

Установка Firebird 3 на современные версии Linux – CentOS 8 и Ubuntu 19

© Василий Сидоров, iBase.ru, 16.03.2020

Описывается минимальный набор действий, необходимых для установки СУБД Firebird версии 3.0 на новые дистрибутивы Linux. Для примеров выбраны два общедоступных и широко распространённых дистрибутива – CentOS 8 (Mandrake/RedHat) и Ubuntu 19 (Debian).

Для «доставки» дистрибутива Firebird на целевую систему в этом руководстве выбран вариант загрузки по ссылке с официального сайта проекта (firebirdsql.org). Таким образом можно устанавливать не только официальные выпуски Firebird, но и промежуточные сборки, что полезно для тестирования, а иногда – необходимо в конкретной ситуации.

В этом руководстве «sudo команда» используется для обозначения команд, требующих привилегий суперпользователя (root).

Предполагается, что уже установлена в минимальном варианте и настроена базовая система, имеющая доступ публичным репозиториям или к их локальным копиям.

Предполагается, что у читателя есть базовые знания о Linux, СУБД Firebird и её архитектурах (Classic/SuperClassic/SuperServer) и что читатель ознакомился с документацией актуальной версии. Как минимум – с информацией о выпуске (Release Notes).

Быстрая установка без подробностей

Этот раздел – памятка для тех, кто уже читал данное руководство, но подзабыл некоторые детали или для тех, кто действительно спешит.

Редактируем файл `/etc/sysctl.conf`, добавляя строку:

```
vm.max_map_count = 256000
```

Сохраняем файл и применяем настройку:

```
sudo sysctl -p /etc/sysctl.conf
```

Дальнейшие инструкции различаются для CentOS 8 и Ubuntu 19, но ССЫЛКА и КАТАЛОГ обозначают ссылку для загрузки с официального сайта проекта Firebird и каталог, который будет создан при распаковке дистрибутива в `/tmp`.

CentOS 8

```
sudo yum -y install epel-release
sudo yum -y makecache
sudo yum -y install libicu libtommath tar
ln -s libncurses.so.5 \
/usr/lib64/libncurses.so.5
```

```
ln -s libtommath.so.1 \  
/usr/lib64/libtommath.so.0  
curl -L ССЫЛКА|tar -zxС /tmp
```

Ubuntu 19

```
sudo apt-get -y install libncurses5 libtommath1  
ln -s libtommath.so.1 \  
/usr/lib/x86_64-linux-gnu/libtommath.so.0  
wget -O- ССЫЛКА|tar -zxС /tmp
```

Собственно установка СУБД Firebird:

```
cd /tmp/КАТАЛОГ  
sudo ./install.sh
```

Если вы хотите лучше понимать, что чего служат все эти действия – продолжаем чтение.

Планирование дискового пространства

На сервере СУБД рекомендуется выделять отдельные разделы для временных файлов (/tmp), файлов баз данных и локальных бэкапов.

К временным относятся lock-файлы, файлы сортировок, файлы «материализации» глобальных временных таблиц (ГТТ) и таблиц мониторинга. Файлы сортировок и глобальных временных таблиц расположены в /tmp, файлы mon\$-таблиц и lock-файлы – в /tmp/firebird.

Файлы сортировок «удаляются» (unlink) сразу после создания, поэтому их нельзя «увидеть» в листинге каталога – только в списке описателей (handles) процесса (помечены как deleted):

```
sudo ls -lhF /proc/`pgrep firebird`/fd
```

В листинге псевдокаталога /proc/.../fd/ отображаются симлинки, а фактическую информацию о файле даёт:

```
sudo stat -L /proc/`pgrep firebird`/fd/НОМЕР
```

где НОМЕР – описатель (дескриптор) интересующего файла.

Вместо вызова «pgrep исполняемый-файл» можно сразу подставить идентификатор интересующего процесса.

Временные файлы могут быть очень большими, поэтому для /tmp рекомендуется выделять не менее 20-30 ГБ. Следует учитывать, что размер файлов сортировок зависит только от объёма данных, явно или неявно сортируемых в запросе и один-единственный пользователь может «создать» гигабайты временных файлов.

Раздел для файлов баз данных должен вмещать файлы всех баз плюс, как минимум, копию файла самой большой базы. Следует учитывать рост файлов баз в перспективе на несколько лет вперёд.

Раздел локальных бэкапов должен вмещать, как минимум, по одному архиву бэкапов всех баз плюс бэкап самой большой базы. Желательно,

чтобы на этом разделе было и место для восстановления самой большой базы. Следует учитывать рост бэкапов и архивов бэкапов в перспективе на несколько лет вперёд.

Настройка системного параметра `vm.max_map_count`

Сервер СУБД Firebird 3.0 динамически выделяет и освобождает системную память, что может приводить к её фрагментации и отказам при попытках выделения новых блоков памяти. В `firebird.log` протоколируются или прямые ошибки выделения памяти (`munmap` и код ошибки 12 в детализации) или косвенные – отказы в подключении из-за нехватки памяти (в детализации – `ENOMEM` и код ошибки 11).

Возможный сценарий фрагментации системной памяти – одновременное отключение от суперсервера большого числа пользователей и последующее подключение новых.

Фрагментацию системной памяти контролирует параметр `vm.max_map_count`, по умолчанию – 64К. Рекомендуется увеличить его значение вчетверо:

```
sudo sysctl vm.max_map_count=256000
```

Чтобы новое значение устанавливалось и при перезагрузке системы, добавляем в файл `/etc/sysctl.conf` строчку:

```
vm.max_map_count = 256000
```

Желательно сделать комментарий, чтобы была понятна причина изменения параметра. Можно сначала отредактировать файл, а затем применить сохранённые в нём установки:

```
sudo sysctl -p /etc/sysctl.conf
```

Создание пользователя `firebird`

Скрипты установки создают, если требуется, специального пользователя (`firebrd`), но иногда лучше создать его заранее. Например, идентификатор пользователя должен отличаться от умалчиваемого (84) для удобной передачи файлов по NFS.

Создаём сначала группу, а потом и пользователя (команды сложены, UID и GID – идентификаторы пользователя и группы):

```
sudo groupadd firebird &&  
sudo useradd -M -b /opt -s /sbin/nologin \  
-g firebird -u UID firebird
```

Для пользователя не создаётся домашний каталог (`-M`), но указываются базовый каталог (`-b`). Оболочка `nologin`, (опция `-s`) исключает стандартную регистрацию в системе. Если требуется определённый идентификатор группы, то первой командой станет:

```
sudo groupadd -g GID firebird
```

Установка необходимых пакетов

Исполняемые файлы СУБД Firebird 3.0 Linux зависят от библиотек ncurses (libncurses.so.5), ICU (без привязки к версии и без отображения в выводе ldd) и tommath (libtommath.so.0). Для загрузки и распаковки архива сборки потребуются утилиты gzip, tar и curl или wget. Версии ICU, gzip, tar и curl/wget – несущественны.

Работа с пакетами зависит от системы и от используемого в системе пакетного менеджера, поэтому рассмотрим их поочередно.

CentOS 8

CentOS 8 использует новый пакетный менеджер – dnf и он же «прозрачно» вызывается по команде yum. Поскольку для наших целей между ними нет разницы – в примерах будет yum.

Обновляем кэш метаданных: `sudo yum makecache`

Пакет libtommath находится в отдельном E(xtra)P(ackages for)E(nterprise)L(inux) репозитории, поэтому проверяем, что он уже подключен:

```
yum -C repolist
```

Опция «только из кэша» (-C или --cache-only) используется, чтобы исключить ненужные проверки и загрузки, ускорив работу yum. Если в списке нет epel-репозитория – устанавливаем его и обновляем кэш метаданных:

```
sudo yum install epel-release &&  
sudo yum makecache
```

Подтверждаем запросы, при необходимости сверяя значения rpm-ключей с уже известными из доверенного источника.

Если возникли проблемы при загрузке метаданных репозитория с https-ресурсов, то редактируем файл /etc/yum.repos.d/epel.repo, заменяя https:// на http:// и повторяем команду обновления кэша.

Проверяем статус нужных пакетов (команда сложена, в примере вывода отфильтрован 32-разрядный пакет):

```
yum -C list \  
ncurses libicu libtommath \  
gzip tar curl wget |  
grep -v i686
```

Installed Packages		
curl.x86_64	7.61.1-11.el8	@anaconda
gzip.x86_64	1.9-9.el8	@anaconda
ncurses.x86_64	6.1-7.20180224.el8	@anaconda
Available Packages		
libicu.x86_64	60.3-1.el8	BaseOS
libtommath.x86_64	1.1.0-1.el8	epel

```
tar.x86_64          2:1.30-4.el8          BaseOS
wget.x86_64         1.19.5-8.el8_1.1     AppStream
```

Видим, что `curl`, `gzip` и `ncurses` размещены в псевдорепозитории установщика (`anaconda`), а `tar` – исключён из минимальной установки системы. Мажорные версии `libncurses` и `libtommath` больше, чем требуется: 6 и 1 вместо 5 и 0, соответственно. Если один и тот же пакет и установлен и доступен – для него выпущено обновление. Устанавливаем недостающие пакеты:

```
sudo yum install \
    libicu libtommath tar
```

Ubuntu 19

Для управления пакетами предназначены утилиты `apt`, `apt-get` и `apt-cache`. Первая рассчитана на интерактивную работу, а две последние – на использование в скриптах. Имена пакетов немного другие и включают версию.

Проверяем статус нужных пакетов (команда сложена, пример вывода сокращён и отфильтрованы 32-разрядные пакеты):

```
apt list libncurses? libicu?? libtommath? \
    gzip tar curl wget |
    grep -v i386
curl 7.65.3-1
gzip 1.10-0 [upgradable...]
libicu63 63.2-2 [installed]
libncurses5 6.1
libncurses6 6.1 [installed,automatic]
libtommath1 1.1.0
tar 1.30 [installed]
wget 1.20.3 [installed]
```

Пакеты, для которых в квадратных скобках указано `installed/upgradable` – установлены. Доступен, но не установлен `ncurses5`, вместо `curl` установлен `wget`. Устанавливаем недостающие пакеты:

```
sudo apt-get install \
    libncurses5 libtommath1
```

Создание симлинков

Поскольку `libtommath.so.1` и `libncurses.so.6` обратно совместимы с `libtommath.so.0` и `libncurses.so.5`, то для Firebird достаточно создать симлинки на имеющиеся версии библиотек.

Находим `libtommath.so.1` (`libncurses.so.?` расположены в этом же каталоге):

```
find /usr -name libtommath.so.1
```

CentOS:

```
/usr/lib64/libtommath.so.1
```

Ubuntu:

```
/usr/lib/x86_64-linux-gnu/libtommath.so.1
```

Создаём симлинки.

CentOS:

```
sudo ln -s libtommath.so.1 \  
/usr/lib64/libtommath.so.0  
sudo ln -s libncurses.so.6 \  
/usr/lib64/libncurses.so.5
```

Ubuntu:

```
sudo ln -s libtommath.so.1 \  
/usr/lib/x86_64-linux-gnu/libtommath.so.0
```

Проверяем результат (команда сложена, примеры вывода сокращены):

```
ls -lhF \  
$(dirname `find /usr -name libtommath.so.1`) |  
grep "lib\(ncurses\|tommath\)\.so\."
```

CentOS:

```
libncurses.so.5 -> libncurses.so.6*  
libncurses.so.6 -> libncurses.so.6.1*  
libncurses.so.6.1*  
libtommath.so.0 -> libtommath.so.1*  
libtommath.so.1 -> libtommath.so.1.1.0*  
libtommath.so.1.1.0*
```

Ubuntu:

```
libncurses.so.5 -> libncurses.so.5.9  
libncurses.so.5.9  
libncurses.so.6 -> libncurses.so.6.1  
libncurses.so.6.1  
libtommath.so.0 -> libtommath.so.1  
libtommath.so.1 -> libtommath.so.1.1.0  
libtommath.so.1.1.0
```

Загрузка дистрибутива СУБД Firebird

На официальном сайте проекта Firebird (firebirdsql.org) публикуются ссылки на дистрибутивы «официальных» выпусков (releases) и «ежедневных» сборок (snapshot build).

Официальные выпуски для линукса доступны в виде архивов (tar.gz) и пакетов deb/rpm, а сборки – только в виде архивов. Мы будем рассматривать «общий установщик» (generic installer из tar.gz).

Архив сборки требуется загрузить и распаковать, но мы совместим оба этих процесса. Распаковку будем делать в /tmp.

curl:

```
curl -L ССЫЛКА | tar -zxС /tmp
```

wget:

```
wget -O- ССЫЛКА | tar -zxС /tmp
```

По умолчанию curl посылает загружаемые данные на стандартный вывод, но не обрабатывает перенаправления и мы добавляем «-L», а wget, наоборот: обрабатывает перенаправления, но записывает данные в файл и мы ставим «-O-». Для tar указываем использование gzip-фильтра и каталог, в который будет выполнена распаковка. По завершении процесса появится каталог вида Firebird-3.0.5.33220-0.amd64 с тремя файлами: install.sh, buildroot.tar.gz и manifest.txt.

Установка Firebird

В ходе предварительной подготовки мы отрегулировали значение системного параметра `vm.max_map_count`, проверили наличие и установили библиотеки ICU, ncurses и tommath. Убедились в правильности версий ncurses и tommath (`libncurses.so.5` и `libtommath.so.0`) и создали необходимые симлинки. Создали, если это необходимо, пользователя `firebird` с заданным идентификатором.

Собственно установка делается очень просто. Переходим в каталог, куда был распакован архив дистрибутива Firebird, проверяем и, при необходимости, устанавливаем флаг «исполняемый» скрипту `install.sh`:

```
chmod +x install.sh
```

запускаем инсталляционный скрипт:

```
sudo ./install.sh
```

нажатием клавиши Enter подтверждаем начало установки, а по получении запроса – вводим пароль `sysdba`.

Скрипт установки автоматически запускает `systemd`-юнит `firebird-superserver` (умалчиваемая архитектура Firebird 3.0). Сервис Firebird будет работать с параметрами по умолчанию: страничный кэш в 2048 страниц (на базу), буфер сортировок в 64 МБ (общий) и подключение только клиентов третьей версии. Просмотр параметров `firebird.conf`:

```
grep -v ^# firebird.conf | grep -v ^$
```

Новые значения из `firebird.conf` будут активированы только после перезапуска сервиса Firebird.

При подборе значений параметров следует учитывать, что есть три основных «потребителя»: страничный кэш (для базы), буфер сортировок (общий) и память, выделенная сервером для клиентских подключений. Управлять можно только первыми двумя – объём памяти клиентских подключений зависит от количества и текста кэшированных запросов, их планов и задействованных в запросах объектов базы. Оценка памяти клиентских подключений делается только эмпирически и может меняться при изменении клиентских приложений и/или объектов базы.

Для архитектуры SuperServer на хостах с малым объёмом памяти (до 12-16 ГБ) не следует выделять страничному кэшу и буферу сортировки (суммарно) более трети-четверти от общего объёма ОЗУ.

Если количество баз не фиксировано и может меняться – общий объём памяти страничного кэша следует делить на максимальное количество баз, которые могут быть на сервере. Размер страничного кэша задаётся в страницах и его надо отдельно пересчитывать в байты.

Для переключения на архитектуру Classic требуется, как минимум, явно указать `ServerMode` в `firebird.conf`, уменьшить там же страничный кэш (не более 2К), уменьшить буфер сортировок (суммарный допустимый объём всех сортировок в памяти, поделённый на максимальное количество подключений), остановить и запретить `firebird-superserver`, разрешить и запустить `firebird-classic.socket`.

Использование архитектуры SuperClassic в Firebird 3.0 не имеет особого смысла: «надёжность» SuperServer и такой же общий буфер сортировок. Отдельный страничный кэш у каждого подключения и такие же, как для архитектуры Classic, «потери» на синхронизацию.

Следует помнить, что в Firebird 3.0 часть параметров (страничный кэш, размеры лок-файла, размер хэш-таблицы и некоторые другие) можно задавать в `databases.conf`, индивидуально для каждой базы.

В архитектуре SuperServer полезно, например, задать небольшое значение `DefaultDbCachePages` в `firebird.conf` и установить индивидуальные страничные кэши нужным базам в `databases.conf`.

Чтобы не выключить, случайно, использование файлового кэша в архитектуре SuperServer, следует установить большое значение параметра `FileSystemCacheThreshold` (по умолчанию – всего 64К). Этот параметр используется как триггер и вы можете безопасно указывать любое значение, которое будет превышать максимальное значение `DefaultDbCachePages` или даже объём установленной в системе памяти.

Вопросы? Пишите на support@ibase.ru.