиту lhb, а не SQL-запрос BACKUP DATABASE.
2. Не использовать для архивирования БД SQL-запрос BACKUP DATABASE, если в БД есть фразовые индексы.

Если задана опция ASYNC, пользователю сразу будет возвращен код завершения, указывающий на результат выполнения SQL-запроса (успешный или неуспешный запуск процесса архивирования). В дальнейшем информацию о текущем состоянии процесса архивирования можно будет получать из системной таблицы $$$INKERNBACK (пункт «Мониторинг процессов асинхронного архивирования»).

Примеры запуска оперативного архивирования

Пример 1

```
BACKUP DATABASE ASYNC FILE 'DB_Sale.lhb' REWRITE COMMENT
 'Оперативный архив от 22.12.2005';
```

Пояснение

Будет запущен процесс оперативного архивирования всей БД в архивный файл «DB_Sale.lhb» (если такой файл уже имеется на диске – он будет предварительно удален без предупреждения), архивный файл будет иметь комментарий «Оперативный архив от 22.12.2005». Процесс будет проходить в асинхронном режиме, т.е. пользователь сразу получит код завершения (0 – если процесс запустился). В дальнейшем процесс архивирования будет протекать независимо от того, работает ли пользователь, запустивший процесс, с БД или нет. Рекомендуется также получить и запомнить идентификатор запущенного процесса для последующего контроля за его состоянием.

Пример 2

```
BACKUP DATABASE FILE 'DB_Sale.lhb' PASSWORD '221205';
```

Пояснение

Будет запущен процесс синхронного архивирования в архивный файл «DB_Sale.lhb». Если заданный архивный файл уже существует, пользователю будет возвращен соответствующий код завершения, и процесс архивирования завершится; в противном случае процесс архивирования начнется, и пользователь (или клиентское приложение) перейдет в состояние ожидания завершения этого процесса. После окончания процесса архивирования (удачного или неудачного) пользователь получит соответствующий код завершения. Архивный файл будет защищен паролем '221205'.

Пример 3

```
BACKUP DATABASE START INCREMENT FILE 'inc.lhb' VOLUMES 100k;
```

Пояснение

Будет выполнено полное сохранение БД для последующего нарастающего архивирования и в системном журнале установлена контрольная точка, начиная с которой необходимо наращивать инкрементный архив. Файл архива «inc.lhb» будет разделяться на тома размером по 100 Кбайт.

Примечание
Если необходимо указать размер в мегабайтах, то следует задавать постфикс «м», например, 1м; если просто в байтах, то постфикс не задается, например, 2000.

Пример 4

```
BACKUP DATABASE INCREMENT FILE 'inc.lhb' VOLUMES 100k;
```

Пояснение

Будет проведена проверка существования в БД контрольной точки нарастающего архивирования, соответствующей файлу архива inc.lhb, с последующим внесением в файл архива накопленных от этой точки изменений. Файлы архива по-прежнему будут разбиваться на тома указанного объема.

Пример 5

```
BACKUP DATABASE STOP INCREMENT FILE 'inc.lhb';
```

Пояснение

Из системного журнала БД будет удалена последняя контрольная точка, соответствующая файлу архива inc.lhb; никакие данные в файл архива не помещаются, процесс архивирования заканчивается с выдачей соответствующего кода завершения.

Остановить оперативное архивирование

Функция

Определение оператора прерывания асинхронного архивирования БД в оперативном режиме.

Спецификация

```
<остановить архивирование>::=
BACKUP STOP {ALL | SINCE <дата>}
[UNTIL <дата>] | <номер процесса>;
<дата>::= дата-время литерал
<номер процесса>::= целочисленный литерал
```

Синтаксические правила

1) Опция ALL задает останов всех процессов архивирования (как тех, которые запущены пользователем, подавшим эту команду, так и тех, которые инициированы другими пользователями БД);
2) Конструкция SINCE <дата> задает останов процессов архивирования, активизированных после указанной <даты>;
3) Конструкция UNTIL <дата> задает останов процессов архивирования, активизированных до указанной <даты>;
4) <Номер процесса> – системный идентификатор процесса архивирования. <Номер процесса> возвращается в поле ROWID блока управления запросом CBL при использовании интерфейса нижнего уровня (см. документ «Интерфейс нижнего уровня») или в переменной ROWIDPCI_ при использовании встроенного SQL (см. документ «Встроенный SQL») в случае успешного запуска процесса архивирования.

Общие правила

1) Прервать можно только асинхронное архивирование;
2) Прервать процесс архивирования может либо пользователь, который его инициировал, либо пользователь с правами DBA (права DBA позволяют прерывать любые активные процессы архивирования, в том числе и других пользователей).

Примеры останова оперативного архивирования

Пример 1

```
BACKUP STOP 4;
```

Пояснение

Будет произведена попытка останова процесса оперативного архивирования, имеющего идентификатор 4. Сначала ядро СУБД ЛИНТЕР проверит активность указанного процесса (если он уже завершился – будет выдан соответствующий код завершения). Если же процесс оперативного архивирования активен, то выполняются следующие действия:

- закрытие файла архива;
- запись в системную таблицу $$$INKERNBACK информации о причине и стадии (в процентах) завершения процесса;
- останов процесса архивирования.

Пример 2

```
BACKUP STOP ALL;
```

Пояснение

Будет произведена попытка останова всех запущенных в асинхронном режиме процессов оперативного архивирования, в том числе и процессов, запущенных другими пользователями СУБД с правами DBA. В случае успешного выполнения запроса пользователю, подавшему запрос, будет возвращено количество остановленных процессов в поле `ROWID` блока управления запросом CBL при использовании интерфейса нижнего уровня (см. документ «Интерфейс нижнего уровня») или в переменной `ROWIDPCI_` при использовании встроенного SQL (см. документ «Встроенный SQL»). Все остановленные процессы завершат свою работу, как описано выше в примере 1 этого раздела.

Пример 3

```
BACKUP STOP ALL SINCE '09.07.2002:12:00' UNTIL '09.07.2002:14:00';
```

Пояснение

Будет произведена попытка останова всех запущенных в указанном диапазоне времени асинхронных процессов оперативного архивирования, в том числе и процессов, запущенных другими пользователями СУБД ЛИНТЕР с правами DBA: будут остановлены процессы, у которых время запуска попадает в интервал между 12:00 и 14:00 девятого июля 2002 года.

Механизм асинхронного архивирования

Асинхронное оперативное архивирование выполняется следующим образом:

- пользователь подает SQL-запрос на запуск оперативного асинхронного архивирования БД и сразу получает от ядра СУБД код завершения об успешном/неуспешном старте процесса;
- запуск последующих процессов асинхронного архивирования можно выполнять по одному и тому же соединению с БД, не дожидаясь завершения предыдущих процессов;
- в случае успешного старта процесса асинхронного архивирования СУБД сама выделяет ему отдельный канал для работы с БД;
- в любом случае (как в случае успешного окончания процесса архивирования, так и неуспешного) этот канал будет корректно закрыт (освобожден) ядром СУБД;
- взаимодействие с выделенным процессу архивирования каналом выполняет ядро СУБД при обработке SQL-запросов `BACKUP STOP`;

Пользователь БД имеет возможность в любой момент времени получить информацию о ходе выполнения асинхронного архивирования путем выборки нужных данных из системной таблицы `$$$INKERNBACK` (например, запросом `SELECT * FROM $$$INKERNBACK`). При запуске процесса архивирования в эту таблицу добавляется строка, соответствующая данному процессу, которая в течение архивирования будет периодически модифицироваться для отражения текущего статуса процесса.

- в поле `Status` будет помещено значение в процентах (от 0 до 100) о процессе архивации.

Структура системной таблицы `$$$INKERNBACK`:

```
CREATE TABLE $$$INKERNBACK (
```

 © Архивирование и восстановление базы данных. АО НПП «РЕЛЭКС», 1990-2026

```
BackId          INTEGER,
UserId          INTEGER,
ChannelId       INTEGER,
LinterRetCode   INTEGER,
Status          INTEGER,
StartTime       DATE,
EndTime         DATE);
```

Описание столбцов таблицы:

Столбец	Описание
BackId	Идентификатор процесса архивирования
UserId	Идентификатор пользователя, подавшего SQL-запрос на оперативное архивирование
ChannelId	Идентификатор канала, открытого для процесса архивирования (канал, который выделяет само ядро СУБД для порожденного процесса)
LinterRetCode	Код завершения процесса архивирования (приложение 1)
Status	Число в процентах, отражающее степень завершения процесса архивирования
StartTime	Дата-время начала процесса архивирования
EndTime	Дата-время окончания процесса архивирования

Если эта таблица существует, то добавляется соответствующая строчка, содержащая необходимые данные. Наличие таблицы проверяется на этапе начального запуска `in-kernel backup` в асинхронном режиме. В случае её отсутствия будет выведена соответствующая ошибка, и процесс не запустится. Если таблица существует, то в нее добавляется строчка, содержащая `BackId`, `UserId`, `ChannelId`, `LinterRetCode` равный нулю, т.к. пока он не известен, `StartTime` и `EndTime`, где `StartTime` будет равно `EndTime` и соответствовать дате-времени старта процесса. По окончании процесса архивации значения `LinterRetCode`, `Status` и `EndTime`, возможно, изменятся в соответствии с действительным кодом завершения процесса, состоянием в процентах и реальным окончанием времени завершения процесса.

Все состояния завершения не асинхронных процессов в эту таблицу попадать не будут, т.к. пользователь получает необходимые данные по окончании процесса архивации.

Мониторинг процессов асинхронного архивирования

Пользователь может в любой момент получить информацию о состоянии процесса оперативного архивирования в статусе прорезервного сохранения путем SQL-запросов:

```
1 SELECT * FROM $$$INKERNBACK;
```

(всех процессов резервного сохранения, запущенных когда-либо)

```
2 SELECT * FROM $$$INKERNBACK WHERE BackId = number;
```

(конкретного процесса, где `number` – идентификатор процесса, возвращенный после успешного старта оного).

Возвращенным значением станут данные из этой таблицы, где наиболее интересным будет значение поля `Status`. Для него возможны следующие варианты:

- от `0` до `100` – статус, в процентах, работы процесса резервного сохранения;
- `100` – завершение процесса (если значение поля `LinterRetCode` равно `0`, то успешное);
- `-1` – в случае некорректной или неудавшейся попытки получить статус работающего или завершившегося процесса.

Коды завершения СУБД ЛИНТЕР при работе резервного сохранения

Код	Описание
0	Успешное завершение
6900	Неправильный параметр
6901	LHB процесс активен
6902	LHB процесс не активен
6903	Неправильный объект (отношение, пользователь, последовательность, процедура ...)
6904	Неправильный номер атрибута/таблицы
6906	Не хватает памяти для канала
6907	LHB не стартовал
6908	Нарастающее архивирование: не найден конец в системном протоколе архивирования
6910	Таблица для триггеров не существует
6911	Процедура уже существует
6912	Триггер уже существует
6913	Невозможно добавить тело процедуры
6916	Невозможно прочитать данные из таблицы `SERVER`
6917	Невозможно прочесть таблицу `$$$REPL`
6918	Невозможно восстановить сервер
6919	Не удается восстановить правило репликации
6920	Сервер для правила репликации не существует
6931	Таблица `$$$INKERNBACK` отсутствует в БД
6932	Нарушение прав доступа к архивному файлу
6933	Не могу найти файл архива
6934	Превышен предел количества одновременно открытых файлов
6935	Файл архива уже существует
6936	Невозможно открыть файл архива
6937	Невозможно удалить файл архива
6938	Невозможно произвести запись в файл архива
6939	Асинхронный процесс архивации остановлен пользователем
6940	Запрошен останов конкретного процесса – не могу найти указанный процесс

Код	Описание
6941	У пользователя нет привилегий DBA
6942	Ошибка чтения из таблицы `$$$INKERNBACK`
6943	Асинхронный процесс архивации уже остановлен
6944	Невозможно прочесть файл архива
6945	Некорректные инкрементные данные в файле архива
6946	Некорректное имя архивного файла
6947	Некорректное имя устройства
6948	Событие принадлежит другому пользователю

Приложение 1
Коды завершения утилиты lhb

Код	Причина
30100	Не могу найти начало ленты
30101	Лента не читается
30102	Ошибка открытия ленты
30103	Ошибка закрытия ленты
30104	Ошибка записи на ленту
30105	Ошибка открытия файла
30106	Нажат Ctrl-Break
30107	Ошибка CRC
30108	Ошибка чтения файла
30109	Ошибка записи в файл
30110	Ошибка удаления файла
30111	Ошибка соединения с БД
30112	Ошибка начала сохранения
30113	Не могу остановить процесс добавления
30114	Не найдена информация о добавлении
30115	Не найдена информация о добавлении
30116	Ошибка запрещения переиспользования системного журнала
30117	Ошибка разрешения переиспользования системного журнала
30118	Ошибка сохранения пользователя
30119	Ошибка сохранения представления
30120	Ошибка сохранения синонима
30121	Ошибка сохранения роли
30122	Ошибка сохранения назначения роли
30123	Ошибка сохранения привилегий
30124	Ошибка сохранения ссылочной целостности
30125	Ошибка восстановления привилегий
30126	Ошибка восстановления пользователя
30127	Ошибка восстановления роли
30128	Ошибка восстановления назначения роли
30129	Ошибка восстановления синонима
30130	Ошибка восстановления ссылочной целостности
30131	Ошибка восстановления представления
30132	Ошибка восстановления таблицы
30133	Ошибка поиска LHB-раздела
30134	Такой файл на ленте уже есть
30135	Ошибка в данных на ленте
30136	Нет маркера 'конец данных' на ленте

Код	Причина
30137	Слишком старая версия Линтера
30138	Слишком новая версия Линтера
30139	Ошибка переименования файла
30140	Ошибка создания каталога
30141	Ошибка запуска программы
30142	Неизвестный метод сжатия блока
30143	Версия Линтера не совпадает с версией архива
30144	Неверно закодированный блок
30145	Файл не является LHB архивом
30146	Нет памяти
30147	Нет данных на ленте
30148	Ошибка блокирования ленты
30149	Ошибка разблокирования ленты
30150	Невозможно продолжить инкрементный архив
30151	Не найден конец ленты
30152	Инкрементный файл не закончен
30153	Ошибка поиска на ленте
30154	Ошибка установки размера блока на ленте
30155	Ошибка записи маркера файла
30156	Ошибка поиска маркера файла
30157	Функция не поддерживается
30158	Не могу восстановить БД - не совпадает порядок байт
30159	Ошибка сохранения таблицы
30160	Ошибка чтения из устройства ввода
30161	Файл не закончен - корректное восстановление невозможно
30162	Ошибка Линтера
30163	Нет свободного места на диске
30164	Не могу сохранить триггер
30165	Не могу сохранить процедуру
30166	Не могу восстановить триггер
30167	Не могу восстановить процедуру
30168	Программа остановлена пользователем
30169	Ошибка сохранения информации о сервере
30170	Ошибка сохранения правила репликации
30171	Ошибка восстановления информации о сервере
30172	Ошибка восстановления правила репликации
30173	Не могу сохранить событие
30174	Не могу восстановить событие
30175	Ошибка сохранения последовательности
30176	Ошибка восстановления последовательности

Код	Причина
30177	Неверные параметры командной строки
30178	Ошибка синтаксиса в скрипте
30179	Ошибка при создании контрольной точки
30180	Ошибка сохранения фразовых индексов
30181	Ошибка сохранения устройств
30182	Контрольная точка не существует в БД
30183	Неверный пароль архива
30184	Ошибка при загрузке сообщений
30185	Ошибка чтения директории

© Архивирование и восстановление базы данных. АО НПП «РЕЛЭКС», 1990-2026

Приложение 2
Примеры применения утилиты lhb

Создание инкрементного архива и его поддержка

Необходимо нам это для того, чтобы в течение определенного времени иметь наиболее полный резервный архив БД, но место на жестком диске ограничено, то есть мы не можем держать сразу несколько копий полностью сохраненной БД.

Пример

Производим запуск инкрементного сохранения:

```
lhb s -u SYSTEM/MANAGER8 -f db.lhb -startinc
```

В файле `db.lhb` будет находиться полностью сохраненная БД, и в самой БД будет установлена контрольная точка, указывающая, с какого места необходимо продолжать инкрементное наращивание. С этого момента все вновь создаваемые файлы журнала также будут оставаться в БД (поэтому, если БД очень быстро разрастается в объеме, то лучше воспользоваться периодическим полным сохранением БД).

Далее, по прошествии определенного времени, нам необходимо дописать к файлу архива те изменения, которые произошли в БД со времени последнего инкрементного архивирования:

```
lhb s -u SYSTEM/MANAGER8 -f db.lhb -inc
```

Файл следует указывать именно тот, для которого мы начинали инкрементный архив. Утилита проверит соответствие контрольной точки, записанной в файл, той, которая находится в БД, и в случае соответствия допишет в конец файла архива те файлы журнала, которые были добавлены за время от последнего сохранения. Контрольная точка в БД будет обновлена для того, чтобы освободить уже ненужные файлы журнала (они впоследствии в ходе работы ядра будут удалены), и следующее инкрементное архивирование будет уже производиться от момента нашего последнего сохранения.

Операцию можно повторять необходимое число раз.

Впоследствии при возникновении каких-либо неприятностей можно будет восстановить данные из этого архива. При старте на восстановленной БД ядро произведет запись всех изменений из журнала в БД, и она будет готова к работе. Если резервное сохранение производилось достаточно регулярно, она будет наиболее близка к изначальной.

По прошествии определенного (возможно длительного) времени необходимо будет начать новый инкрементный архив БД. Ведь если БД довольно сильно разрослась в объеме за это время, то лучше будет произвести полное или начать новое инкрементное резервное сохранение, так как последующее восстановление можно осуществить более быстро, нежели произвести восстановление первоначальной БД и всех накопленных файлов журнала, и предоставить ядру СУБД ЛИНТЕР переносить изменения из файлов журнала в файлы самой БД. При определенных условиях это может потребовать значительного времени.

Исходя из всего вышесказанного, прежде чем начать новое инкрементное сохранение, лучше закрыть старое.

Пример

Если все еще существует файл `db.lhb` (в который мы накапливали инкрементные изменения), то подаем команду:

```
lhb s -u SYSTEM/MANAGER8 -f db.lhb -stopinc
```

После чего контрольная точка, соответствовавшая файлу архива, удалится из БД. Никакие новые данные при этом в файл архива сохранены не будут. Дальнейшее инкрементное наращивание этого файла станет невозможным.

Другой вариант, как удалить более не используемую контрольную точку (необходимость может возникнуть в случае, если, например, файл архива был поврежден):

```
lhb cp -u SYSTEM/MANAGER8 -list
```

Результатом будет примерно следующее:

```
Linter On-line BACKUP utility v.4.1.0.893
for RDBMS Linter SQL v. 6.0.20 for Linux
Copyright (C) 1990-2023 Relex, Inc. All rights reserved.
CP(00): Fri Jul 19 10:18:41.04 2023 BACKUP (LongLived) File: 0
CP(01): Fri Jul 19 10:20:05.19 2023 BACKUP (LongLived) File: 0
```

В БД установлены две контрольные точки, каждая из которых была создана в результате старта инкрементного архивирования. Допустим, что первая из них нами уже не будет использоваться для инкрементного архивирования, и поэтому вполне логично её удалить:

```
lhb cp -u SYSTEM/MANAGER8 -clear 0
```

В результате первая контрольная точка удалится, а файлы журнала освободятся, и если они нигде больше не нужны, будут удалены.

Создание полного архива БД

Преимущества использования метода полного архивирования данных:

- наиболее быстрое последующее восстановление БД из файла архива;
- сохранение информации о фразовых индексах поддерживается только для полного сохранения;
- не используются контрольные точки, и, как следствие, не остаются более не используемые файлы журнала.

Недостатки:

- относительно длительный процесс архивации;
- т.к. БД может достигать значительного размера, то хранение нескольких файлов архива может потребовать немало места;
- по вышеуказанным причинам относительно затруднительно вести частое резервное сохранение БД.

Поэтому метод, применяемый при резервном архивировании, должен выбираться системным администратором в каждом случае индивидуально. Наиболее оптимальный – комбинирующий методы полного резервного сохранения и инкрементного.

Пример

Необходимо произвести полное сохранение БД в файл `full.lhb`, который в свою очередь следует разбить на тома по 650Мб (например, для хранения на CD). Хотелось бы, чтобы утилита не выдавала запрос на создание нового тома и чтобы сохранилась информация об имеющихся фразовых индексах. Подаем следующую команду:

```
lhb s -u SYSTEM/MANAGER8 -f full.lhb -v 650M -qc NV -pi
```

Другой вариант запуска: имеется текстовый файл `run.txt`, в котором написана эта строка:

```
s -u SYSTEM/MANAGER8 -f full.lhb -v 650M -qc NV -pi
```

Тогда подаем команду читать параметры из внешнего файла:

```
lhb ef run.txt
```

Результат будет аналогичен предыдущему.

Восстановление из полного архива БД

При восстановлении не имеет значения, какое было сохранение БД – полное или инкрементное.

Ядро СУБД ЛИНТЕР может быть не запущено.

Перед восстановлением желательно создать/очистить каталог, в который будет производиться восстановление БД (настоятельно рекомендуется, чтобы восстанавливаемый каталог не был каталогом существующей БД). Также желательно убедиться в наличии свободного места на диске, исходя из того, что его может потребоваться в два-три раза больше, чем занимает сам файл архива, из-за используемой в нем компрессии данных.

Пример

Подаем команду на восстановление (предположим, что файл архива был разбит на тома, и мы не хотим подтверждать открытие каждого нового тома) в каталог `NEW_DB`:

```
lhb r -u SYSTEM/MANAGER8 -f full.lhb -qc NV -p NEW_DB
```

В результате все файлы восстановленной БД будут помещены в каталог `NEW_DB`. Можно производить запуск ядра СУБД ЛИНТЕР на восстановленной БД.

Если восстановленная БД содержала информацию о фразовых индексах, то при старте ядро проанализирует эту информацию и создаст заново необходимые фразовые индексы (что может потребовать определенного времени). Затем БД будет готова к работе.

Сохранение и восстановление отдельных объектов БД из архива БД

Сохранение отдельных объектов отличается от полного сохранения структурой создаваемого файла архива. При сохранении отдельных объектов файлы журнала не сохраняются.

Для восстановления объектов необходимо работающее ядро СУБД ЛИНТЕР на уже существующей БД. Также желательно, чтобы восстанавливаемые объекты отсутствовали в БД.

При сохранении объектов сохраняется их структура, а для таблиц также сохраняются данные (если не указан ключ `-otwd` при сохранении таблиц).

Случаи необходимости сохранения и восстановления отдельных объектов БД:

- какой-либо из объектов БД очень часто меняется по сравнению с другими (например, таблица), и поэтому её периодическое сохранение более выгодно, чем создание полного архива БД или инкрементного;
- для переноса в другую БД некоторых объектов (например, таблиц и их триггеров, пользователей и ролей);
- при сильной нехватке свободного места на диске.

Пример

Допустим, необходимо сохранить таблицу CUSTOMER со всеми её зависимыми объектами и внешними ссылками. Файл архива `customer.lhb` зашифруем по паролю <password> и снабдим комментарием, что это таблица CUSTOMER.

```
lhb s -u SYSTEM/MANAGER8 -f customer.lhb -g password -c "CUSTOMER
 table" -ot CUSTOMER -oref -d
```

Для последующего восстановления данной таблицы в БД вводим команду:

```
lhb r -u SYSTEM/MANAGER8 -f customer.lhb
```

Утилита предложит ввести пароль и попытается восстановить таблицу в БД. Если таблица с таким именем уже существует у данного пользователя, то будет выдано соответствующее предупреждение.

Другой возможный вариант: у пользователя <BOLT> есть таблица CLIENTS, и он желает передать её со всем содержимым пользователю <SYSTEM>. Для этого пользователь <BOLT> сохраняет данную таблицу в файл:

```
lhb s -u BOLT/ITALIC -f clients.lhb -ot CLIENTS
```

и передает созданный файл архива пользователю <SYSTEM>, а тот уже восстанавливает эту таблицу как свою по команде:

```
lhb s -u SYSTEM/MANAGER8 -f clients.lhb -ot CLIENTS -own
```

Следует всегда в подобных случаях стараться переносить объекты, от которых эта таблица зависит.

Пример использования языка сценариев для проведения регулярного инкрементного архивирования данных

Преимущества от использования языка сценариев для архивирования данных: можно гибко настроить график сохранения по конкретным временным условиям в определенные файлы архива.

Пример листинга файла сценариев, который позволяет сохранять БД при запуске и затем каждый день в 02:00; при этом предыдущие файлы переименовываются соответственно в `arc1.lhb...arc4.lhb`. Свежий файл имеет имя `db.lhb`. Если при запуске `lhb` задать ключ `-fl FILE.LOG`, то история сохранения будет накапливаться в файле `FILE.LOG`. Содержимое файла `script.bsl`:

```
/* ----------------------------------------------------------------
 */
variables:
   USERNAME     ="SYSTEM";  /* user name */
   USERPASSWORD ="MANAGER8"; /* user password */
   ARCDEVICE    ="./";      /* for new files */
   ARCFNAME     ="";        /* new name for old file */
   CHKSUF       = ".lhb";   /* suffix for checkpoint file */
   NUMFILE      = 1;
/* ----------------------------------------------------------------
 */
rights:
```

```
everyday (time = '02:00')
   {
   NUMFILE = 1;
   while (NUMFILE < 5)
      {
      if (exist (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
                 ".lhb"))
         {
         if (NUMFILE == 1)
            delete (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
                  ".lhb");
         Else
            rename (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
                  ".lhb",
                  ARCDEVICE+"arc" + TOSTR(NUMFILE-1) + ".lhb");
         } /* if */
      NUMFILE = NUMFILE + 1;
      } /* while */
   rename (ARCDEVICE+"db.lhb", ARCDEVICE + "arc" +
         TOSTR(NUMFILE-1) + ".lhb");
   backup ("s -u " + USERNAME + "/" + USERPASSWORD + " -f " +
     ARCDEVICE + "db.lhb" + " -qc DF");
   logprint (CTIMESTAMP() + "---File" + "db" + CHKSUF +
         "created.\n");
exception: /* for everyday */
   print ("Error=" + TOSTR(CERROR) +
      ",LinError=" + TOSTR(LINERROR) +
      ",SysError=" + TOSTR(SYSERROR));
   logprint (CTIMESTAMP() + "---Error=" +
      TOSTR(CERROR) +
      ",LinError=" + TOSTR(LINERROR) +
      ",SysError=" + TOSTR(SYSERROR));
   stop;
   } /* everyday */
/*
 ----------------------------------------------------------------
 */
special:
before /* just after the start */
   {
   NUMFILE = 1;
   while (NUMFILE < 5)
      {
      if (exist (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
               ".lhb" ))
         {
```

```
         if (NUMFILE == 1)
            delete (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
                  ".lhb" );
         Else
            rename (ARCDEVICE+"arc" + TOSTR(NUMFILE) +
                  ".lhb",
               ARCDEVICE+"arc" + TOSTR(NUMFILE-1) + ".lhb" );
      } /* if */
   NUMFILE = NUMFILE + 1;
   } /* while */
   rename (ARCDEVICE + "db.lhb", ARCDEVICE + "arc" +
      TOSTR(NUMFILE-1) + ".lhb");
   backup ("s -u " + USERNAME + "/" + USERPASSWORD + " -f " +
     ARCDEVICE + "db.lhb" + " -qc DF");
   logprint (CTIMESTAMP() + "---File" + "db"+ CHKSUF +
      "created.\n");
   }
after /* after stop or Ctrl-C */
   {
   print ("---Stop backup system");
   if (CERROR != 0)
      logprint (CTIMESTAMP() + "---Error present:" +
               TOSTR(CERROR));
   logprint (CTIMESTAMP() + "---Stop backup system\n");
   }
iferr /* global */
   {
   print ("Error=" + TOSTR(CERROR) +
         ",LinError=" + TOSTR(LINERROR) +
         ",SysError=" + TOSTR(SYSERROR));
   logprint (CTIMESTAMP() + "---Error=" +
         TOSTR(CERROR) +
         ",LinError=" + TOSTR(LINERROR) +
         ",LinError=" + TOSTR(LINERROR));
   stop;
   }
/*
 ---------------------------------------------------------------
 */
```

Для запуска данного файла сценария вводим команду:

```
lhb script -ft script.bsl -fl FILE.LOG
```

Утилита выполнит полное сохранение БД и затем будет ожидать 2:00, после чего повторит очередное полное сохранение в новый файл архива.

Указатель ключей

-?, 27
-bg, 26
-briefversion, 27
-c, 22
-clear, 36
-crpart, 36
-d, 31
-db, 24
-dbonly, 25
-dev, 36
-diff, 25
-dp, 35
-f, 21
-fl, 22
-fo, 34
-ft, 21
-g, 22
-h, 27
-inc, 34
-ldev, 36
-le, 26
-list, 36, 36
-lr, 26
-movecp, 24
-mypid, 27
-n, 23
-nocompress, 25
-notestcrc, 27
-oa, 31
-oall, 28
-oap, 31
-ob, 33
-oe, 29
-on, 33
-only, 32
-op, 33
-oq, 33
-or, 31
-oref, 30
-os, 30
-osr, 31
-ot, 29
-otr, 33
-otwd, 30
-ou, 29
-out, 26
-out+, 26
-ov, 30
-own, 34
-p, 23
-pass, 22
-passfile, 23
-pf, 21
-pg, 22
-pi, 35
-pid, 27
-priority, 35
-qc, 37
-qq, 37
-qx, 37
-setpass, 23
-startinc, 34
-stopinc, 34
-takeforeign, 33
-u, 21
-v, 22
-version, 27
-vi, 34
-wait, 23
-wd, 25