

База знаний РЕД База Данных

- СУБД РЕД База Данных
 - [Описание продукта](#)
 - [Документация](#)
 - [Системные требования](#)
 - [Установка РЕД База Данных \(Открытая редакция\)](#)
 - [Обновление СУБД РЕД База Данных \(Открытая редакция\)](#)
 - [Диагностика проблем](#)
 - [Часто задаваемые вопросы по СУБД РЕД База Данных \(FAQ\)](#)

Описание продукта

СУБД Ред База Данных (далее СУБД или Ред База Данных) используется для упорядоченного хранения и обработки больших объемов информации. Ред База Данных представляет собой мощную современную СУБД с открытым кодом. Ядро Ред Базы Данных построено на основе одной из самых известных и распространенных в мире СУБД с открытым кодом – **Firebird**, которая используется в решениях различного масштаба: от встроенных аппаратных систем и решений для небольших компаний до IT-систем крупнейших транснациональных корпораций с размерами баз данных до десятков терабайт и десятками миллионов транзакций в день.

Назначение

СУБД Ред База Данных предоставляет пользователям следующие возможности:

- поддержка всех основных платформ и операционных систем (среди них: Windows, семейство Linux, BSD Unix, IBM AIX, HP-UX, Sun Solaris и др.);
- поддержка 64-битных систем;
- поддержка многопроцессорных и многоядерных аппаратных платформ;
- высокое быстродействие;
- возможность хранения базы данных в одном отдельном файле;
- возможность аутентификации и авторизации пользователей с использованием в качестве источников сведений об учетных записях пользователей защищенной БД пользователей или системного каталога;
- возможность «горячего» резервного копирования БД и инкрементного резервного копирования, в т.ч. с применением аппаратных решений для резервного копирования;
- наличие модулей сопряжения практически для всех используемых сред разработки, результатов тестов этих модулей и гарантия стабильной работы;
- возможность работы во «встроенном» ПО (**embedded**) в локальном режиме в виде библиотеки **DLL** без отдельной установки и настройки СУБД Ред База Данных, в том числе поддержка встраивания в виртуальную машину Java;
- возможность одновременной модификации базы данных несколькими пользователями;
- обратная совместимость с базами данных от предыдущих версий **Firebird**;
- многоверсионная архитектура;
- модульная архитектура;

- модульная архитектура,
- соответствие большинству требований стандарта ISO/ANSI SQL;
- низкие требования к аппаратному обеспечению для небольших баз данных;
- большие возможности по расширению функционала самой СУБД посредством модулей;
- ядро, изначально основанное на многоверсионной архитектуре (**MGA**);
- полное соответствие принципам атомарности, непротиворечивости, изоляции, долговечности (**ACID**).

Вы можете подробнее ознакомиться с установкой **СУБД РЕД База Данных**, а также пройти обучающий курс по работе с **СУБД РЕД База Данных**, просмотрев наши обучающие видео:

- на **RuTube**:
 - [Установка СУБД Ред База Данных на РЕД ОС](#);
 - [Курс «Администрирование СУБД Ред База Данных 3.0»](#).
- в **Яндекс.Дзен**:
 - [Установка СУБД Ред База Данных на РЕД ОС](#);
 - [Курс «Администрирование СУБД Ред База Данных 3.0»](#).

На наших каналах вы также сможете найти много другой полезной информации.

Документация

Наиболее полную, актуальную и регулярно дорабатываемую документацию можно найти [на сайте СУБД РЕД База Данных](#) в разделе «[Документация](#)».

Системные требования

[Аппаратная часть](#)

[Программная часть](#)

Аппаратная часть

Для установки и нормальной работы СУБД Ред База Данных персональный компьютер должен быть оснащен комплектующими со следующими характеристиками:

- Процессор **Intel Pentium D** и выше, процессор **AMD** с поддержкой **x86_64** или процессор **Intel** с поддержкой набора инструкций **EM64T** для архитектуры **x86_64**.
- Оперативная память от **2048Mb**.
- Запоминающее устройство объемом **не менее 16 Гб**.
- Клавиатура 101/102-х клавишная русская/латинская.
- Сетевая карта с поддержкой **Ethernet**.

Рекомендованные системные требования подбираются индивидуально в зависимости от размера базы данных, количества пользователей и характера нагрузки.

Программная часть

Для установки и эксплуатации СУБД в ОС семейства *Windows* не требуется установка дополнительного программного обеспечения.

Для установки и эксплуатации СУБД в ОС семейства Linux необходимы библиотеки: **glibc 2.4 и выше, libstdc++ от gcc 3.4.3 и выше.**

Установка РЕД База Данных (Открытая редакция)

[Установка для Linux на примере РЕД ОС 7.3](#)

[Загрузка установочного файла на РЕД ОС 7.3](#)

[Последовательность установки СУБД на РЕД ОС 7.3](#)

[Установка для ОС Windows](#)

[Загрузка установочного файла на ОС Windows](#)

[Последовательность установки СУБД на ОС Windows](#)

Вы можете подробнее ознакомиться с установкой **СУБД РЕД База Данных**, а также пройти обучающий курс по работе с **СУБД РЕД База Данных**, просмотрев наши обучающие видео:

- на **RuTube**:
 - [Установка СУБД Ред База Данных на РЕД ОС](#);
 - [Курс «Администрирование СУБД Ред База Данных 3.0»](#).
- в **Яндекс.Дзен**:
 - [Установка СУБД Ред База Данных на РЕД ОС](#);
 - [Курс «Администрирование СУБД Ред База Данных 3.0»](#).

На наших каналах вы также сможете найти много другой полезной информации.

Установка для Linux на примере РЕД ОС 7.3

Загрузка установочного файла на РЕД ОС 7.3

Скачать РЕД Базу Данных можно [на официальном сайте](#) компании в разделе «**Загрузки**».

РЕД БАЗА ДАННЫХ
Главная Новости Продукты Услуги Уязвимости Документация **Загрузки** О нас

Ред База Данных 3.0

СУБД Ред База Данных (Red Database) – промышленная российская система управления базами данных с открытым кодом. Ядро СУБД Ред База Данных построено на основе одной из самых известных и распространенных в мире СУБД с открытым кодом – Firebird, которая используется в решениях различного масштаба: от встроенных аппаратных систем и решений для небольших компаний до ИТ-систем крупнейших корпораций с размерами баз данных до десятков терабайт и десятками миллионов транзакций в день. СУБД Ред База Данных соответствует принятым в мировой практике промышленным стандартам качества, надежности и безопасности. В состав дистрибутива СУБД Ред База Данных входят: сервер СУБД (исполняемые и исходные коды), дополнительные модули расширения СУБД, средства инсталляции, настройки и администрирования СУБД.

v3.0.12	25 октября 2023 г. 10:13
v3.0.11	14 июня 2023 г. 16:33
v3.0.10	10 февраля 2023 г. 10:31
v3.0.9	1 июля 2022 г. 17:25
v3.0.8.0	12 августа 2021 г. 10:17

[Показать все версии](#)

Для скачивания необходимо авторизоваться или зарегистрироваться.

Для установки в РЕД ОС необходимо скачать бинарный файл нужной разрядности и назначить на него права:

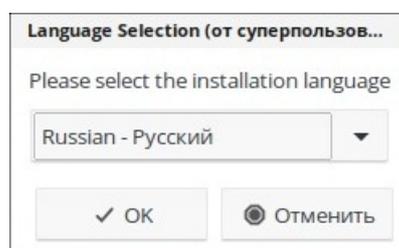
```
sudo chmod +x <путь_до_файла>
```

Для старта установки запустите установочный бинарный файл:

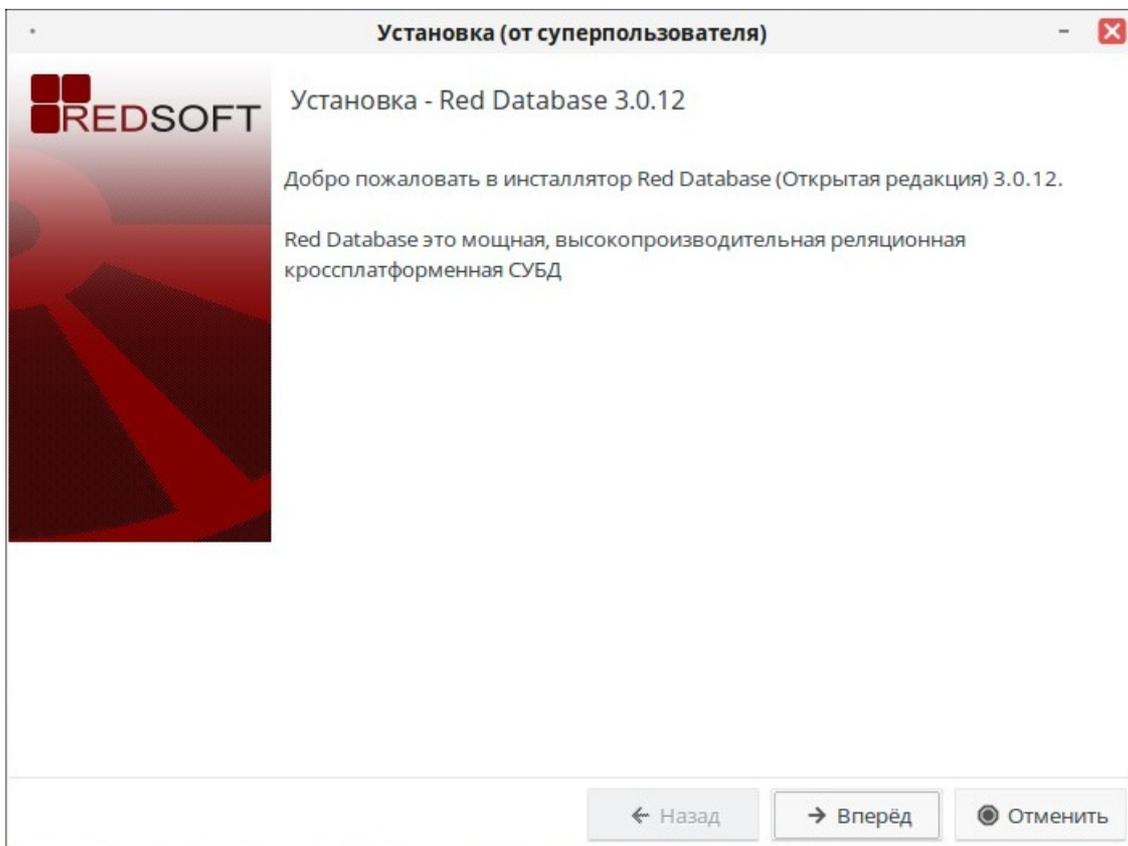
```
sudo ./rdb30-linux-x86_64-open-3.0.12.bin
```

Последовательность установки СУБД на РЕД ОС 7.3

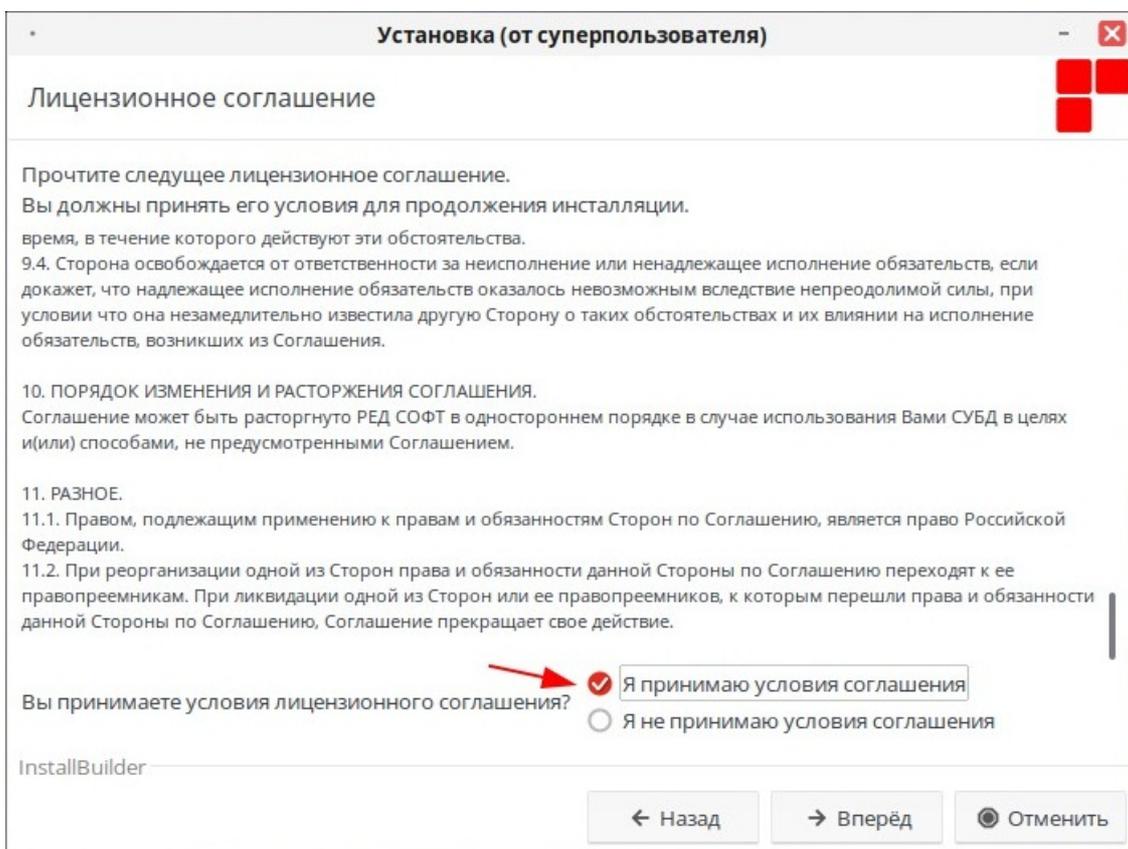
После запуска бинарного файла будет открыто окно настройки языка установщика. Выберите из списка необходимый язык и нажмите «**ОК**».



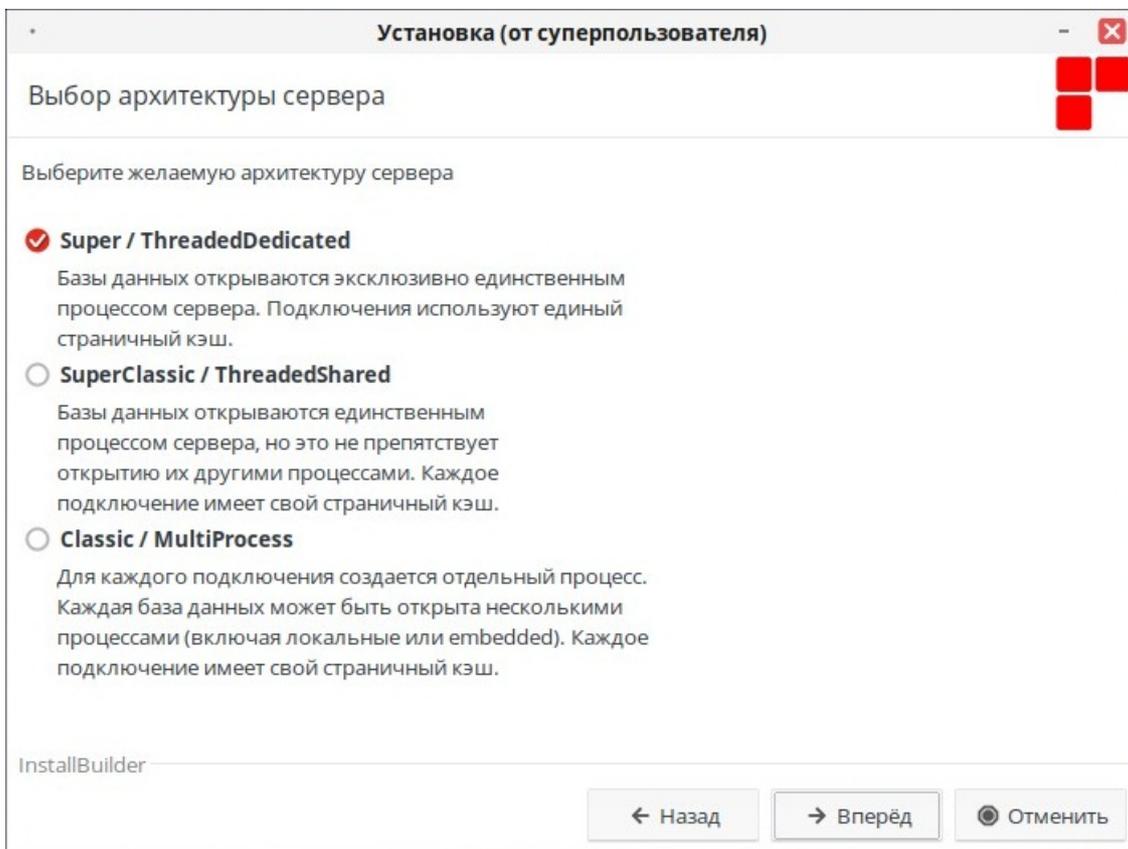
Будет открыто приветственное окно установщика. Нажмите «**Вперед**».



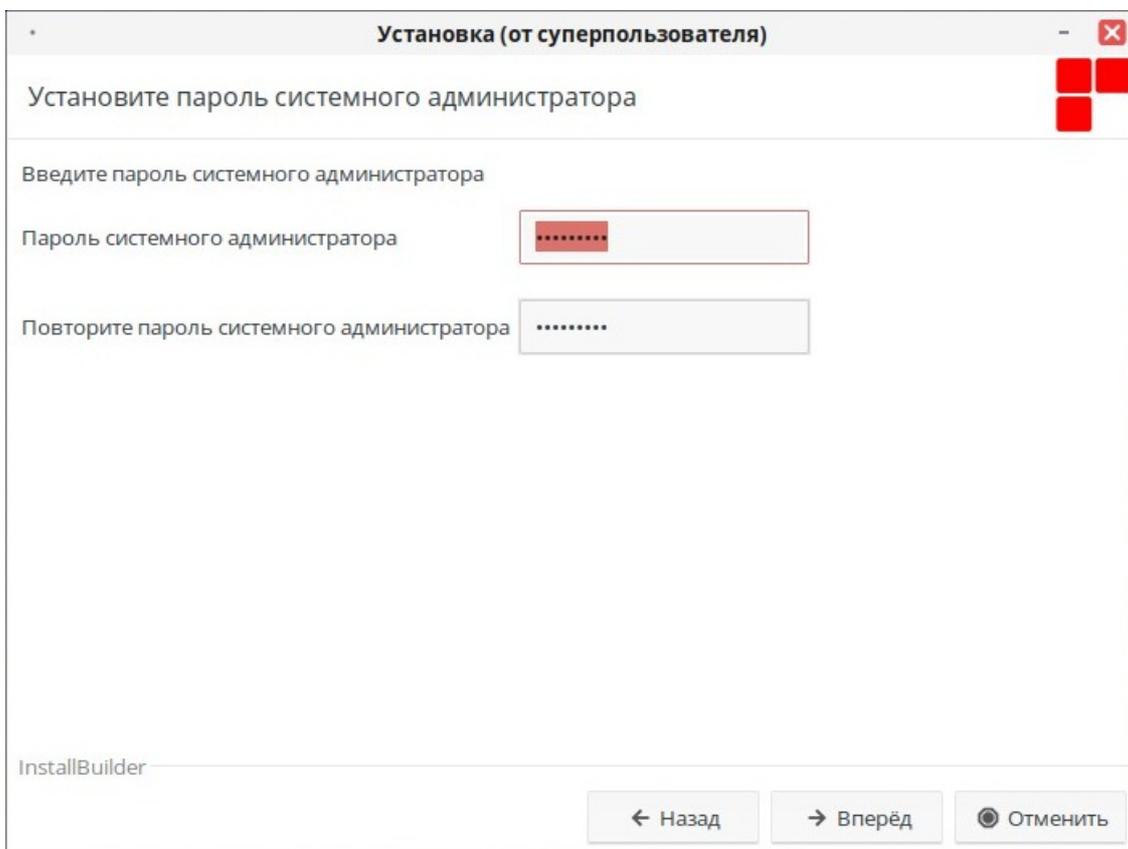
Далее прочтите и примите условия лицензионного соглашения, установив флаг для параметра «**Я принимаю условия соглашения**». Нажмите «**Вперед**».



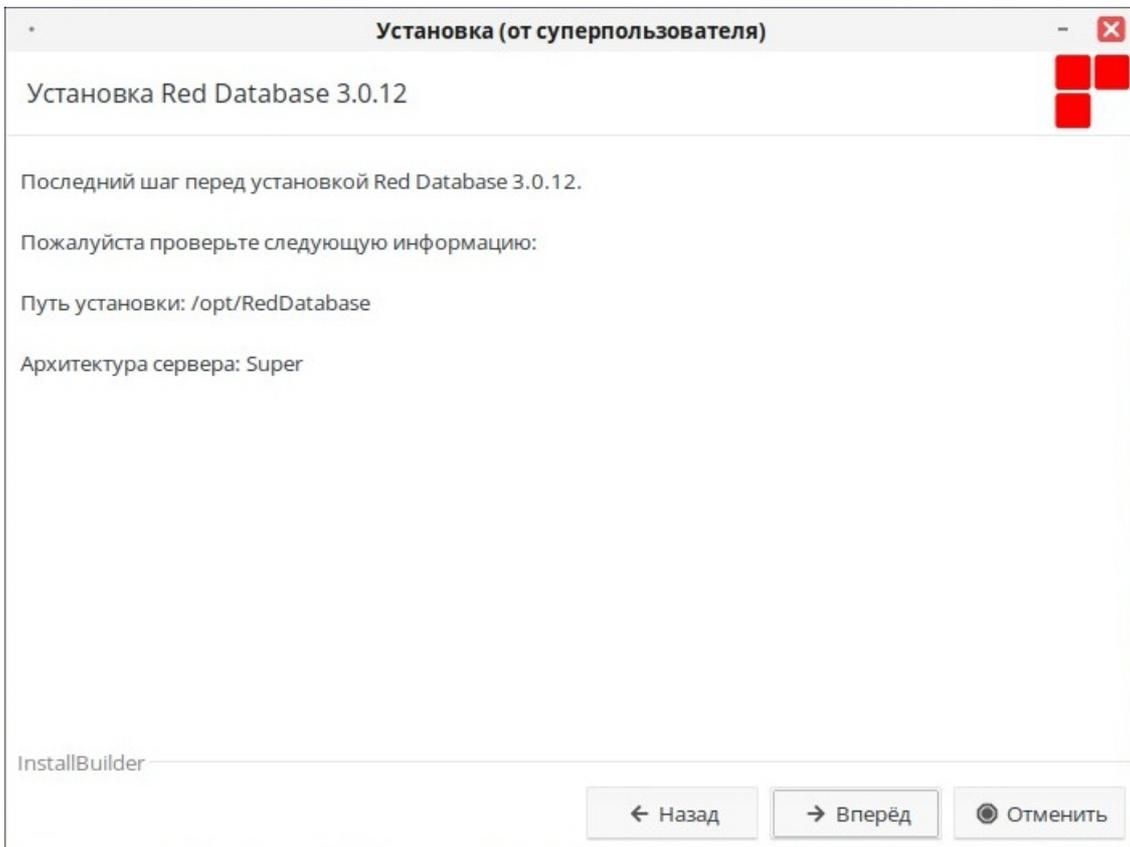
Выберите требуемую архитектуру сервера, установив флаг для соответствующего параметра. Нажмите «**Вперед**».



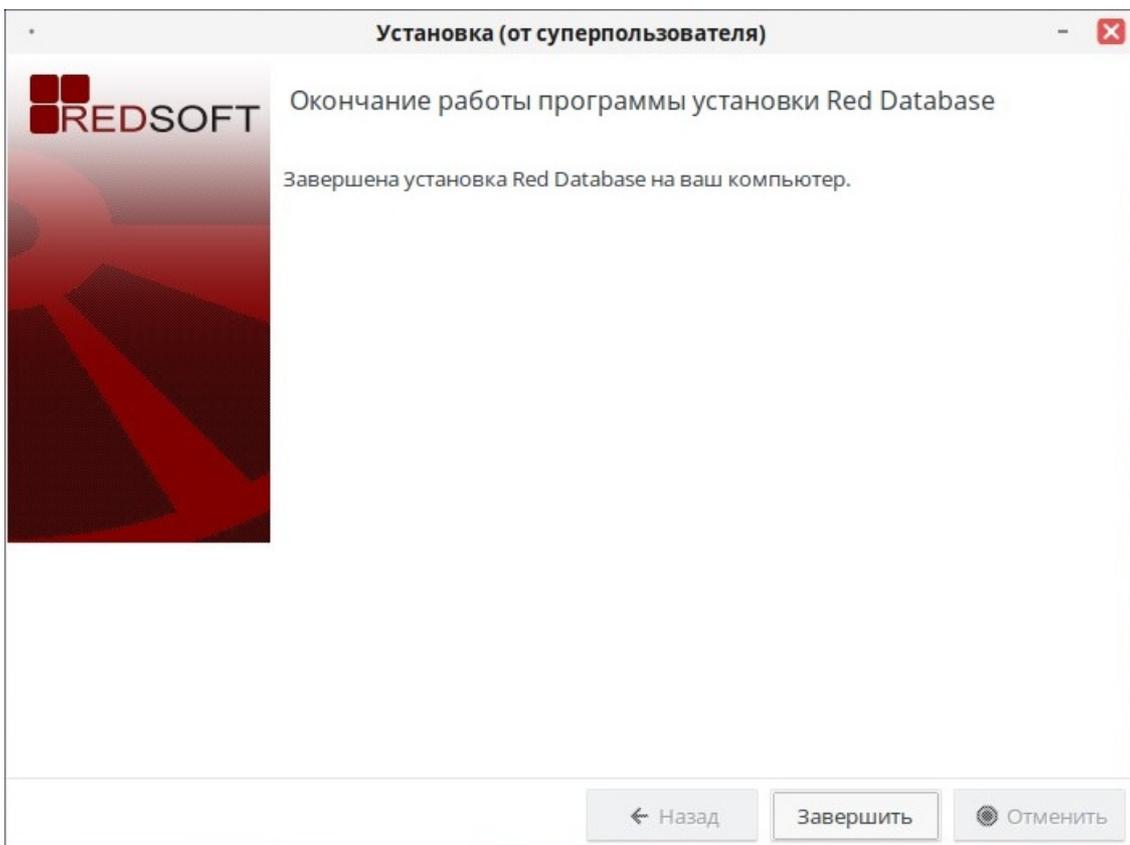
Установите пароль системного администратора. Пароль необходимо повторить для избежания опечаток. Нажмите «**Вперед**».



Проверьте указанные ранее параметры. Если все настроено верно, нажмите «**Вперед**» для запуска процесса установки. В противном случае нажмите «**Назад**» и внесите изменения в необходимые конфигурации.



После завершения процесса установки будет выведено соответствующее сообщение. Нажмите **«Завершить»**.



Установка для ОС Windows

Загрузка установочного файла на ОС Windows

Скачать РЕД Базу Данных можно [на официальном сайте](#) компании в разделе «**Загрузки**».

Версия	Дата загрузки
v3.0.12	25 октября 2023 г. 10:13
v3.0.11	14 июня 2023 г. 16:33
v3.0.10	10 февраля 2023 г. 10:31
v3.0.9	1 июля 2022 г. 17:25
v3.0.8.0	12 августа 2021 г. 10:17

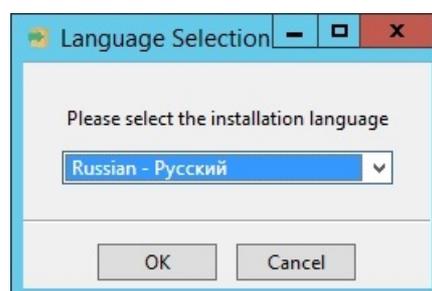
[Показать все версии](#)

Для скачивания необходимо авторизоваться или зарегистрироваться.

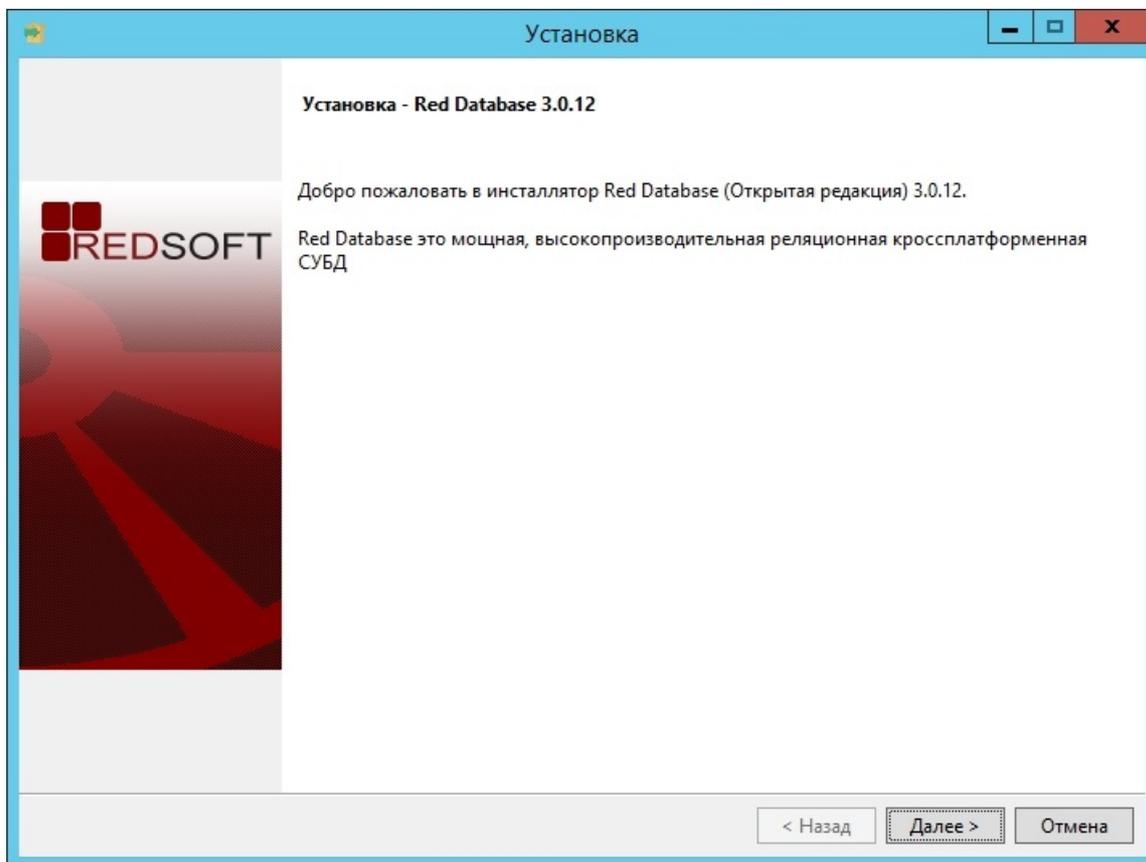
Для установки на Windows скачайте файл с расширением **.exe** нужной архитектуры. Далее от имени администратора запустите установщик.

Последовательность установки СУБД на ОС Windows

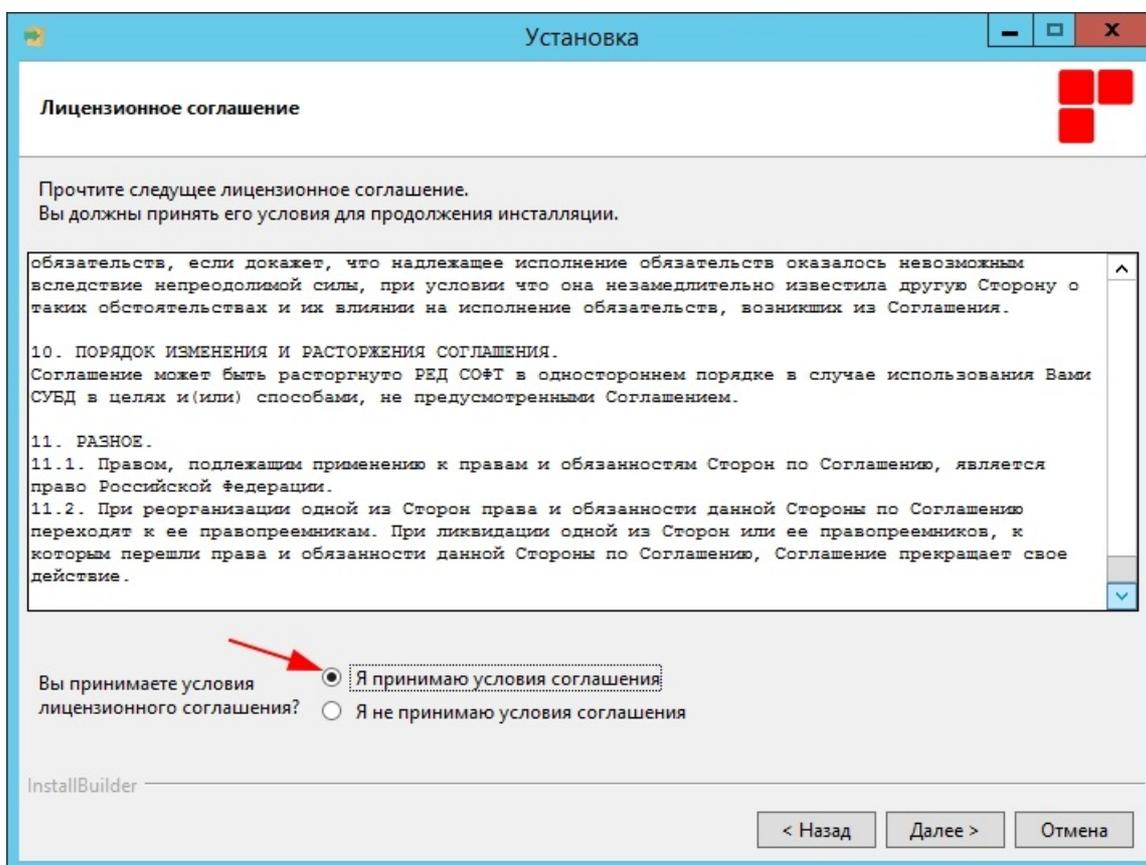
После запуска **exe**-файла будет открыто окно настройки языка установщика. Выберите из списка необходимый язык и нажмите «**ОК**».



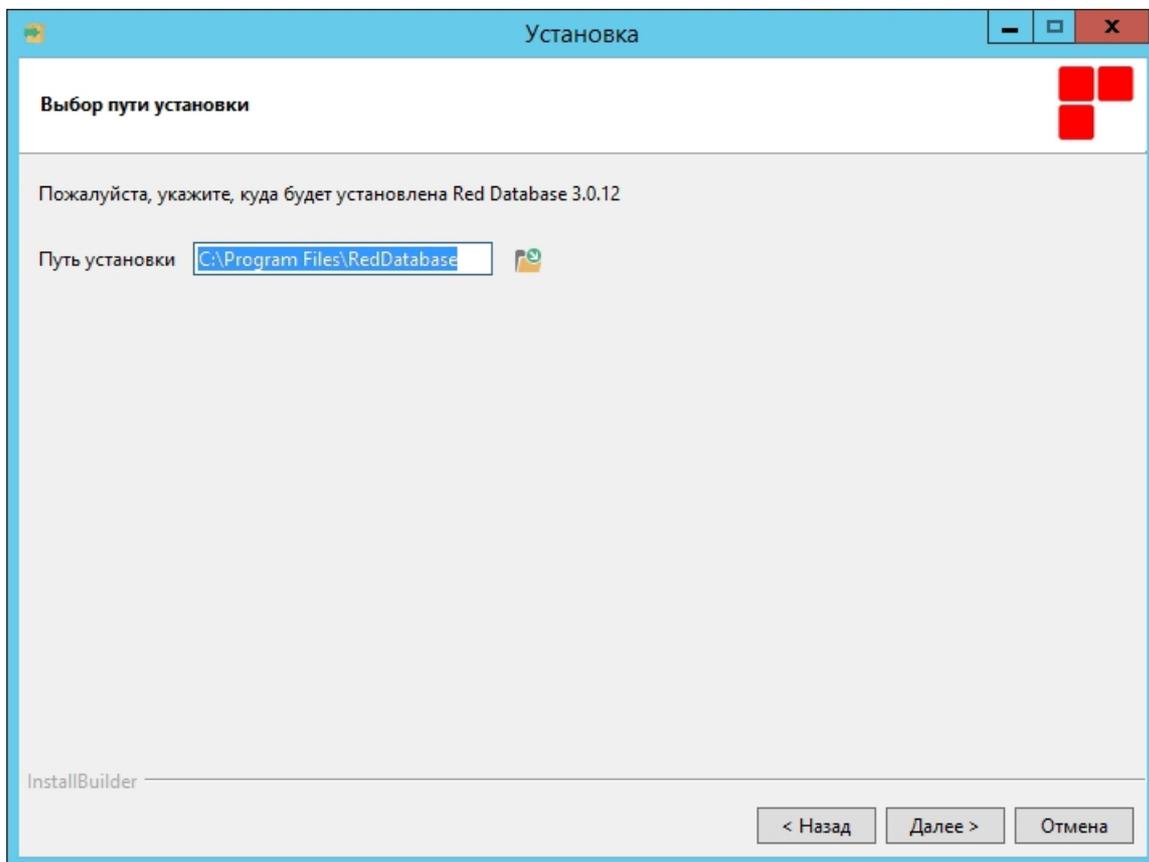
Будет открыто приветственное окно установщика. Нажмите «**Далее**».



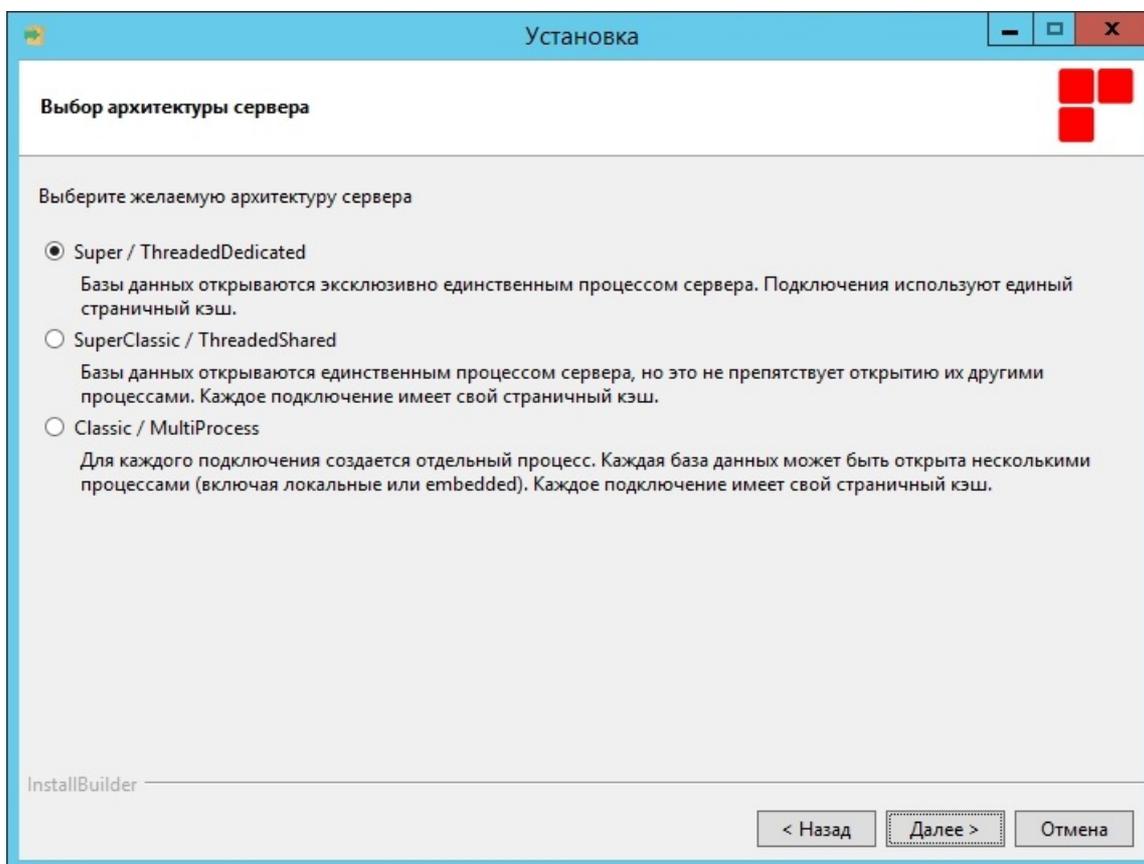
Далее прочтите и примите условия лицензионного соглашения, установив флаг для параметра «**Я принимаю условия соглашения**». Нажмите «**Далее**».



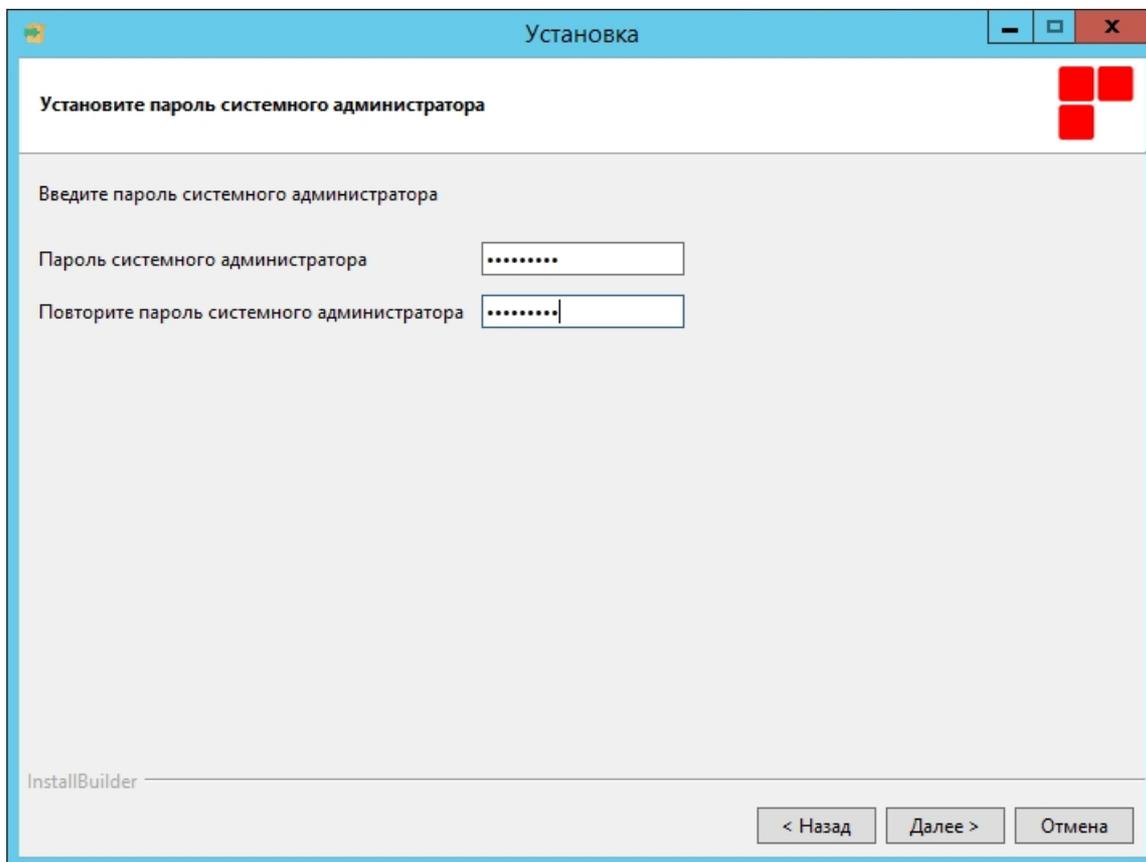
Выберите каталог для установки программы. По умолчанию используется **C:\Program Files\RedDatabase**.



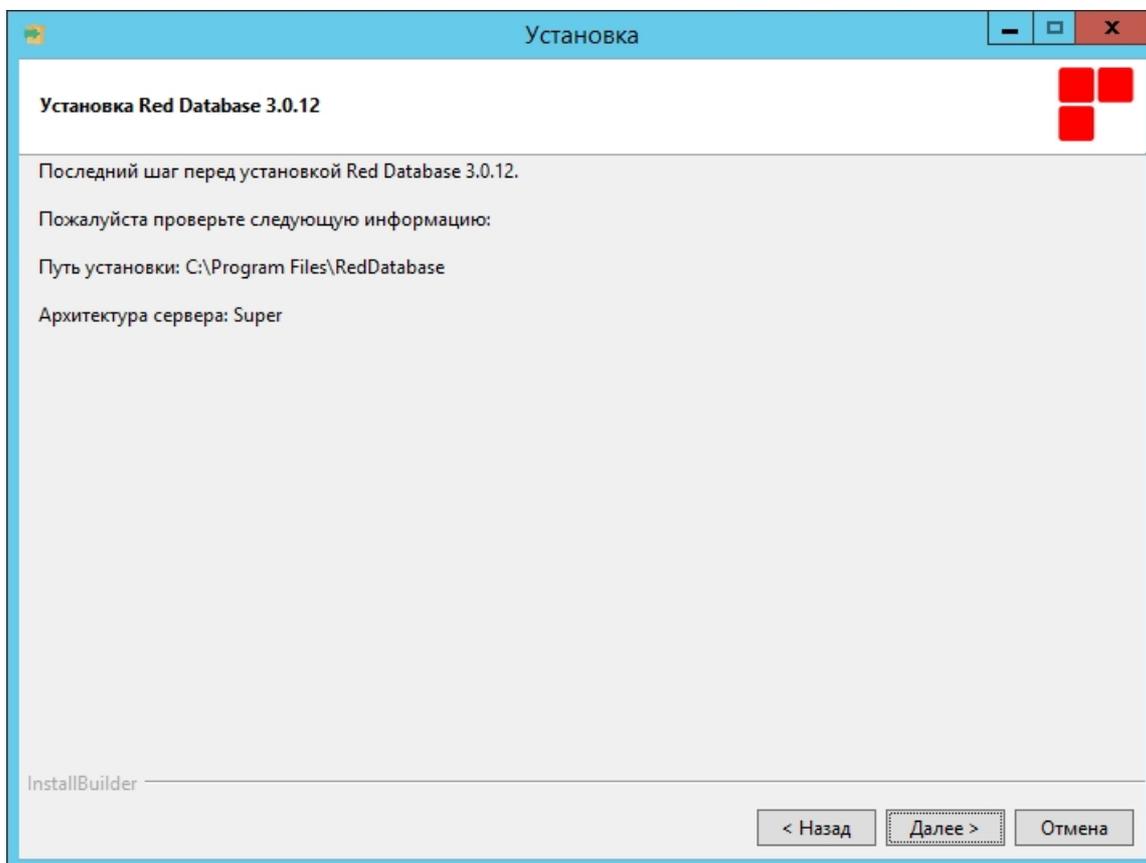
Выберите требуемую архитектуру сервера, установив флаг для соответствующего параметра. Нажмите «**Далее**».



Установите пароль системного администратора. Пароль необходимо повторить для избежания опечаток. Нажмите «**Далее**».



Проверьте указанные ранее параметры. Если все настроено верно, нажмите «**Далее**» для запуска процесса установки. В противном случае нажмите «**Назад**» и внесите изменения в необходимые конфигурации.

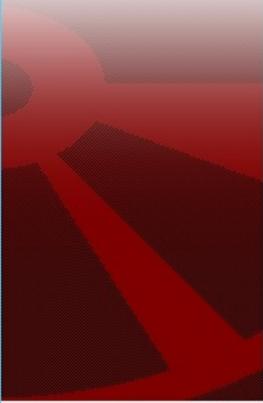


После завершения процесса установки будет выведено соответствующее сообщение. Нажмите «**Завершить**».



Окончание работы программы установки Red Database

Завершена установка Red Database на ваш компьютер.



< Назад

Завершить

Отмена

Обновление СУБД РЕД База Данных (Открытая редакция)

Скачать актуальную версию СУБД РЕД База Данных можно [на официальном сайте](#) компании.

Общий алгоритм обновления выглядит следующим образом:

1. Остановить СУБД Ред База Данных.
2. Проверить отсутствие подключений ко всем файлам баз данных на сервере и отсутствие процессов СУБД.
3. Сделать резервную копию каталога с СУБД (например, переименовав его).
4. Удалить текущую версию СУБД.
5. Выполнить установку новой версии СУБД.
6. Скопировать конфигурационные файлы со старой версии СУБД, дополнительно может потребоваться копирование плагинов (**plugins**) и *java (jar)* библиотек, которые не входят в типовой дистрибутив СУБД и устанавливаются и настраиваются отдельно.

Подробнее об обновлении СУБД можно ознакомиться в [Руководстве администратора](#) на сайте СУБД.

Диагностика проблем

Наличие мусора

Свободное место на диске

Оперативная память

Нагрузка на процессор

Нагрузка на диск

Таблица блокировок

Активность индексов

Поломка БД

Запрос дополнительной информации для первичной диагностики проблемы при обращении на ЛТП1

Анализ проблем производительности и поломки БД:

- Низкая производительность, симптомы:
 - долгое выполнение запросов;
 - долгое открытие окон в пользовательском интерфейсе;
 - жалобы на работу, вплоть до того, что вообще невозможно подключиться.
- Поломка БД, симптомы:
 - зависание процессов СУБД;
 - аварийное завершение процессов СУБД РБД при выполнении запросов;
 - ошибки в логе **firebird.log**.

Проблема производительности требует анализа на всех уровнях:

- прикладное ПО (толстый клиент, тонкий клиент, сервер приложений);
- сеть (проблемы и ошибки сети, загрузка сети, антивирусы и фаерволы);
- гипервизор на сервере (конкуренция с другими VM, память, процессор);
- ОС СУБД (постоянное потребление процессора на уровне 100%, очереди к диску, сеть, файловая система);
- проблемы внутри БД (блокировки, мусор, отсутствие индексов, неоптимальные планы).

Все показатели требуется проверять в момент пиковой нагрузки на сервер или зависания и сравнивать с показателями системы, работающей в штатном режиме.

Примеры проведения диагностики проблем

Наличие мусора

Проверку наличия мусора можно осуществить утилитой **gstat**:

```
/opt/RedDatabase/bin/gstat -h /path/to/db.fdb
```

Пример вывода:

```
Database "/opt/db/red.fdb"
Gstat execution time Wed Jun 30 15:53:27 2021

Database header page information:
Flags      0
Generation 236121
System Change Number 24
Page size  16384
Server     RedDatabase
ODS version 12.0
Oldest transaction 234835
Oldest active 234836
Oldest snapshot 234836
Next transaction 234839
Sequence number 0
Next attachment ID 1604
Implementation HW=AMD/Intel/x64 little-endian OS=Linux CC=gcc
Shadow count 0
Page buffers 2048
Next header page 0
Database dialect 3
Creation date Aug 18, 2020 15:08:32
Attributes force write

Variable header data:
Database backup GUID: {C45D8AFE-4293-4EEA-B798-F0360284C9D6}
Sweep interval: 0
Database GUID: {A5D296A8-833A-4377-F496-187F6ABB0D5E}
*END*

Gstat completion time Wed Jun 30 15:53:27 2021
```

Рекомендации:

- Большая разница между **OAT** (Oldest active) и **OIT** (Oldest transaction) — повод для принудительной сборки мусора с помощью команды `gfix -sweep`.
- Большая разница между **Oldest active** и **Next transaction** — долгие транзакции, рекомендуется обратиться к разработчикам прикладного ПО для исключения длинных транзакций. Если такое исключение невозможно, то рекомендуется принудительное завершение таких транзакций (например, каждый день завершать транзакции старше 12-24 часов) и сборка мусора командой `gfix -sweep`.

Свободное место на диске

Проверка наличия свободного места на диске, в том числе в каталоге `/tmp`, производится командой:

```
df -h
```

Рекомендации:

- при отсутствии или недостаточном объеме свободного места необходимо его физическое увеличение либо удаление ненужных файлов.

Оперативная память

Проверка наличия свободной оперативной памяти и занятость **swap** осуществляется командой

```
free -m
```

Использование файла подкачки — плохой признак, который также может быть причиной низкой производительности. Интенсивность использования **swap** можно оценить с помощью утилиты `atop`, см. параметры `swout` и `swin`, их активное изменение говорит о частом использовании **swap**.

Рекомендации:

- проверьте параметры конфигурационного файла **firebird.conf**:
 - `TempBlockSize` — минимальный размер блока сортировки и шаг его

расширения (при необходимости). Позволяет несколько ускорить работу алгоритма сортировки за счёт выделения памяти большими блоками. Определение оптимальных для конкретной задачи настроек может быть произведено экспериментальным путем, принимая во внимание размеры файлов сортировки. При наличии больших сортировок рекомендуется увеличить значение параметра (максимально 2147483647 байт).

- **TempCacheLimit** — для **SuperServer** и **SuperClassic** рекомендуется 64 Мб, для **Classic** = 8 Мб;
- **TempDirectories** — указываются через точку с запятой, следуют в порядке очереди; пока не заполнится первая директория, вторая будет ожидать. Рекомендуется упорядочить каталоги по скорости доступа.

- Настройте использование **swap**:

Создайте файл **/etc/sysctl.d/ram.conf**:

```
nano /etc/sysctl.d/ram.conf
```

Со следующим содержимым:

```
vm.swappiness = 10  
vm.dirty_ratio = 60  
vm.dirty_background_ratio = 2  
vm.min_free_kbytes = 1048576
```

После сохранения изменений выполните перезагрузку операционной системы.

Нагрузка на процессор

Проверка нагрузки на процессор:

```
top (atop, nmon)
```

Длительная 100% загрузка одного ядра — признак зависания.

Рекомендации:

- принудительно завершите этот процесс (в случае с архитектурой **Classic**), предварительно сняв **дамп** для передачи на исследование разработчикам, после чего обратитесь в Техническую поддержку;

- выполните команду `gfix -sweep` без активных подключений к БД;
- проверьте активность индексов и выполните пересчет селективности индексов (рекомендуется выполнять минимум 1 раз в неделю в период минимальной нагрузки);
- низкая загрузка процессоров — ожидание блокировки, смотрите таблицу блокировок (см. п. «[Таблица блокировок](#)»).
- проверьте трейс, возможно необходимо добавление индекса.

Нагрузка на диск

Проверка нагрузки на диск осуществляется командой:

```
iostat -xz (iostat -p sda -x -t 1)
```

где **sda** - имя исследуемого блочного устройства ввода/вывода.

Важное значение имеет длина очереди к диску `aqu-sz` или `avgqu-sz` (название параметра зависит от версии утилиты) — средний размер очереди запросов к диску.

Рекомендации:

- Проверить опции монтирования для `SSD = noatime,discard` (не рекомендуется использовать опцию монтирования **discard** на программном рейде **mdadm** из **NVME SSD**).
- Необходимо понять, какие запросы выполняют наибольшее количество input/output-операций. В этом помогут таблицы мониторинга (`MON$`) и анализ трейса. Возможно, потребуется создание дополнительного индекса и пересчет селективности индексов (`SET STATISTICS INDEX INDEX_NAME`).

Если на ЭВМ, на котором установлена СУБД Ред База Данных, используется схема реализации компьютерной памяти **NUMA** (Non-Uniform Memory Access), то работу архитектуры можно проверить с помощью утилиты **numastat**.

В выводе утилиты промахов по памяти (`numa_miss` — сколько раз промахнулись и вынуждены обращаться к памяти другого процессора) должно быть на порядок меньше попаданий (`numa_hit`), в идеале параметр `numa_miss` должен быть равен 0.

Рекомендации:

Если значение `numa_miss` высокое, следует привязать процессы Ред Базы к одному

из процессоров (`numactl -C 0 /opt/RedDatabase/bin/rdbserver`), а к другому процессору можно привязать сервер приложений или веб-сервер, если он расположен на том же сервере.

Дополнительно необходимо протестировать производительность дисковой подсистемы с помощью утилиты **fio** :

```
fio fio.conf
```

Пример конфигурационного файла:

```
[ro]
blocksize=16k
directory=/mnt/ssd
rw=randread
direct=1
size=15g
ioengine=psync
iodepth=5
runtime=600
iodepth=5
ioengine=psync

[wo]
blocksize=16k
directory=/mnt/ssd
rw=randwrite
direct=1
size=15g
ioengine=psync
iodepth=5
runtime=600
iodepth=5
ioengine=psync
```

Основные параметры конфигурационного файла:

- блоки `ro` и `wo` (на чтение и на запись) — размер блока 16Кб, можно

установить 8Кб, если размер страницы в БД равен 8;

- `directory` — каталог, в котором будет проводиться тестирование;
- `rw-randread` — рандомное чтение/запись;
- `direct=1` — флаг `o_direct`;
- `size=100` — размер файла 100Мб (из этого файла производится рандомное чтение/запись);
- `runtime=60` — время тестирования;
- `iodepth=5` — глубина очереди к диску, которую **fiio** выдает в файл;
- `ioengine=psync` — используемые функции чтения/записи, которыми пишет СУБД.

Параметры, на которые необходимо обратить внимание в выводе теста **fiio**:

- `IOPS` — количество операций чтения и записи в секунду;
- `lat` — задержка диска, необходимо ориентироваться на среднее (**avg**) значение.

Таблица блокировок

Примеры использования:

- печать всей таблицы:

```
/opt/RedDatabase/bin/rdb_lock_print -a -d /path/to/db > /path/to/lock_table.txt
```

- печать блокировок с очередями

```
/opt/RedDatabase/bin/rdb_lock_print -n -l -o -d /path/to/db > /path/to/lock_table.txt
```

- поиск зависшего процесса

```
/opt/RedDatabase/bin/rdb_lock_print -q 30 -d /path/to/db
```

В выводе `rdb_lock_print -a -d` можно увидеть следующие параметры:

- `Length` — максимальный размер таблицы блокировок на момент снятия снапшота;
- `Used` — текущий размер таблицы блокировок;

- **Mutex wait** — время ожидания мьютексов в процентах;
- **Hash slots** — ширина хэш-таблицы, которая используется для поиска блокировок. Число должно быть простым, чтобы хэш-алгоритм производил хорошее распределение;
- **Hash length** (*min/avg/max*) — длина хэш-таблицы;
- **Owners** — число соединений к серверу.

Рекомендации:

- Если в пиковые нагрузки в параметре **Hash length** значения **avg** > 3 или **max** > 10, то необходимо увеличивать параметр **LockHashSlots** (в **firebird.conf**), значение должно быть простым числом (максимум 65521).
- Высокий **Mutex wait** считается проблемой, нужно проверить селективность индексов и их активность.
- Встречается, что все процессы висят на ожидании одной блокировки. Это можно увидеть по выводу команды:

```
/opt/RedDatabase/bin/rdb_lock_print -c -n -l -o -d | grep "Pending"
```

Максимальное число и покажет проблемную блокировку. По типу и ключу можно попытаться понять, какая это блокировка, страница или другой ресурс. Это позволит предоставить рекомендации прикладному разработчику.

- Повисший процесс, который выдаст **rdb_lock_print -q 30**, рекомендуется завершить, предварительно сняв **дамп**.

Активность индексов

Определить неактивные индексы можно запросом:

```
select rdb$index_name from rdb$indices
where (rdb$system_flag is null or rdb$system_flag = 0)
and RDB$INDEX_INACTIVE > 0
order by rdb$foreign_key nulls first
```

Рекомендации:

- активируйте все индексы:

```
EXECUTE BLOCK AS
  DECLARE VARIABLE stmt VARCHAR(1000);
BEGIN
  for select 'ALTER INDEX '||rdb$index_name ||' ACTIVE;'
  from rdb$indices
  where (rdb$system_flag is null or rdb$system_flag = 0)
  and RDB$INDEX_INACTIVE > 0
  order by rdb$foreign_key nulls first
  into :stmt do EXECUTE STATEMENT :stmt;
END
```

- активируйте нужные индексы вручную.

Поломка БД

Поломка БД — это изменения в файле, нарушающие целостность и приводящие к ошибкам.

Основные причины:

- аппаратные (память, диск);
- комплексные (батарейка рейд-контроллера);
- программные (ошибки в коде СУБД).

При поломке БД ошибки могут встречаться как непосредственно при работе пользователей, так и в файле **firebird.log**.

Критические ошибки в файле *firebird.log*:

- internal Firebird consistency check (cannot find tip page);
- Error while trying to read from file;
- Error while trying to write to file;
- internal Firebird consistency check (can't continue after bugcheck);
- internal Firebird consistency check (decompression overran buffer);
- Pointer page (sequence NUM) lost;
- Pointer page (sequence NUM) inconsistent;
- Missing index root page;
- Transaction inventory pages lost;

- Transaction inventory pages confused;
- internal Firebird consistency check (can't continue after bugcheck);
- internal Firebird consistency check (cannot continue after replication failure).

При подозрении на поломку необходимо выполнить проверку файла базы данных через утилиту **gfix**. Для проверки требуется монопольный доступ к базе (при сбоях проверки добавьте ключ **-ignore**):

```
/opt/RedDatabase/bin/gfix -v -full
```

Если ошибки не обнаружены, вывод команды будет пуст.

При обнаружении ошибок вывод команды будет иметь примерно следующий вид:

```
Summary of validation errors
Number of record level errors : 93
Number of data page errors : 25
Number of index page errors : 29
Number of database page errors : 14
```

Рекомендации:

- выполните проверку:

```
/opt/RedDatabase/bin/gfix -v -full
```

- проверьте лог **firebird.log** на предмет наличия ошибок или отправьте его в службу техподдержки;
- проверьте оборудование (диск, ОЗУ);
- выполните ремонт БД по результатам анализа проверки.

Запрос дополнительной информации для первичной диагностики проблемы при обращении на ЛТП1

Все показатели требуется смотреть в момент пиковой нагрузки на сервер или зависания. Сбор данных необходимо производиться под учетной записью с правами **администратора (sudo)**. Для диагностики необходимо установить следующие пакеты:

```
sudo dnf install numactl sysstat atop htop fio
```

1. Подробное описание проблемы:

- проблема присутствует постоянно или в определенные моменты;
- снижение производительности — общее или при определенных операциях;
- проводились ли какие-либо работы на сервере (обновление, замена оборудования и пр.).

2. Характеристики оборудования (ОЗУ, ЦПУ, hdd или ssd).

3. Вывод команды:

```
df -h
```

4. Вывод команды:

```
sudo gstat -h /path/to/db.fdb
```

5. Вывод команды:

```
sudo rdb_lock_print -l -n -o -d /path/to/db.fdb
```

6. Вывод команды:

```
sudo rdb_lock_print -q 30 -d /path/to/db
```

7. Вывод утилиты `top (atop, htop)` для определения нагрузки на ЦПУ.

8. Вывод команды:

```
free -m
```

9. Вывод утилиты `numastat` (при наличии **NUMA**-архитектуры):

```
numastat
```

10. Вывод утилиты `iostat` для оценки очереди к диску (выполнение займет ~1 минуту):

```
iostat -xzm -t 1 60 > /tmp/iostat.txt
```

При этом сформируется файл **/tmp/iostat.txt**, который необходимо предоставить.

11. Копия **firebird.log**, **firebird.conf**, **databases.conf** и **replication.conf** (при наличии) из каталога установки СУБД.

12. Копия **/var/log/messages** за интервал наблюдения проблемы и в течение месяца до нее.

13. Копия **/etc/fstab**.

14. Вывод команд:

```
sudo dmesg -T > /tmp/dmesg.txt
```

```
sudo journalctl -p err > /tmp/journalctl.txt
```

При этом создадутся файлы **dmesg.txt** и **journalctl.txt** в каталоге **/tmp**.

15. Проверить активность индексов запросом:

```
select rdb$index_name from rdb$indices
where (rdb$system_flag is null or rdb$system_flag = 0)
and RDB$INDEX_INACTIVE > 0
order by rdb$foreign_key nulls first
```

16. Протестировать производительность дисковой подсистемы, на которой расположена БД. Сделать это можно с помощью утилиты **fio**:

```
sudo fio /tmp/fio.conf
```

Файл **fio.conf** необходимо создать с помощью любого текстового редактора, например **nano**:

```
sudo nano /tmp/fio.conf
```

Пример конфигурационного файла:

```
[ro]
blocksize=16k
directory=/mnt/ssd
rw=randread
direct=1
size=15g
ioengine=psync
```

```
iodepth=5
runtime=600
iodepth=5
ioengine=psync

[wo]
blocksize=16k
directory=/mnt/ssd
rw=randwrite
direct=1
size=15g
ioengine=psync
iodepth=5
runtime=600
iodepth=5
ioengine=psync
```

Основные параметры конфигурационного файла:

- блоки `ro` и `wo` (на чтение и на запись) — размер блока 16Кб, можно установить 8Кб, если размер страницы в БД равен 8;
- `directory` — каталог, в котором будет проводиться тестирование;
- `rw-randread` — случайное чтение/запись;
- `direct=1` — флаг `o_direct`;
- `size=100` — размер файла 100Мб (из этого файла производится случайное чтение/запись);
- `runtime=60` — время тестирования;
- `iodepth=5` — глубина очереди к диску, которую **fiio** выдает в файл;
- `ioengine=psync` — используемые функции чтения/записи, которыми пишет СУБД.

Часто задаваемые вопросы по СУБД РЕД База Данных (FAQ)

+ Где скачать СУБД РЕД База Данных?

Единственным правильным источником для получения сборки является официальный сайт. Дистрибутивы из остальных источников использовать нежелательно, это может быть потенциально опасным.

Официальный источник (он же официальный сайт) — <https://reddatabase.ru/ru/downloads/>.

В разделе «Загрузки» можно увидеть все файлы, доступные для скачивания. Необходимо быть зарегистрированным пользователем. При этом, если учетная запись имеет соответствующие допуски, то список доступных для скачивания продуктов шире.

Чтобы релиз-кандидаты были доступны для скачивания, необходимо в своем профиле установить флаг для параметра «Показывать релиз-кандидаты», в этом случае можно видеть свежие версии, которые, вероятно, в последующем станут релизами. Это может потребоваться, если есть подозрения, что в релиз-кандидате была исправлена какая-либо серьезная ошибка. У каждой версии есть описание, можно просмотреть список изменений и исправленных проблем.

Клиентам выдается учетная запись (логин и пароль) с доступом к приобретенным продуктам.

Основные файлы на странице загрузки:

- **.bin** — установочный файл для ОС семейства Linux;
- **.tar.gz** — архив с файлами СУБД для ОС семейства Linux;
- **.exe** — установочный файл для ОС семейства Windows;
- **.zip** — архив с файлами СУБД для ОС семейства Windows;
- ***dbg-linux-x86_64.tar.gz** — отладочные символы для СУБД, установленной на ОС семейства Linux;
- ***dbg-windows-x86_64.zip** — отладочные символы для СУБД, установленной на ОС семейства Windows.

+ Где скачать документацию на СУБД РЕД База Данных?

Документация доступна на официальном сайте СУБД РЕД Базы Данных — <https://reddatabase.ru/ru/documentation/>.

Документация регулярно обновляется, исправляется и дополняется при обновлении СУБД, поэтому рекомендуется проверять новые версии документации.

+ Где найти видеоролики по СУБД РЕД База Данных?

Обучающие видеоролики доступны на официальных каналах СУБД РЕД База Данных:

- на **RuTube** — <https://rutube.ru/channel/29879950/>;
- на **Яндекс.Дзен** — <https://dzen.ru/reddatabase>.

+ Какие есть информационные ресурсы в Интернете по СУБД РЕД База Данных?

СУБД Ред База Данных в **Telegram** — <https://t.me/reddatabase>.

Канал СУБД Ред База Данных на **RuTube** — <https://rutube.ru/channel/29879950/>.

Канал СУБД Ред База Данных на **Яндекс.Дзен** — <https://dzen.ru/reddatabase>.

Список наиболее значительных изменений в новых версиях СУБД РЕД База Данных — <https://reddatabase.ru/ru/news/>.

Площадка для общения **СУБД РЕД База Данных** и **Firebird** — https://t.me/firebird_reddatabase.

+ Какую архитектуру СУБД выбрать?

Архитектура **Super** — рекомендована к использованию на версии **3.0.9+**. Имеет общий кеш, в версии **3.0** она стала действительно многопоточной, до этого она не была ориентирована на многоядерные процессоры. Страничный кеш общий, что дает увеличение производительности. Однако при аварийном завершении того самого единственного процесса, аварийно прекращает свою работу весь сервер.

Архитектура **Classic** — классическая архитектура, где каждый коннект происходит в отдельном процессе. Каждый процесс имеет собственный кеш данных, что является более надежным решением, но в то же время, в ряде случаев, показывает производительность ниже, чем архитектура **Super**.

Архитектура **Superclassic** — не рекомендована к использованию, в дальнейшем, возможно, будет изъята из дистрибутива.

+ Чем отличается текущая версия СУБД от Firebird?

Сравнение версии **3.0** с версией **2.6**, а также с **Firebird 3.0** находится на официальном сайте в разделе «Продукты» — <https://reddbatabase.ru/ru/comparison/>.

+ Чем отличается одна редакция СУБД от другой?

Основные различия описаны на официальном сайте в разделе «Продукты» — <https://reddbatabase.ru/ru/editions/>.

+ Как установить СУБД на Linux?

Необходимо предоставить права на выполнение. Далее можно выполнять установку в графическом либо в текстовом режиме.

Для назначения прав выполните команду:

```
sudo chmod a+x /path/to/rdb.bin
```

Для установки в графическом режиме выполните:

```
./rdb.bin
```

Далее следуйте инструкциям установщика.

Для установки в текстовом режиме выполните команду:

```
./rdb.bin --mode text
```

Далее следуйте инструкциям установщика.

+ В чем заключается типовая настройка СУБД после её установки?

Ключевые параметры конфигурационного файла **firebird.conf** для

временных файлов:

- **TempTableDirectories** — это каталог, в котором будут создаваться файлы временных таблиц, сейчас они создаются в каталоге **/tmp**.
- **TempDirectories** — список временных каталогов для сортировок, временных таблиц и т. п., разделяются точкой с запятой (;), используются по порядку. При отсутствии параметра используются переменные окружения **FIREBIRD_TMP**, **TEMP**, **TMP**, поэтому в **systemd** необходимо прописать параметр **FIREBIRD_TMP**. Рекомендуется упорядочить каталоги по скорости доступа, сначала оперативная память, затем **ssd**, затем **hdd**. Если оперативной памяти в избытке, первым можно указать каталог, смонтированный в ОЗУ. Рекомендуется временные каталоги размещать в разделах, отличных от размещения БД. После настройки необходимо следить за состоянием этих каталогов для своевременного исправления возникающих проблем. Пример:

```
TempDirectories = /var/tmp;/mnt/ssd;/mnt/hdd_tmp
```

- **TempCacheLimit** — используется для ограничения объема временного пространства, которое может быть кешировано в оперативной памяти. В этом пространстве будут сохраняться данные сортировок, буферов записи, небольших временных блоков перед материализацией и т.д. В целях увеличения производительности, желательно установить это значение больше длины временных файлов в каталогах **TempDirectories**, если это позволяет объем доступной оперативной памяти.

Ключевые параметры конфигурационного файла **firebird.conf** для кеширования:

- **DefaultDbCachePages** — количество страниц одной базы данных, находящихся в кеш-памяти одновременно. Супер-сервер использует единый кеш (по умолчанию 2048 страниц), т. е. **DefaultDbCachePages** общий для всех подключений. В архитектуре **Classic** используется свой кеш на каждый процесс, поэтому в архитектуре **Super** значение необходимо устанавливать больше. Измеряется в количестве страниц, поэтому от размера 1 страницы будет зависеть размер памяти, которая будет задействована под кеш.

Предполагается использовать четверть ОЗУ для нужд кеша СУБД.

Значение кеша задается параметром **page buffer** (см. вывод **gstat -h**) для каждой БД в отдельности, либо значением **DefaultDbCachePages** из конфигурационного файла **firebird.conf** для всего сервера, либо параметром **DefaultDbCachePages** в конфигурационном файле

databases.conf.

В приоритете - параметр `page buffer`, если `page buffer = 0`, то значение берется из конфигурационных файлов.

Архитектура **Super**:

```
MemorySize/(4*PageSize)
```

Архитектура **Classic**:

```
MemorySize/(4*ConnNum*PageSize)
```

Настройку можно уточнять по итогу мониторинга сервера: учесть объемы для сортировки, файловый кеш, ядро ОС, другие приложения. Самый важный параметр, на который стоит обратить внимание — это количество чтений (Read) страниц и количество Fetch страниц. **Fetch** — страница берется из кеша (из памяти), если в кеше её нет, то она читается с диска (что замедляет скорость работы). Чем ближе эти значения (write и cache) друг к другу, тем хуже. В идеальном прогетом кеше чтений почти не будет, будет только запись — это в архитектуре "супер". В классике кеш у коннектов разный, поэтому чтений там больше. Это связано с необходимостью записи страниц.

- **FileSystemCacheThreshold** — если значение указано больше, чем в **DefaultDBCachePages**, то включается использование файлового кэша. Если значение меньше — отключается. Предопределённых рекомендаций не существует. Подбор оптимального значения параметра зависит от диска и нагрузки, поэтому необходимо наблюдать за использованием диска и подбирать оптимальное значение. Для **Classic** оптимальный параметр обязателен, для **Super** менее критичен.

Далее перечислены ключевые параметры, касающиеся авторизации:

- **AuthServer** и **AuthClient** — набор методов аутентификации, разрешенных на сервере и клиенте:
 - Безопасная парольная (**Srp**) аутентификация;
 - Традиционная (**Legacy_Auth**) аутентификация;
 - Доверительная (**Win_Sspi**) аутентификация для ОС Windows;
 - Многофакторная (**Multifactor**) аутентификация с применением политик безопасности;
 - Доверенная аутентификация через механизм GSSAPI (**Gss**)
- **UserManager** — плагин для работы с базой данных пользователей:
 - Srp;

- Legacy_UserManager;
 - Multifactor_Manager.
 - Используется первый плагин из списка
- **DefaultUserManagers** — набор стандартных плагинов, в которых по умолчанию будет создаваться пользователь или меняться пароль у пользователя.

Прочие важные параметры **firebird.conf**:

- **ServerMode** — режим работы сервера (**Super, Classic, SuperClassic**), требуется перезапуск при изменении, при смене режима необходимо корректировать параметр кеша.
- **RemoteBindAddress** — позволяет привязать входящие соединения к определенному сетевому интерфейсу. При этом все входящие соединения через другие сетевые интерфейсы будут запрещены.
- **LockAcquireSpins** — количество попыток, которые будут сделаны в условном режиме захвата мьютекса таблицы блокировок. Этот параметр позволяет, вместо обращения к ядру, ждать некую атомарную переменную, что уменьшает количество тактов процессора на межпроцессорные вызовы.
- **LockHashSlots** — число слотов хэширования блокировок. Следует увеличивать если под нагрузкой в таблице блокировок много коллизий. См. позднее `rdb_lock_print -d <database> | <alias>` и `Hash lengths`. Как правило выставляется по максимуму, из опыта.



В чем заключается типовая настройка сервера после установки СУБД?

Для увеличения допустимого числа процессов СУБД и доступных им ресурсов:

- в `/etc/security/limits.conf` лимиты `nproc` и `nofile` для пользователя, от имени которого запускается СУБД (firebird), устанавливаются в 10000;
- параметр ядра `vm.max_map_count` необходимо увеличить до **256000**, написав в `/etc/sysctl.conf` строку:

```
vm.max_map_count=256000
```

Для немедленного применения настройки необходимо выполнить:

```
sysctl -p
```

- для systemd-систем в `/usr/lib/systemd/firebird.service` пропишите:

```
Environment = FIREBIRD_TMP=/path/to/tmp
```

- При наличии достаточного объема **RAM** можно смонтировать в память временный каталог, куда сервер будет выгружать файлы. Для этого в **/etc/fstab** необходимо строку:

```
tmpfs /tmp tmpfs size=32G 0 0
```

где **32G** — максимальный размер, до которого может расшириться временный каталог в памяти;

- отключить **transparent Huge Pages**, например, записав в **/etc/rc.local**:

```
echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
```

+ Как понять, какая редакция СУБД Ред База данных у меня установлена?

Необходимо изучить содержимое каталога установки СУБД.

Актуально для **3.0+**:

- если есть **replication.conf** — это промышленная редакция;
- если его нет, но есть **jvm.args** — это стандартная редакция;
- иначе открытая.

Если есть возможность подключиться к любой БД на сервере, то редакцию можно определить, выполнив запрос:

```
select rdb$get_context('SYSTEM', 'EDITION') from rdb$database;
```

+ Как узнать установленную версию СУБД?

Для определения установленной версии СУБД можно воспользоваться утилитой **gbak** с ключом **-z**, например:

```
/opt/RedDatabase/bin/gbak -z
```

Пример вывода команды:

```
gbak:gbak version LI-V3.0.9.0 RedDatabase 3.0 rc.5
```

```
(6dcadeecf66529566457de89e35603e10380fa17)
gbak: ERROR:requires both input and output filenames
gbak:Exiting before completion due to errors
```

где **LI-V3.0.9.0 RedDatabase 3.0 rc.5** — версия СУБД.

+ Как обновить СУБД на Linux (минорное обновление)?

Для обновления СУБД на Linux необходимо:

- остановить СУБД и проверить статус службы командой:

```
sudo systemctl stop firebird
systemctl status firebird
```

- Закомментировать алиас в **databases.conf** и через **isql** проверить, что к БД нет подключений:

```
select count(*) from mon$attachments;
exit;
```

- Проверить, что БД никто не использует утилитой **lsof** (от **root** или **sudo**), не должны выводиться pid'ы процессов:

```
sudo lsof /path/to/db.fdb
sudo lsof /opt/RedDatabase/security3.fdb
```

- **Переименовать /opt/RedDatabase** в **/opt/RedDatabase_<старая_версия>**. Старую версию можно узнать например с помощью утилиты **gbak**:

```
/opt/RedDatabase/bin/gbak -z
```

Переименование:

```
sudo mv /opt/RedDatabase /opt/RedDatabase_3.0.5.116
```

- установить новую версию СУБД, используя установочный бинарный файл, для этого назначить необходимые права:

```
sudo chmod a+x /path/to/rdb.bin
```

```
[REDACTED]
```

и запустить установку:

```
[REDACTED]
```

```
./rdb.bin
```

- Из каталога со старой версией СУБД скопировать в новый каталог необходимые конфигурации (**directories.conf**, **databases.conf**, **firebird.conf**, **fbtrace_sec.conf**, **fbtrace_dba.conf**, **replication.conf**, **fbjava.yaml**), дополнительные плагины (если такие имеются) и базы данных безопасности (**security3.fdb**, **java-security.fdb**);
- раскомментировать алиас базы в **databases.conf**:
- запустить РЕД Базу Данных и проверить статус службы (должно быть **active (running)**);

```
[REDACTED]
```

```
sudo systemctl start firebird  
systemctl status firebird
```

- проверить успешность подключения к базе по сетевому протоколу через **isql** (полный путь до файла БД):

```
[REDACTED]
```

```
/opt/RedDatabase/bin/isql -u sysdba -p masterkey localhost:alias
```

+ Как проверить БД на наличие ошибок?

Проверить БД на наличие ошибок можно утилитой **gfix**, требуется монопольный доступ к БД:

```
[REDACTED]
```

```
/opt/RedDatabase/bin/gfix -v -full /path/to/database.fdb
```

gfix может попытаться исправить некоторые некритичные ошибки. Для отмены исправления некритичных ошибок необходимо добавить ключ **-n**:

```
[REDACTED]
```

```
/opt/RedDatabase/bin/gfix -v -full -n /path/to/database.fdb
```

+ Какие ошибки можно считать критичными при проверке?

Все ошибки файла БД после проверки фиксируются в логе `/opt/RedDatabase/firebird.log`. Список наиболее критичных из них:

- `internal Firebird consistency check (cannot find tip page)` — нарушение некоторой целостности, например, не получается найти страницу, на которой отмечен статус транзакции, или находится не та страница;
- `Error while trying to {read from, write to} file` — не может прочитать/записать файл БД;
- `internal Firebird consistency check (decompression overran buffer)` — например, при декомпрессии какой-либо записи обнаруживается, что она превышает размер положенного буфера;
- `Pointer page (sequence NUM) {lost, inconsistent}` — потеряна страница указателей;
- `Missing index root page` — потеря **root page** для индекса;
- `Transaction inventory pages lost` — потерянная страница транзакций;
- `internal Firebird consistency check (cannot continue after replication failure)`.

+ В чем заключается типовой ремонт базы данных?

Типовой ремонт заключается в процедуре резервного копирования (`gbak -b -v -g`) и восстановления (`gbak -c -v -o`) поврежденной базы данных.

Ни в коем случае нельзя удалять поврежденную БД до полного завершения ремонта.

+ Что делать с ошибкой `invalid request BLR` при восстановлении резервной копии?

В данном случае можно попробовать перекомпилировать проблемные процедуры и еще раз сделать резервную копию. Если используется СУБД 3.0 и выше, восстановить такую резервную копию можно с ключом `-ig`, который восстановит процедуру с пустым **BLR**. Далее её необходимо перекомпилировать из исходников.

+ Что делать, если процесс создания резервной копии или процесс ее восстановления аварийно завершается при прохождении данных из определенной таблицы?

В данном случае БД можно восстановить с пропуском данных проблемной таблицы (или нескольких таблиц). Для этого используется ключ `-skip_d`.

Пример команды:

```
/opt/RedDatabase/bin/gbak -user SYSDBA -pas masterkey -b -v -g -skip_d  
'(TABLE_NAME_1|TABLE_NAME_2)' /path/to/database.fdb /path/to/backup.fbk -y  
/path/to/backup.log
```

При этом нужно помнить, что деактивируются внешние ключи, которые касаются таблиц с пропущенными данными. После восстановления БД таким способом необходимо загрузить недостающие данные и активировать вручную внешние ключи.

+ Как проверить наличие мусора в БД?

Утилитой `gstat` с ключом `-h`:

```
/opt/RedDatabase/bin/gstat -h /path/to/database.fdb
```

В выводе нужно обратить внимание на маркеры:

```
Oldest transaction, Oldest active, Oldest snapshot, Next transaction.
```

Разница между `Oldest transaction` и `Oldest snapshot` должна стремиться к единице. Однако зачастую разница этих параметров достигает больших значений, что негативно влияет на производительность. Для решения данной проблемы рекомендуется регулярная (ежедневная) сборка мусора.

+ Как принудительно собрать мусор в БД?

Принудительный сбор мусора в БД производится утилитой `gfix`:

```
/opt/RedDatabase/bin/gfix -sweep /path/to/database.fdb
```

+ Как сделать резервную копию базы данных?

Утилитой `gbak`, стандартная команда выглядит следующим образом:

```
/opt/RedDatabase/bin/gbak -user SYSDBA -pas masterkey -b -v -g
```

```
/path/to/database.fdb /path/to/backup.fbk -y /path/to/backup.log
```

Для того чтобы убедиться, что процесс создания резервного копирования успешно завершился, необходимо проверить его лог, формирование которого определяется ключом `-y` (см. команду создания резервной копии) — **/path/to/backup.log**. При успешном завершении лог заканчивается следующими строками:

```
gbak:writing names mapping  
gbak:closing file, committing, and finishing. 82432 bytes written  
gbak:stopped at Mon Aug 22 11:37:08 2022
```

При аварийном завершении или наличии ошибок следует разбираться в проблеме предметно.

+ Как восстановить резервную копию БД?

Восстановление резервной копии БД производится утилитой `gbak`. Стандартная команда выглядит следующим образом:

```
/opt/RedDatabase/bin/gbak -user SYSDBA -pas masterkey -c -v -o  
/path/to/backup.fbk /path/to/database.fdb -y /path/to/restore.log
```

При необходимости нужно использовать ключ `-ig` — при ошибках с восстановлением **BLR** хранимых процедур и функций процесс создания резервной копии продолжается, при этом в логе останется запись о пропуске **BLR**.

Для того чтобы убедиться, что восстановление резервной копии успешно завершено, следует проверить его лог, формирование которого определяется ключом `-y` (см. команду восстановления резервной копии) — **/path/to/restore.log**.

При успешном завершении лог заканчивается следующими строками:

```
gbak:fixing views dbkey length  
gbak:updating ownership of packages, procedures and tables  
gbak:adding missing privileges  
gbak:adjusting system generators  
gbak:finishing, closing, and going home  
gbak:adjusting the ONLINE and FORCED WRITES flags  
gbak:stopped at Mon Aug 22 11:52:21 2022
```

БД переводится в ONLINE-режим.

При аварийном завершении или наличии ошибок следует разбираться в проблеме предметно.

+ Как установить на сервер отладочные символы для РЕД Базы данных и для чего они нужны?

Отладочные символы — специальные данные о связи адресов и исходных кодов, некие описания разных переменных, какая инструкция к какой строке кода относится. Нужны для анализа зависаний и получения стеков, чтобы разработчик смог посмотреть работу процесса в виде текстовой информации, которую он видит у себя в процессе разработки.

Для установки необходимо скачать нужный архив с официального сайта. В названии архива присутствует обозначение **dbg** и версия отладочных символов, например, **RedDatabase-3.0.9-dbg-linux-x86_64.tar.gz**.

ВАЖНО!

Версия СУБД и версия символов должны строго совпадать.

Далее архив необходимо распаковать в нужное место. Для этого:

- перейдите в **mc**;
- перейдите в папку с отладочными символами;
- откройте архив как обычную директорию;
- в подкаталогах из архива существуют скрытые папки **debug**, в которых как раз и находятся все отладочные символы. Скопируйте все каталоги из архива в директорию **RedDatabase** (каждый каталог из архива в соответствующий ему каталог в /opt/RedDatabase).

На этом установка закончена.

+ Как настроить сбор дампов на Linux?

Необходимо использовать **systemd-coredump** для автоматического сбора дампов:

https://redos.red-soft.ru/base/manual/utilites/coredump/?sphrase_id=285533

+ Что такое репликация и как её настроить?

Репликация — непрерывное копирование (реплицирование) данных (изменений этих данных) с одного сервера (*master*) на один или несколько других (*slave*, реплика, зеркало).

В СУБД Ред База Данных реализовано два вида логической репликации:

- **Синхронная** — обеспечивает прямое подключение к *slave* и отправку изменений на него непосредственно до коммита транзакции;
- **Асинхронная** — предполагает запись в промежуточный журнал, который передается на зеркало и там применяется.

Примеры настройки репликации описаны [Руководстве администратора](#) в подразделе "Настройка системы репликации".

+ Как проверить работу асинхронной репликации?

Утилита **rdblogmgr** – запускать на мастере:

```
sudo rdblogmgr -u sysdba -p masterkey -d /db/replicated.fdb
```

В выводе можно увидеть следующие параметры:

- *current sequence* – текущий сегмент;
- *last modified* – последнее изменение;
- *total log size* – всего в логе байт в 1 сегменте;
- *free segments* – свободных сегментов 1, полных сегментов 0, архивных 0.

Пример вывода:

```
Log status:  
  
Current sequence: 8  
Last modified: 2023-12-13 16:48:19  
Total log size: 946259 bytes in 2 segments  
Free segments: 2, full segments: 0, archived segments: 0
```

Утилита **rdbreplmgr** – запускать на зеркале с ключом **-s**:

```
sudo rdbreplmgr -u sysdba -p masterkey /db/replica.fdb -s
```

В выводе команды можно увидеть:

- *Control file* — контрольный файл, содержит текущие счетчики пролитых логов, сколько осталось, сколько активных транзакций (бывает так, что в одном сегменте транзакция стартует, а в другом завершается, и в итоге сегмент можно удалить, когда все транзакции в нем завершены), при переинициализации его нужно удалять;
- *Current segment* — количество обработанных сегментов, должно постоянно расти;
- *Oldest segment* — количество старых сегментов, должно быть в значении **absent**;
- *Oldest segment* — в идеале **absent**;
- Total segments in the queue — количество сегментов в очереди на обработку, в идеале должно стремиться к 0.

Пример вывода:

```
Status for replica /opt/db/red.fdb:
Master database: 192.168.1.10:/opt/db/red.fdb
Master GUID: {71615D98-0551-48D3-EE94-9441AFF3FE04}
Archive directory: /mnt/repl/logsrdb/arch/
Control file: /mnt/repl/logsrdb/arch/{71615D98-0551-48D3-EE94-9441AFF3FE04}
Current segment: 16 (as of 2020-06-17 09:36:35)
Oldest segment: absent
Total segments in the queue: 1
```

+ Почему при использовании репликации важно наличие первичных ключей в таблице?

Если на мастере создали таблицу без первичного ключа и уникального индекса, то репликационные журналы перестают вливаться на зеркале, при этом отображается ошибка вида:

```
my-server (slave) Tue Dec 22 11:24:35 2020
Database: /path/to/database.fdb
ERROR: Table MY_TABLE has no unique key
```

В конфигурационном файле **replication.conf** на мастере необходимо прописать параметр `exclude_without_pk = true`, при этом таблицы без

первичного ключа не будут реплицироваться на базу зеркала.

+ Что такое трейс и как его запустить?

Трейс (аудит) — инструмент для аудита событий, происходящих внутри БД, позволяет просматривать:

- подключения (отключения) к БД;
- все стадии выполнения запросов, включая счетчики производительности;
- события безопасности;
- работа со службами;
- выполнение хранимых процедур, функций и триггеров;
- работа транзакций, параметры транзакций и многое другое.

Информации может быть много, поэтому в конфигурационном файле доступны различные фильтры (например, по подключению или по времени). Аудит можно сделать на уровне сервера (системный), а можно открыть интерактивную сессию аудита.

- системный аудит — запуск происходит с помощью раскомментирования и изменения параметра `enabled` в конфигурационном файле `fbtrace.conf` — `enabled = true;`
- интерактивный сеанс аудита — запуск происходит с помощью утилиты `rdbtracemgr` и заранее подготовленного конфигурационного файла.

```
rdbtracemgr <параметры_подключения> <действия>
[<параметры>]
```

Пример команды:

```
/opt/RedDatabase/bin/rdbtracemgr -SE service_mgr -START -NAME my_trace -
CONFIG my_cfg.txt
```

+ Как найти неактивные индексы?

Для этого необходимо подключиться к базе данных и выполнить SQL-запрос в любом менеджере БД:

```
select rdb$index_name from rdb$indices where (rdb$system_flag is null or
rdb$system_flag = 0) and RDB$INDEX_INACTIVE > 0;
```

+ Как активировать все неактивные индексы?

С помощью SQL-запроса в любом менеджере БД:

```
EXECUTE BLOCK AS
DECLARE VARIABLE stmt VARCHAR (1000);
BEGIN
for select 'ALTER INDEX '||rdb$index_name ||' ACTIVE;'
from rdb$indices
where (rdb$system_flag is null or rdb$system_flag = 0)
and RDB$INDEX_INACTIVE > 0
order by rdb$foreign_key nulls first
into :stmt do EXECUTE STATEMENT :stmt;
END
```

+ Как получить структуру индексов определенной таблицы?

С помощью SQL-запроса в любом менеджере БД:

```
SELECT S.RDB$INDEX_NAME, S.RDB$FIELD_NAME FROM RDB$INDICES I
NATURAL JOIN RDB$INDEX_SEGMENTS S WHERE
I.RDB$RELATION_NAME='MYTABLE' ORDER BY S.RDB$FIELD_POSITION
```

где вместо 'MYTABLE' необходимо указать имя нужной таблицы.

+ Как получить список ролей или пользователей, которые имеют права на вставку в таблицу?

С помощью SQL-запроса в любом менеджере БД:

```
SELECT RDB$USER FROM RDB$USER_PRIVILEGES WHERE
RDB$RELATION_NAME='COUNTRY' AND RDB$PRIVILEGE='I' AND
RDB$USER_TYPE=13 AND RDB$OBJECT_TYPE=0
```

+ Как найти все подключения старше суток?

С помощью SQL-запроса в любом менеджере БД:

```
SELECT MON$ATTACHMENT_ID FROM MON$ATTACHMENTS  
WHERE DATEDIFF (HOUR, MON$TIMESTAMP, CURRENT_TIMESTAMP) > 24
```

Для их принудительного завершения:

```
DELETE FROM MON$ATTACHMENTS  
WHERE DATEDIFF (HOUR, MON$TIMESTAMP, CURRENT_TIMESTAMP) > 24
```

+ Как принудительно завершить подключения с транзакциями старше суток?

С помощью SQL-запроса в любом менеджере БД:

```
DELETE FROM MON$ATTACHMENTS  
WHERE MON$ATTACHMENT_ID IN (  
SELECT MON$ATTACHMENT_ID FROM MON$TRANSACTIONS  
WHERE DATEDIFF (HOUR, MON$TIMESTAMP, CURRENT_TIMESTAMP) > 24)
```

+ Как принудительно завершить соединения с запросами старше суток?

С помощью SQL-запроса в любом менеджере БД:

```
DELETE FROM MON$ATTACHMENTS  
WHERE MON$ATTACHMENT_ID IN (  
SELECT MON$ATTACHMENT_ID FROM MON$STATEMENTS  
WHERE DATEDIFF (HOUR, MON$TIMESTAMP, CURRENT_TIMESTAMP) > 24)
```

+ Как найти самые читающие с диска запросы?

С помощью SQL-запроса в любом менеджере БД:

```
SELECT MON$SQL_TEXT FROM MON$STATEMENTS
WHERE MON$STAT_ID = (SELECT FIRST 1 MON$STAT_ID FROM MON$IO_STATS
WHERE MON$STAT_GROUP=3 ORDER BY MON$PAGE_READS DESC)
```

Для их завершения:

```
DELETE FROM MON$ATTACHMENTS
WHERE MON$STAT_ID = (SELECT FIRST 1 MON$STAT_ID FROM MON$IO_STATS
WHERE MON$STAT_GROUP=1 ORDER BY MON$PAGE_READS DESC)
MON$RECORD_STATS
```

+ Как найти запросы, которые потребляют больше всего памяти?

С помощью SQL-запроса в любом менеджере БД:

```
SELECT MON$SQL_TEXT FROM MON$STATEMENTS
WHERE MON$STAT_ID = (SELECT FIRST 1 MON$STAT_ID FROM
MON$MEMORY_USAGE WHERE MON$STAT_GROUP=3 ORDER BY
MON$MEMORY_USED DESC)
```

+ Как найти клиентские подключения, которые потребляют больше всего памяти?

С помощью SQL-запроса в любом менеджере БД:

```
SELECT MON$REMOTE_ADDRESS, MON$REMOTE_PROCESS
FROM MON$ATTACHMENTS
WHERE MON$STAT_ID = (SELECT FIRST 1 MON$STAT_ID FROM
MON$MEMORY_USAGE WHERE MON$STAT_GROUP=1 ORDER BY
MON$MEMORY_USED DESC)
```

+ Какие утилиты доступны "из коробки" для работы с файлом базы данных?

Для этих целей можно использовать утилиту `isql`. С помощью нее можно создавать базу данных, подключаться и выполнять SQL-запросы в базе данных.

Доступна в любой редакции СУБД, входит в состав дистрибутива.

ISQL — это утилита командной строки для работы с базами данных РЕД Базы Данных при помощи языка структурированных запросов (Structured Query Language — SQL).

Пример команды для подключения:

```
/opt/RedDatabase/bin/isql -u sysdba -p masterkey localhost:employee
```

где:

- *sysdba* — имя пользователя, от которого подключаемся к БД;
- *masterkey* — пароль этого пользователя;
- *employee* — алиас базы данных.

Подробное описание возможностей данной утилиты доступно в [Руководстве администратора](#) в разделе «**Утилита ISQL**».

+ Почему нужно следить за наличием свободного места на разделе с базой данных?

Нехватка места на разделе с базой данных чревата поломкой самой базы, поэтому рекомендуется на разделе иметь хотя бы 30% свободного места (минимум 10%). При этом в логе могут появляться ошибки вида:

```
my.server.local (master) Sun Sep 20 04:43:29 2020
Database: /path/to/database.fdb
ERROR: Log file /path/to/repl/logs/database.fdb.log-142 write failed (error 28)
my.server.local (master) Sun Sep 20 04:43:29 2020
Database: /path/to/database.fdb
WARNING: Slave is to be detached after error. Replica may become
inconsistent at this point.
```

+ Что нужно отслеживать на сервере СУБД?

1. Наличие мусора и выполнение регулярной сборки мусора.

2. Наличие свободного места в ключевых разделах:

- раздел с БД;
- в корневом разделе;
- в разделе с установленной СУБД;
- наличие свободного места в каталоге **/tmp**.

3. Загрузку ОЗУ (параметр **available** в выводе **free -m**).

4. Лог **firebird.log** (сообщения об ошибках, остановках сервиса).

5. Системный лог (сообщения об аварийном завершении).

6. Проверка лога создания резервной копии и тестового восстановления резервной копии.

7. Проверка наличия долгих транзакций.

+ Как посмотреть размер объектов в БД?

Для проверки размера объектов в БД необходимо собрать статистику по файлу базы данных с помощью команды:

```
/opt/RedDatabase/bin/gstat -user sysdba -password masterkey -a -r -s  
my_database.fdb > /home/user/Загрузки/full_gstat.txt
```

По окончании выполнения команды будет сформирован файл **/home/user/Загрузки/full_gstat.txt**, используемый далее для подсчета размера объектов.

Пример вывода статистики:

```
Database "/databases/my_base.fdb"  
Gstat execution time Fri Dec 22 11:15:11 2023  
  
Database header page information:  
Flags 0  
Generation 164  
System Change Number 0  
Page size 8192  
Server RedDatabase  
ODS version 12.3  
Oldest transaction 126  
Oldest active 127  
Oldest snapshot 127
```

```
Next transaction      127
Autosweep gap        1
Sequence number       0
Next attachment ID    27
Implementation        HW=AMD/Intel/x64 little-endian OS=Linux CC=gcc
Shadow count         0
Page buffers          0
Next header page      0
Database dialect      3
Creation date         Dec 22, 2023 10:36:15
Attributes            force write
```

Variable header data:

```
Sweep interval:      20000
Database GUID: {EA77BD7A-868B-4459-F5A2-3E578634467F}
*END*
```

...

DOCUMENT (132)

Primary pointer page: 204, Index root page: 205

Total formats: 1, used formats: 1

Average record length: 524.10, total records: 1000000

Average version length: 0.00, total versions: 0, max versions: 0

Average fragment length: 0.00, total fragments: 0, max fragments: 0

Average unpacked length: 1010.00, compression ratio: 1.93

Pointer pages: 45, data page slots: 72056

Data pages: 72056, average fill: 92% Primary pages: 72056, secondary pages: 0, swept pages: 0

Empty pages: 3, full pages: 72052

Fill distribution:

0 - 19% = 3

20 - 39% = 0

40 - 59% = 0

60 - 79% = 0

80 - 99% = 72053

Для определения размера таблицы в выводе следует обратить внимание на параметр **Data pages**, затем умножить количество **Data pages** нужной таблицы на размер страницы базы данных (размер страницы определяется параметром **Page size** из заголовка базы данных). В результате будет получен размер данных таблицы в байтах без учета блобов и индексов.

Например, таблица **DOCUMENT** по статистике имеет **72056 страниц**, размер одной страницы — **8192 Б**. Размер таблицы равен 72056 x 8192 = ~563 МБ, среднее заполнение 92%.

Размер блобов

Для определения размера блобов в выводе следует обратить внимание на строку **Blobs**, содержащую значение параметра **total length**. Если в таблице есть блобы, то в статистике таблицы будет присутствовать следующая строка:

```
Blobs: 1289124, total length: 12577030488, blob pages: 863868
```

Здесь параметр **total length** равен **12577030488** — т.е. примерно 11,7 ГБ.

Размер индексов

Для определения размера индексов следует обратить внимание на значение параметра **leaf buckets**, затем умножить значение параметра **leaf buckets** на размер страницы базы данных (размер страницы определяется параметром **Page size** из заголовка базы данных). В результате будет получен размер индекса. Например:

```
Index RDB$PRIMARY27 (0)
  Root page: 72963, depth: 3, leaf buckets: 1205, nodes: 1000000
  Average node length: 9.71, total dup: 0, max dup: 0
  Average key length: 6.72, compression ratio: 1.00
  Average prefix length: 3.01, average data length: 3.74
  Clustering factor: 999982, ratio: 1.00
  Fill distribution:
    0 - 19% = 1
    20 - 39% = 0
    40 - 59% = 0
    60 - 79% = 0
    80 - 99% = 1204
```

Здесь параметр **leaf buckets** имеет размер **1205 страниц**, размер одной страницы (из статистики) — **8192 Б**. Размер индекса равен $1205 \times 8192 = \sim 9,4$ МБ.