Утвержден

РУСБ.10144-01-УД

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

ДОКУМЕНТО-ОРИЕНТИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ

Руководство администратора

РУСБ.10144-01 95 01

Листов 94

2013

АННОТАЦИЯ

Настоящий документ является руководством администратора документо-ориентированной системы управления базами данных специального назначения РУСБ.10144-01 (далее по тексту — ДОСУБД).

В документе приведено назначение ДОСУБД и описание ее установки и настройки.

Рассмотрены вопросы повышения отказоустойчивости и производительности с помощью технологий репликации и шардинга.

Описаны серверные службы ДОСУБД и утилиты командной строки, включая инструменты резервного копирования и восстановления данных.

Приводится информация о взаимодействии администратора с СЗИ, в том числе управление учетными записями пользователей и выполнение санкционированных операций по управлению правилами разграничения доступа как с помощью интерактивного командного интерфейса доступа к ДОСУБД, так и с помощью прикладного программного интерфейса ДОСУБД на языках C и C++.

Описано тестирование мандатного разграничения доступа ДОСУБД.

Приведен список сообщений об ошибках.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ

ДОСУБД предназначена для создания информационных и управляющих систем в составе автоматизированных систем, обрабатывающих информацию ограниченного распространения.

ДОСУБД функционирует под управлением операционной системы специального назначения Astra Linux Special Edition РУСБ.10015-01 (далее по тексту — ОС СН).

ДОСУБД по своим функциональным возможностям соответствует документо-ориентированной системе управления базами данных с открытыми исходными текстами MongoDB.

ДОСУБД обеспечивает выполнение требований специального нормативного документа ФСТЭК России «Руководящий документ. Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей» по 3 уровню контроля.

ДОСУБД обеспечивает решение следующих задач, связанных с обработкой информации ограниченного доступа, на основе интеграции средств разграничения доступа ДОСУБД к защищаемым объектам объектам ДОСУБД со средствами защиты информации из состава ОС СН:

– поддержка конфиденциальности хранимых данных с использованием дискреционного разграничения доступа;
– поддержка конфиденциальности хранимых данных с использованием мандатного разграничения доступа;
– документо-ориентированная организация хранения данных;
– манипулирование данными и обеспечение эффективного доступа к ним;
– поддержка отказоустойчивости и масштабируемости;
– обеспечение копирования и восстановления данных;
– регистрация попыток доступа к защищаемым объектам ДОСУБД;
– регистрация действий по изменению правил разграничения доступа.

2. СОСТАВ ДОСУБД

В состав ДОСУБД входят следующие компоненты:

– «Сервер ДОСУБД»;

– «Клиенты ДОСУБД»;

– «Прикладной программный интерфейс ДОСУБД»;

– «Тесты мандатного разграничения доступа в ДОСУБД».

Данные компоненты являются составными частями ДОСУБД и взаимодействуют между собой в процессе ее функционирования, и располагаются в соответствующих deb-пакетах.

2.1. Сервер ДОСУБД

Компонент «Сервер ДОСУБД» содержит набор программ сервера ДОСУБД:

– `mongod` — основная сервисная служба ДОСУБД, реализующая сервер СУБД, и обеспечивающая обработку запросов, управление форматом данных и выполняющая фоновые операции обслуживания БД;

– `mongos` — сервисная служба ДОСУБД, управляющая шардингом, обеспечивающая обработку запросов на уровне приложения и определяющая расположение требуемых данных в кластере для выполнения запрашиваемой операции.

2.2. Клиенты ДОСУБД

Компонент «Клиенты ДОСУБД» содержит набор клиентских программ ДОСУБД:

– `mongo` — интерактивный командный интерфейс доступа к ДОСУБД, реализованный на JavaScript и предоставляющий интерфейс для системных администраторов, а также для разработчиков для проверки запросов непосредственно к серверу ДОСУБД. Утилита `mongo` также является полнофункциональной JavaScript-средой для работы с ДОСУБД;

– `mongodump` — клиентская утилита получения резервных копий с сервера ДОСУБД в двоичном формате BSON;

– `mongoexport` — клиентская утилита выгрузки с сервера ДОСУБД данных в текстовых форматах JSON или CSV;

– `mongofiles` — утилита прямого доступа к файлам ДОСУБД без необходимости запуска службы сервера;

– `mongoinmport` — клиентская утилита загрузки в указанную БД ДОСУБД данных в текстовых форматах JSON или CSV, выгруженных ранее с помощью утилиты `mongoexport` или подготовленных другими средствами;

– `mongorestore` — клиентская утилита импорта данных (восстановления) из ре-

зервной копии в двоичном формате BSON, созданной ранее с помощью утилиты `mongodump`, в указанную БД ДОСУБД;

– `mongostat` — клиентская утилита получения значений счетчиков операций в БД ДОСУБД. Утилита `mongostat` предоставляет информацию, агрегированную по типу доступа (вставка, запрос, обновление, удаление), что облегчает понимание распределения нагрузки на сервер.

2.3. Прикладной программный интерфейс ДОСУБД

Компонент «Прикладной программный интерфейсы ДОСУБД» содержит библиотеки прикладного программного интерфейса доступа к ДОСУБД:

– `libmongoclient` — библиотека C++ драйвера для доступа к ДОСУБД;

– `libbson`, `libmongoc` — библиотека C драйвера для доступа к ДОСУБД.

2.4. Тесты мандатного разграничения доступа в ДОСУБД

Компонент «Тесты мандатного разграничения доступа в ДОСУБД» содержит сценарии и эталоны результатов для тестирования функциональных возможностей по мандатному разграничению доступа, размещаемые в каталоге `/usr/share/mongodb/setests/` при установке соответствующего компоненту пакета, указанного в п. 3.1.

3. УСТАНОВКА ДОСУБД

ДОСУБД функционирует под управлением ОС СН.

При установке ДОСУБД потребуется установить библиотеки, размещающиеся на следующих оптических носителях из состава ОС СН:

– РУСБ 10015-01 12 01-1 Текст программы. Загрузочный модуль. Часть 1;

– РУСБ 10015-01 12 01-1 Текст программы. Загрузочный модуль. Часть 2. Средства разработки.

3.1. Состав загрузочного модуля ДОСУБД

Загрузочный модуль поставляется в виде пакетов, которые приведены ниже:

– `mongodb-server` — сервер ДОСУБД;

– `mongodb-clients` — клиенты ДОСУБД;

– `mongodb-dev` — прикладной программный интерфейс ДОСУБД на языке C++;

– `mongo-c-driver-dev` — прикладной программный интерфейс ДОСУБД на языке C;

– `mongodb` — мета-пакет ДОСУБД;

– `mongodb-se-test` — тесты мандатного разграничения доступа в ДОСУБД.

3.2. Установка клиентской части ДОСУБД

Установка выполняется от имени суперпользователя `root` или пользователя, имеющего право выполнять монтирование внешних носителей и установку пакетов ПО с использованием `sudo`. Для установки клиентской части ДОСУБД необходимо выполнить следующую последовательность действий:

1) Выполнить монтирование в корневую ФС оптического диска с дистрибутивом ДОСУБД с помощью команды:

```
sudo mount /dev/cdrom /mnt
```

2) Перейти в каталог, в котором находятся пакеты дистрибутива ДОСУБД:

```
cd /mnt
```

3) Выполнить следующую команду:

```
sudo dpkg -i mongo-clients*
```

Будет выдано сообщение сообщение об ошибке из-за проблем зависимостей.

4) Для завершения установки необходимо выполнить команду:

```
sudo apt-get -f install
```

3.3. Установка серверной части ДОСУБД

Установка выполняется от имени суперпользователя `root` или пользователя, имеющего право выполнять монтирование внешних носителей и установку пакетов ПО с ис-

пользованием sudo.

Для установки серверной части ДОСУБД необходимо установить клиентскую часть ДОСУБД (см. 3.2) и выполнить следующую последовательность действий:

1) Выполнить монтирование оптического диска с дистрибутивом ДОСУБД в корневую ФС с помощью команды:

```
sudo mount /dev/cdrom /mnt
```

2) Перейти в каталог, в котором находятся пакеты дистрибутива ДОСУБД:

```
cd /mnt
```

3) Выполнить следующую команду:

```
sudo dpkg -i mongodb-server*
```

Будет выдано сообщение сообщение об ошибке из-за проблем зависимостей.

4) Для завершения установки необходимо выполнить команду:

```
sudo apt-get -f install
```

3.4. Установка прикладного программного интерфейса ДОСУБД

Установка выполняется от имени суперпользователя root или пользователя, имеющего право выполнять монтирование внешних носителей и установку пакетов ПО с использованием sudo. Для установки пакетов прикладного программного интерфейса ДОСУБД необходимо выполнить следующую последовательность действий:

1) Выполнить монтирование в корневую ФС оптического диска с дистрибутивом ДОСУБД с помощью команды:

```
sudo mount /dev/cdrom /mnt
```

2) Перейти в каталог, в котором находятся пакеты дистрибутива ДОСУБД:

```
cd /mnt
```

3) Выполнить следующие команды для установки пакетов прикладного программного интерфейса ДОСУБД на языке C++ и C, соответственно:

```
sudo dpkg -i mongodb-dev*
sudo dpkg -i mongo-c-driver-dev*
```

Будет выдано сообщение сообщение об ошибке из-за проблем зависимостей.

4) Для завершения установки необходимо выполнить команду:

```
sudo apt-get -f install
```

3.5. Установка набора тестов мандатного разграничения доступа в ДОСУБД

Установка выполняется от имени суперпользователя root или пользователя, имеющего право выполнять монтирование внешних носителей и установку пакетов ПО с использованием sudo. Для установки набора тестов мандатного разграничения доступа в ДОСУБД необходимо установить серверную часть ДОСУБД (см. 3.3) и выполнить следующую последовательность действий:

1) Выполнить монтирование в корневую ФС оптического диска с дистрибутивом ДОСУБД с помощью команды:

```
sudo mount /dev/cdrom /mnt
```

2) Перейти в каталог, в котором находятся пакеты дистрибутива ДОСУБД:

```
cd /mnt
```

3) Выполнить следующую команду:

```
sudo dpkg -i mongodb-se-test*
```

Будет выдано сообщение сообщение об ошибке из-за проблем зависимостей.

4) Для завершения установки необходимо выполнить команду:

```
sudo apt-get -f install
```

4. НАСТРОЙКА ДОСУБД

Процесс, являющийся клиентом ДОСУБД, запускается с мандатным контекстом (мандатными атрибутами) установленным специальным PAM-модулем. Указанный модуль использует библиотеку безопасности `libparsec` для извлечения информации о пользователе, необходимой для правильной настройки контекста безопасности PARSEC (см. документ РУСБ.10015-01 97 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1»). В такой контекст входят: мандатный контекст процесса, набор полномочий процесса, правила протоколирования процесса. Контекст безопасности передается по наследству процессам-потомкам. Таким образом, совокупность процессов, принадлежащих одному пользователю и имеющих общий контекст безопасности процесса, является пользовательской сессией, а контекст безопасности сессии — контекстом безопасности пользователя.

Для обеспечения надежности функционирования механизмов дискреционного и мандатного разграничения доступа пользователи ДОСУБД должны пройти процедуру идентификации и аутентификации.

ВНИМАНИЕ! Для обеспечения корректного функционирования мандатного разграничения доступа должно быть обеспечено однозначное соответствие пользователей ДОСУБД и пользователей ОС СН.

При реализации функций мандатного разграничения доступа ОС СН может функционировать в одном из двух режимов:

1) Локальный режим, в котором вся необходимая информация о мандатных атрибутах и привилегиях пользователей извлекается из соответствующих локальных БД подсистемы безопасности PARSEC ОС СН. В этом случае необходимо обеспечить доступ на чтение к данным БД для учетной записи службы ДОСУБД (см. 4.2).

2) Режим ЕПП, в котором вся необходимая информация о мандатных атрибутах и привилегиях пользователей извлекается из БД ALD подсистемы безопасности PARSEC ОС СН. В этом случае необходимо выполнить соответствующую настройку (см. 4.3). Дополнительная информация о ЕПП приведена в документе РУСБ.10015-01 95 01 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора».

4.1. Настройка /etc/mongodb.conf

Значения параметров, определяющих порядок функционирования ДОСУБД, устанавливаются в конфигурационном файле `/etc/mongodb.conf`. Для установки значений всех параметров в этом файле используется следующая форма:

```
<Параметр> = <Значение>
```

Параметры, значения которых должны быть определены для нормального функционирования ДОСУБД, и их описание представлены в таблице 1.

Таблица 1

Параметр	Описание
verbose	Значение по умолчанию: false. Увеличивает количество внутренней отчетности, возвращаемое в стандартном потоке вывода или в файле журнала, генерируемом с именем logpath. Для включения verbose или предоставления возможности увеличения количества выводимой информации с vvvv необходимо установить следующие значения параметров: verbose = true vvvv = true
port	Значение по умолчанию: 27017. Определяет TCP-порт для экземпляра mongod или mongos, на котором осуществляется ожидание запросов на соединение от клиентов. В UNIX-подобных системах требуются права суперпользователя root для доступа к портам с номерами меньше чем 1024
bind_ip	Значение по умолчанию: 127.0.0.1. Определяет IP-адрес для экземпляра mongod или mongos, на котором осуществляется ожидание запросов на соединение от клиентов. Для привязки экземпляра mongod или mongos к множеству IP-адресов можно указать список их значений, разделенных запятой
keyFile	Значение по умолчанию: None. Определяет файл ключей, в котором хранится аутентификационная информация. Параметр используется только для соединений между членами набора реплики. **ВНИМАНИЕ!** Определение значения параметра автоматически включает режим аутентификации для всех пользователей. Перед использованием данного параметра необходимо наличие хотя бы одного зарегистрированного пользователя, имеющего доступ к администрированию
auth	Значение по умолчанию: false. Задание значения true включает режим аутентификации пользователей при доступе к БД. Управление пользователями осуществляется с использованием командной оболочки ДОСУБД. Если пользователи не существуют, то через локальный интерфейс можно получить доступ к БД до создания первого пользователя. **ВНИМАНИЕ!** Перед включением режима аутентификации необходимо наличие хотя бы одного зарегистрированного пользователя, имеющего доступ к администрированию

Окончание таблицы 1

Параметр	Описание
`logpath`	Значение по умолчанию: `/var/log/mongodb/mongodb.log`. Определяет имя файла журнала, который будет содержать всю диагностическую информацию. Если значение параметра не определено, то протоколирование осуществляется в стандартный поток вывода. Если значение параметра `logappend` не установлено в `true`, то файл журнала будет перезаписываться при каждом перезапуске процесса
`logappend`	Значение по умолчанию: `true`. Для добавления новых записей в конец файла журнала для данного параметра должно быть установлено значение `true`. Если значение параметра не определено, то протоколирование осуществляется в стандартный поток вывода. Если значение параметра `logappend` не установлено в `true`, то файл журнала будет перезаписываться при каждом перезапуске процесса
`dbpath`	Значение по умолчанию: `/var/lib/mongodb`. Устанавливает значение для каталога, назначенного экземпляру `mongod` для сохранения его данных. Если не указано, `mongod` будет осуществлять поиск файлов данных в каталоге `/db/data`
`replSet`	Значение по умолчанию: `none`. Используется для настройки набора реплики. В качестве значения указывается имя набора реплики. Все хосты должны иметь одинаковое имя набора реплики.

4.2. Настройка доступа к локальным БД PARSEC

В случае использования локальной информации подсистемы безопасности PARSEC необходимо обеспечить доступ на чтение к соответствующим БД мандатных атрибутов и привилегий пользователей `/etc/parsec/macdb` и `/etc/parsec/capdb` подсистемы безопасности PARSEC ОС СН. Дополнительная информация о подсистеме безопасности PARSEC ОС СН приведена в документе РУСБ.10015-01 97 01-1. Для предоставления указанных прав доступа учетная запись службы ДОСУБД должна быть добавлена в списки контроля доступа (ACL — Access Control List) следующим образом:

1) Установить ACL по умолчанию для каталога с БД мандатных атрибутов подсистемы безопасности PARSEC для локальных пользователей. Данный ACL будет автоматически устанавливаться ОС СН для нового файла локального пользователя, создаваемого в каталоге с БД мандатных атрибутов подсистемы безопасности PARSEC при первичной установке мандатных атрибутов для локального пользователя. ACL позволит пользователю `mongodb`, от имени которого функционирует ДОСУБД, получить доступ на чтение к файлу с мандатными атрибутами нового локального пользователя:

```
#setfacl -d -m u:mongodb:r /etc/parsec/macdb
```

2) Установить ACL, позволяющий пользователю mongodb, от имени которого функционирует ДОСУБД, получить доступ на чтение к файлам всех пользователей, существующих к моменту настройки ДОСУБД в каталоге с БД мандатных атрибутов подсистемы безопасности PARSEC для локальных пользователей:

```
#setfacl -R -m u:mongodb:r /etc/parsec/macdb
```

3) Установить ACL, позволяющий пользователю mongodb, от имени которого функционирует ДОСУБД, получить доступ на чтение содержимого каталога с БД мандатных атрибутов подсистемы безопасности PARSEC для локальных пользователей:

```
#setfacl -m u:mongodb:rx /etc/parsec/macdb
```

4) Установить ACL по умолчанию для каталога с БД привилегий безопасности PARSEC для локальных пользователей. Данный ACL будет автоматически устанавливаться ОС СН для нового файла локального пользователя, создаваемого в каталоге с БД привилегий подсистемы безопасности PARSEC при первичной установке привилегий для локального пользователя. ACL позволит пользователю mongodb, от имени которого функционирует ДОСУБД, получить доступ на чтение к файлу с привилегиями PARSEC нового локального пользователя:

```
#setfacl -d -m u:mongodb:r /etc/parsec/capdb
```

5) Установить ACL, позволяющий пользователю mongodb, от имени которого функционирует ДОСУБД, получить доступ на чтение к файлам всех пользователей, существующих к моменту настройки ДОСУБД в каталоге с БД привилегий подсистемы безопасности PARSEC для локальных пользователей:

```
#setfacl -R -m u:mongodb:r /etc/parsec/capdb
```

6) Установить ACL, позволяющий пользователю mongodb, от имени которого функционирует ДОСУБД, получить доступ на чтение содержимого каталога с БД привилегий подсистемы безопасности PARSEC для локальных пользователей:

```
#setfacl -m u:mongodb:rx /etc/parsec/capdb
```

ВНИМАНИЕ! Помимо файла журнала, определенного параметром logpath в конфигурационном файле /etc/mongodb.conf (см. 4.1), ДОСУБД осуществляет регистрацию событий с использованием средств протоколирования подсистемы безопасности PARSEC ОС СН. Регистрация событий осуществляется на основе шаблона, определенного в файле /etc/parsec/mlog/events_mongo.conf. Запрещено изменение указанного файла вручную.

4.3. Настройка для работы в ЕПП

Для работы ДОСУБД в ЕПП необходимо обеспечить доступ к серверу домена, выполнив в соответствии с документацией на ОС СН (см. руководства `man` по утилитам ОС СН) следующие действия:

1) Создать в БД ALD с помощью утилиты администрирования `ald-admin` ALD-принципала, соответствующего настраиваемой ДОСУБД. Принципал создается с автоматически сгенерированным случайным ключом:

```
ald-admin service-add mongodb/server.my_domain.org
```

При этом `server.my_domain` — полное доменное имя системы, под управлением которой должен функционировать конкретный экземпляр ДОСУБД.

2) ввести данного принципала в группу сервисов `mac` для предоставления принципалу ДОСУБД доступа к мандатным атрибутам и привилегиям пользователей, хранящимся в БД ALD, выполнив команду:

```
ald-admin sgroup-svc-add mongodb/server.my_domain.org --sgroup=mac
```

3) Создать файл ключа Kerberos в системе, под управлением которой должна функционировать ДОСУБД. В соответствии с документацией на ОС СН создание данного файла может быть выполнено с помощью утилиты администрирования ALD `ald-client` с помощью выполнения следующей команды:

```
ald-client update-svc-keytab mongodb/server.my_domain.org
      --ktfile="/var/lib/mongodb/mongodb.keytab"
```

Полученный файл должен быть доступен пользователю, от имени которого должна функционировать ДОСУБД на чтение:

```
chown mongodb /var/lib/mongodb/mongodb.keytab
chmod 400 /var/lib/mongodb/mongodb.keytab
```

ВНИМАНИЕ! Путь к файлу ключа Kerberos задается переменной окружения `KRB5_KTNAME` при запуске службы сервера и имеет значение по умолчанию `/var/lib/mongodb/mongodb.keytab` в скрипте запуска службы `/etc/init.d/mongodb`. Значение может быть переопределено установкой описанной переменной в файле `/etc/default/mongodb`.

5. СЛУЖБЫ ДОСУБД

Основными компонентами ДОСУБД являются:

– mongod — служба, являющаяся процессом ядра СУБД (см. 5.1);

– mongos — служба, являющаяся контроллером и маршрутизатором запросов для кластера шардинга (см. 5.2);

– mongo — командная оболочка ДОСУБД (см. 8.1).

5.1. Служба mongod

Процесс службы mongod является первичным для ДОСУБД. Он обрабатывает запросы к данным, управляет форматами данных и выполняет служебные операции в фоновом режиме.

5.1.1. Основные опции

Опции командной строки mongod приведены в таблице 2. Опции в первую очередь полезны для тестирования. В общем случае необходимо использовать опции конфигурационного файла для управления поведением ДОСУБД, которые полностью совместимы с опциями mongod.

Таблица 2

Параметр	Описание
--help, -h	Показывает справку по вызову команды
--version	Показывает версию
--config <filename> -f <filename>	Указывает конфигурационный файл, который используется для управления функционированием. Несмотря на то, что опции конфигурационного файла эквиваленты доступным в командной строке, конфигурационный файл является предпочтительным методом управления функционированием ДОСУБД. **ВНИМАНИЕ!** Конфигурационный файл должен иметь кодировку ASCII. mongod не поддерживает не ASCII кодировку, включая UTF-8
--verbose, -v	Увеличивает количество внутренней отчетности, возвращаемое в стандартном потоке вывода или в файле журнала генерируемом с именем logpath. Используется форма -v для управления количеством выводимой информации посредством включения опции несколько раз, например, -vvvvv

Продолжение таблицы 2

Параметр	Описание
`--quiet`	Выполняет экземпляр `mongod` в режиме ограничения выводимой информации. Опция не выводит следующие выходные данные: – результаты выполнения команд ДОСУБД, включая `drop`, `dropIndexes`, `diagLogging`, `validate`, и `clean`; – результаты работы репликации; – данные об установленных соединениях; – данные о закрытых соединениях
`--port <port>`	Указывает TCP-порт для ожидания `mongod` клиентских соединений. По умолчанию `mongod` ожидает соединения с использованием порта `27017`. В UNIX-подобных системах необходимы привилегии суперпользователя `root` для использования портов с номерами меньше `1024`
`--bind_ip <ip address>`	IP-адрес, который использует процесс `mongod` для ожидания входящих соединений. По умолчанию `mongod` ожидает соединения на интерфейсе `127.0.0.1`. Для привязки экземпляра определения `mongod` или `mongos` к множеству IP-адресов можно указать список их значений, разделенных запятой.
`--maxConns <number>`	Указывет максимальное число одновременных соединений, которые может принять `mongod`. Значение не будет действовать, если оно превышает установленное в ОС СН пороговое значение для максимального числа соединений. **ВНИМАНИЕ!** Невозможно установить значение `maxConns`, превышающее `20000`
`--objcheck`	Принуждает `mongod` проверять все полученные запросы клиентов для предотвращения вставки клиентами в БД некорректных документов. Для объектов с высокой степенью вложенности поддокументов опция `--objcheck` может привести к небольшому снижению производительности. Можно использовать опция `--noobjcheck` для отключения проверки объекта. Опция `--objcheck` используется по умолчанию
`--noobjcheck`	Отключает проверку документа, которую ДОСУБД выполняет для всех входящих BSON-документов
`--logpath`	Определяет имя файла журнала, который будет содержать всю диагностическую информацию. Если значение параметра не определено, то протоколирование осуществляется в стандартный поток вывода. Если дополнительно не указана опция `--logappend`, то файл журнала будет перезаписываться при каждом перезапуске процесса
`--logappend`	Используется для добавления службой `mongod` новых записей в конец файла журнала вместо перезаписывания содержимого журнала при перезапуске процесса

Продолжение таблицы 2

Параметр	Описание
`--syslog`	Направляет все протоколирование в журнал `syslog` вместо стандартного потока вывода или файла журнала **ВНИМАНИЕ!** Опция `--syslog` не может использоваться одновременно с `--logpath`
`--pidfilepath <path>`	Указывает положение файла, содержащего PID (идентификатор процесса) `mongod`. Опция полезна для отслеживания процесса `mongod` в сочетании с опцией `mongod --fork`. Если опция `--pidfilepath` не указана, то `mongos` не создает PID-файл
`--keyFile`	Определяет файл ключей, в котором хранится аутентификационная информация. Параметр используется только для соединений между членами набора реплики **ВНИМАНИЕ!** Использование опции автоматически включает режим аутентификации для всех пользователей. При использовании данной опции необходимо наличие хотя бы одного зарегистрированного пользователя, имеющего доступ к администрированию
`--nounixsocket`	Отключает ожидание соединений с использованием сокетов UNIX. `mongod` ожидает входящие соединения с использованием сокетов UNIX, если не указана опция `--nounixsocket`, опция `--bind_ip` не задана или опцией `--bind_ip` задано значение `127.0.0.1`.
`--unixSocketPrefix <path>`	Определяет путь к сокету UNIX. Если опция не имеет значения `mongod` создает сокет, используя префикс `/tmp`. ДОСУБД всегда создает сокет UNIX и осуществляет ожидание на нем входящих соединений если не установлена опция `--nounixsocket`, опция `--bind_ip` не задана или опцией `--bind_ip` задано значение `127.0.0.1`
`--fork`	Разрешает `mongod` работать в режиме службы, переключая процесс в фоновый режим. Данный режим функционирования является обычным, за исключением тестирования
`--auth`	Включает режим аутентификации пользователей при доступе к БД. Управление пользователями осуществляется с использованием командной оболочки ДОСУБД. Если пользователи не существуют, то через локальный интерфейс можно получить доступ к БД до создания первого пользователя. **ВНИМАНИЕ!** Перед включением режима аутентификации необходимо наличие хотя бы одного зарегистрированного пользователя, имеющего доступ к администрированию
`--cpu`	Включает режим протоколирования использования процессорного времени. `mongod` генерирует данные каждые 4 секунды и записывает результат в стандартный поток вывода или файл журнала, указанный опцией `--logpath`
`--dbpath`	Устанавливает значение для каталога, назначенного экземпляру `mongod` для сохранения его данных. Если не указано, `mongod` будет осуществлять поиск файлов данных в каталоге `/db/data`

Продолжение таблицы 2

Параметр	Описание
`--diaglog <value>`	Создает журнал диагностических сообщений для поиска и устранения неисправностей и записи данных о серьезных ошибках. ДОСУБД создает такие файлы в каталоге, заданном опцией `--dbpath`. Имя файла начинается со строки `diaglog` и завершается временем начала протоколирования в виде шестнадцатеричной строки. Значение опции определяет степень детализации. Возможны следующие значения: – `0` — протоколирование отключено; – `1` — протоколирование операций записи; – `2` — протоколирование операций чтения; – `3` — протоколирование операций чтения и записи; – `7` — протоколирование операций записи и некоторых операций чтения. Возможно использовать утилиту `mongosniff` для повторного просмотра протокола и его исследования. Например, для файла `/data/db/diaglog.4f76a58c` может быть использована следующая команда: `mongosniff --source DIAGLOG /data/db/diaglog.4f76a58c` Установка значения `0` отключает запись диагностического протокола. Однако, `mongod` будет продолжать держать файл протокола открытым. Для переименования или удаления файла диагностического протокола необходимо завершить работу `mongod`

Продолжение таблицы 2

Параметр	Описание

`--directoryperdb`	Изменяет шаблон хранения каталога данных для сохранения файлов каждой БД в отдельном каталоге. Данная опция создает для каждой БД каталоги внутри каталога, указанного опцией `--dbpath`. Совместно с файловой системой и конфигурированием устройств возможно настроить ДОСУБД для хранения данных на наборе отдельных устройств для увеличения скорости записи и емкости диска.
	ВНИМАНИЕ! Для существующих БД использование опции `--directoryperdb` требует выполнения миграции в каталоги.
	Пример
	Существует каталог `dbpath` со следующим содержимым:
	`journal`
	`mongod.lock`
	`local.0`
	`local.1`
	`local.ns`
	`test.0`
	`test.1`
	`test.ns`
	Для использования опции `--directoryperdb` необходимо модифицировать `dbpath` следующим образом:
	`journal`
	`mongod.lock`
	`local/local.0`
	`local/local.1`
	`local/local.ns`
	`test/test.0`
	`test/test.1`
	`test/test.ns`

`replSet`	Значение по умолчанию: `none`. Используется для настройки набора реплики. В качестве значения указывается имя набора реплики. Все хосты должны иметь одинаковое имя набора реплики
`--journal`	Включает операции протоколирования в специальный журнал для обеспечения согласованности данных
`--journalOptions, <arguments>`	Используется для тестирования. Не рекомендуется использовать в обычном режиме, т.к. может быть нарушена целостность БД

Продолжение таблицы 2

Параметр	Описание
--journalCommitInterval <value>	Определяет максимальное разрешенное mongod время между операциями записи в специальный журнал. Возможные значения от 2 до 300 миллисекунд. Уменьшение значения приводит к увеличению размера специального журнала и увеличению нагрузки на диск. Значение по умолчанию 100 миллисекунд для одного блочного устройства (например, физического тома, RAID-массива или тома LVM), содержащего и файлы данных и специальный журнал. Если для ведения специального журнала выделено отдельное блочное устройство, то значение по умолчанию 30 миллисекунд. Для выполнения более частой записи в журнал можно указать j:true. В этом случае при выполнении операции записи с j:true mongod сокращает journalCommitInterval до третьей части установленного значения
--ipv6	Включает поддержку IPv6. Позволяет клиентам подключаться к mongod с использованием сетей IPv6. По умолчанию в mongod поддержка IPv6 отключена в mongod и утилитах 8
--jsonp	Разрешает JSONP-доступ через интерфейс HTTP
--noauth	Отключает аутентификацию. Используется по умолчанию. **ВНИМАНИЕ!** Опция не должна использоваться после завершения настройки ДОСУБД
--nohttpinterface	Отключает интерфейс HTTP
--nojournal	Отключает запись в специальный файл
--noprealloc	Отключает предварительное выделение файлов данных запись в специальный файл. Сокращает время запуска, но может привести к значительным проблемам с производительностью при выполнении нормальных операций
--noscripting	Отключает обработчика сценариев
--notablescan	Запрещает операции, которые требуют сканирования таблиц
--nssize <value>	Определяет размер по умолчанию для пространства имен файлов. Опция не используется для существующих пространств имен файлов. Максимальное значение 2047 МБайт. Значение по умолчанию 16 МБайт, что порождает приблизительно 24000 пространств имен. Каждая коллекция, как и каждый индекс, считается пространством имен
--nojournal	Отключает запись в специальный файл

Продолжение таблицы 2

Параметр	Описание
`--profile <level>`	Изменяет уровень профилирования ДОСУБД, управляя выводом информации о производительности в стандартный поток вывода или в файл журнала. Доступны следующие уровни: – `0` — профилирование отключено; – `1` — включает только медленные операции; – `2` — включает все операции; По умолчанию профилирование отключено. Профилирование ДОСУБД может снизить производительность
`--quota`	Включает ограничение для максимального числа файлов данных, которые богут быть в БД. При запуске с опцией `--quota` максимальное число файлов в БД `8`. Управление квотой производится опцией `--quotaFiles`
`--quotaFiles <number>`	Изменяет ограничение для максимального числа файлов данных, которые богут быть в БД. Опция требует использования опции `--quota`. По умолчанию значение `--quotaFiles` равно `8`
`--rest`	Делает доступным прикладной программный интерфейс REST

Продолжение таблицы 2

Параметр	Описание
`--repair`	Запускает процесс восстановления на всех БД. Опция эквивалентна остановке и выполнению команды `repairDatabase` для всех БД. **ВНИМАНИЕ!** При выполнении нормальных операций команда `repairDatabase`, включая `db.repairDatabase()` в командной оболочке `mongo`, и опция `--repair` в `mongod` используются только для сжатия файлов БД и освобождения дискового пространства. Необходимо убедиться, что данные операции удаляют или не сохраняют любые поврежденные данные в процессе восстановления. При проведении восстановления для члена реплики необходимо использовать копию данных при наличии и не использовать `repairDatabase`. При использовании специального журнала обеспечения согласованности, как правило не возникает необходимости выполнения repairDatabase. В случае некорректного выключения сервер сможет восстановить файлы данных в исходное состояние автоматически. При использовании опции `--repair` и при наличии данных в специальном файле журнала, `mongod` выдаст отказ в запуске. Необходимо запустить `mongod` без опции `--repair` и позволить `mongod` восстановить данные из специального журнала. Данное действие завершится быстрее и позволит получить более целостный и согласованный набор данных. Для продолжения операции восстановления, несмотря на файлы специального журнала, необходимо завершить `mongod` корректно и выполнить перезапуск с опцией `--repair`. Опция `--repair` копирует данные из файлов источника в новые файлы данных с путем `repairpath` и затем заменяет оригинальные файлы данных восстановленными файлами данных. Если каталог `repairpath` находится на том же устройстве, что и `dbpath`, то можно использовать опцию `--repair` без ущерба целостности набора данных
`--repairpath <path>`	Указывает корневой каталог содержащий файлы данных ДОСУБД для использования с опцией `--repair`. По умолчанию используется каталог `_tmp` в `dbpath`.

Продолжение таблицы 2

Параметр	Описание
--setParameter <options>	Указывает опцию для настройки при запуске. Используется несколько опций с опцией --setParameter. Команда БД setParameter позволяет получить доступ ко многим из параметров. Опция --setParameter поддерживает следующие значения: – enableLocalhostAuthBypass; – enableTestCommands; – journalCommitInterval; – logLevel; – logUserIds; – notablescan; – quiet; – replApplyBatchSize; – replApplyBatchSize; – replIndexPrefetch; – supportCompatibilityFormPrivilegeDocuments; – syncdelay; – textSearchEnabled; – traceExceptions
--slowms <value>	Определяет зачение «slow» для опции --profile. БД протоколирует все медленные запросы в журнал даже при выключенном профилировании. При включенном профилировщике БД он осуществляет запись в коллекцию system.profile
--smallfiles	Включает режим, в котором ДОСУБД использует меньший размер файла по умолчанию. Опция --smallfiles сокращает начальный размер файлов данных и ограничивает их 512 Мбайтами. Кроме того --smallfiles сокращает размер каждого файла специального журнала с 1 ГБ до 128 МБ. Необходимо использовать --smallfiles при наличии большого числа БД, каждая из которых содержит малое количество данных. Использование опции --smallfiles может привести mongod к созданию большого числа файлов, что может сказаться на производительности для больших БД
--shutdown	При использовании в сценариях управления опция --shutdown корректно и безопасно завершает процесс mongod. При вызове mongod с данной опцией необходимо использвать опцию --dbpath явно или в конфигурационном файле с опцией --config

Окончание таблицы 2

Параметр	Описание
`--syncdelay <value>`	`mongod` записывает данные в специальный журнал очень быстро и медленно в файлы данных. Опция `--syncdelay` управляет периодом сброса ДОСУБД данных в файлы БД с использованием операции fsync. Значение по умолчанию 60 секунд. Рекомендуется использовать значение по умолчанию практически для всех случаев. Команда `serverStatus` выдает отчет о состоянии фонового потока сборас данных на диск с использованием поля `backgroundFlushing`. Опция `syncdelay` не влияет на файлы журнала. **ВНИМАНИЕ!** При установке для опции `--syncdelay` значения `0` ДОСУБД не синхронизирует файлы, отображенные в память с диском. Не рекомендуется устанавливать значение 0 для работающих ДОСУБД
`--sysinfo`	Возвращает системную диагностическую информацию и выходит. Информация содержит размер страницы памяти, число физических страниц и число доступных физических страниц
`--upgrade`	При необходимости изменяет формат данных на диске для файлов, указанных опцией `--dbpath` до последней версии. Опция используется только в случае операций `mongod` с фалами данных в старом формате. Примечание. В большинстве случаев нет необходимости использовать данную опцию
`--traceExceptions`	Используется только для внутренней диагностики

5.1.2. Опции репликации

Опции репликации `mongod` приведены в таблице 3.

Таблица 3

Параметр	Описание
`--replSet <setname>`	Используется для настройки репликации с набором реплики. Определяет имя набора. Все хосты должны иметь одинаковое имя набора
`--oplogSize <value>`	Определяет максимальный размер журнала операций репликации в мегабайтах. `mongod` создает журнал операций основываясь на максимальном доступном свободном месте на диске. В общем случае размер журнала операций равен 5% доступного места на диске. После первого создания `mongod` журнала операций использование опции `--oplogSize` не приводит к изменению размера журнала

Окончание таблицы 3

Параметр	Описание
`--fastsync`	В контексте репликации набора реплики опция используется для установки члена набора реплики со снимка каталога БД другого члена набора реплики. В противном случае `mongod` будет пытаться выполнять начальную синхронизацию для нового члена набора реплики. **ВНИМАНИЕ!** Если данные не синхронизированы полностью и `mongod` начинает работу с `fastsync`, то `secondary` или `slave` будут постоянно рассинхронизирован с `primary`, что может привести к возникновению серьезных проблем с согласованностью данных
`--replIndexPrefetch`	Используется в сочетании с `--replSet`. Значение по умолчанию – `all`. Доступны следующие значения: `none`, `all`, `_id_only`. По умолчанию вторичные члены набора реплики будут загружать все индексы, относящиеся к операции в памяти до применения операций из журнала операций. Существует возможность таким образом модифицировать поведение, чтобы вторичные члены загружали только индекс `_id`. Необходимо указать `_id_only` или `none` для предотвращения загрузки `mongod` любого индекса в память

5.1.3. Опции репликации Master-Slave

Опции репликации Master-Slave `mongod` приведены в таблице 4. Указанные опции предоставляют доступ к БД репликации `master-slave`. Наличие данных опций делает предпочтительной конфигурацию набора реплики для репликации БД.

Таблица 4

Параметр	Описание
`--master`	Конфигурирует `mongod` для работы в качестве `master` (мастера) репликации
`--slave`	Конфигурирует `mongod` для работы в качестве `slave` (подчиненного) репликации
`--source <host><:port>`	Используется с опцией `--slave` и определяет сервер, который будет реплицировать данный экземпляр ДОСУБД
`--only <arg>`	Используется с опцией `--slave` и определяет единственную БД для репликации
`--slavedelay <value>`	Используется с опцией `--slave` и определяет задержку в секундах для ожидания применения операций с узла `master` (мастера)

Окончание таблицы 4

Параметр	Описание
`--autoresync`	Используется с опцией `--slave` и позволяет `slave` (подчиненному) автоматически выполнять повторную синхронизацию в случае когда он отстает от мастера более чем на 10 секунд. Использование может быть проблематичным если опция `--oplogSiz` определяет слишком маленькое значение. Если размер журнала операций недостаточно велик для хранения разницы между текущим состоянием мастера и состоянием подчиненного, то экземпляр будет принудительно синхронизировать себя без необходимости. При установке значения `false` для опции `--autoresync`, подчиненный не будет пытаться автоматически синхронизировать себя более чем один раз в течении периода, равного 10 минутам

5.1.4. Опции кластера шардинга

Опции `mongod` для кластера шардинга приведены в таблице 5.

Таблица 5

Параметр	Описание
`--configsvr`	Определяет, что данный экземпляр `mongod` обслуживает конфигурационную БД кластера шардинга. При запуске с данной опцией, клиенты не смогут записать данные в БД, отличные от `config` и `admin`. По умолчанию порт для `mongod` с данной опцией `27019`, а каталог `--dbpath /data/configdb`, если не указано иное. Опция `--configsvr` также устанавливает опцию `--smallfiles` и созадет журнал операций. Опция `--configsvr` не должна использоваться совместно с `--replSet` или `--shardsvr`. Конфигурационный сервер не может быть узлом шардинга или частью набора реплики
`--shardsvr`	Конфигурирует `mongod` для работы в качестве узла кластера шардинга. По умолчанию порт для `mongod` с данной опцией `27018`
`--moveParanoia`	В ходе частичного переноса (миграции) данных опция `--moveParanoia` определяет, что `mongod` сохраняет все переносимые с данного узла шардинга документы в подкаталоге `moveChunk` в каталоге `dbpath`. ДОСУБД не удаляет данные из данного подкаталога

5.1.5. Опции SSL

Опции для использования SSL в `mongod` приведены в таблице 6.

Таблица 6

Параметр	Описание
`--sslOnNormalPorts`	Включает использование SSL в `mongod`. С опцией `--sslOnNormalPorts` `mongod` использует SSL-преобразование для всех соединений, устанавливаемых через порт ДОСУБД по умолчанию или порт, указанный опцией `--port`. По умолчанию опция `--sslOnNormalPorts` отключена
`--sslPEMKeyFile <filename>`	Определяет pem-файл, содержащий сертификат и ключ SSL. Имя файла указывается с использованием относительного или абсолютного пути. При использовании опции `--sslOnNormalPorts` необходимо указать pem-файл опцией `--sslPEMKeyFile`
`--sslPEMKeyPassword <value>`	Определяет пароль для открытия файла, заданного опцией `--sslPEMKeyFile` и содержащего пару «сертификат-ключ>. Опция `--sslPEMKeyPassword` используется только если пароль устанавливался при создании файла
`--sslCAFile <filename>`	Определяет pem-файл, содержащий корневой сертификат цепочки от центра сертификации. Имя файла указывается с использованием относительного или абсолютного пути
`--sslCRLFile <filename>`	Определяет pem-файл, содержащий список отозванных сертификатов. Имя файла указывается с использованием относительного или абсолютного пути
`--sslWeakCertificateValidation`	Отключает проверку сертификатов SSL, включаемую опцией `--sslCAFile`, и `mongod` будет принимать соединения, если клиент не представил сертификат при установке соединения. Если клиент представил сертификат при установке соединения `mongod` запущен с опцией `--sslWeakCertificateValidation`, то `mongod` будет проверять сертификат, используя корневой сертификат цепочки, заданный опцией `--sslCAFile`, и отклонять соединения клиентов с недействительными сертификатами. Опция `--sslWeakCertificateValidation` применяется если существуют клиенты, которые не имеют или не могут представить сертификат
`--sslFIPSMode`	Определяет использование режима FIPS (Federal Information Processing Standard) в установленной библиотеке OpenSSL

5.1.6. Использование mongod

В общем случае вызов `mongod` в контексте сценария инициализации или управления будет выглядеть следующим образом:

```
mongod --config /etc/mongodb.conf
```

5.2. Служба mongos

Процесс службы mongos является сервисом маршрутизации для кластера шардинга ДОСУБД, который обрабатывает запросы на уровне приложений, определяя положение необходимых данных в кластере шардинга. Как приложение mongos ведет себя аналогично другим экземплярам ДОСУБД.

5.2.1. Основные опции

Опции командной строки mongos приведены в таблице 7.

Таблица 7

Параметр	Описание
--help, -h	Показывает справку по вызову команды
--version	Показывает версию
--config <filename> -f <filename>	Указывает конфигурационный файл, который используется для управления функционированием. Несмотря на то, что опции конфигурационного файла эквиваленты доступным в командной строке, конфигурационный файл является предпочтительным методом управления функционированием ДОСУБД. **ВНИМАНИЕ!** Конфигурационный файл должен иметь кодировку ASCII. mongod не поддерживает не ASCII кодировку, включая UTF-8. Не все конфигурационные опции, используемые для mongod, имеют значение в контексте mongos
--verbose, -v	Увеличивает количество внутренней отчетности, возвращаемое в стандартном потоке вывода или в файле журнала генерируемом с именем logpath. Используется форма -v для управления количеством выводимой информации посредством включения опции несколько раз, например, -vvvvv
--quiet	Выполняет экземпляр mongos в режиме ограничения выводимой информации
--port <port>	Указывает TCP-порт для ожидания mongos клиентских соединений. По умолчанию mongos ожидает соединения с использованием порта 27017. В UNIX-подобных системах необходимы привилегии суперпользователя root для использования портов с номерами меньше 1024
--bind_ip <ip address>	IP-адрес, который использует процесс mongos для ожидания входящих соединений. По умолчанию mongos ожидает соединения на интерфейсе 127.0.0.1. Для привязки экземпляра определения mongod или mongos к множеству IP-адресов можно указать список их значений, разделенных запятой

Продолжение таблицы 7

Параметр	Описание
`--maxConns <number>`	Указывет максимальное число одновременных соединений, которые может принять `mongos`. Значение не будет действовать, если оно превышает установленное в ОС СН пороговое значение для максимального числа соединений. Опция полезна для `mongos` при наличие клиента, который создает коллекции, а их закрытие осуществляется по таймауту. **ВНИМАНИЕ!** Невозможно установить значение `maxConns`, превышающее `20000`
`--objcheck`	Принуждает `mongos` проверять все полученные запросы клиентов для предотвращения вставки клиентами в БД некорректных документов. Данная опция `--objcheck` влияет на производительность и не используется по умолчанию
`--logpath`	Определяет имя файла журнала, который будет содержать всю диагностическую информацию. Если значение параметра не определено, то протоколирование осуществляется в стандартный поток вывода. Если дополнительно не указана опция `--logappend`, то файл журнала будет перезаписываться при каждом перезапуске процесса
`--logappend`	Используется для добавления службой `mongos` новых записей в конец файла журнала вместо перезаписывания содержимого журнала при перезапуске процесса
`--setParameter <options>`	Указывает опцию для настройки при запуске. Может использоваться несколько вхождений опции `--setParameter`. Команда БД `setParameter` позволяет получить доступ ко многим из параметров. Опция `--setParameter` поддерживает следующие значения: – `enableLocalhostAuthBypass`; – `enableTestCommands`; – `logLevel`; – `logUserIds`; – `notablescan`; – `quiet`; – `supportCompatibilityFormPrivilegeDocuments`; – `syncdelay`; – `textSearchEnabled`
`--syslog`	Направляет все протоколирование в журнал `syslog` вместо стандартного потока вывода или файла журнала. **ВНИМАНИЕ!** Опция `--syslog` не может использоваться одновременно с `--logpath`
`--pidfilepath <path>`	Указывает положение файла, содержащего PID (идентификатор процесса) `mongos`. Опция полезна для отслеживания процесса `mongos` в сочетании с опцией `mongos --fork`. Если опция `--pidfilepath` не указана, то `mongos` не создает PID-файл

Продолжение таблицы 7

Параметр	Описание
`--keyFile`	Определяет файл ключей, в котором хранится аутентификационная информация. Параметр используется только для соединений между `mongos` и узлами кластера шардинга. **ВНИМАНИЕ!** Использование опции автоматически включает режим аутентификации для всех пользователей. При использовании данной опции необходимо наличие хотя бы одного зарегистрированного пользователя, имеющего доступ к администрированию
`--nounixsocket`	Отключает ожидание соединений с использованием сокетов UNIX. `mongos` ожидает входящие соединения с использованием сокетов UNIX, если не указана опция `--nounixsocket`, опция `--bind_ip` не задана или опцией `--bind_ip` задано значение `127.0.0.1`
`--unixSocketPrefix <path>`	Определяет путь к сокету UNIX. Если опция не имеет значения `mongos` создает сокет, используя префикс `/tmp`. ДОСУБД всегда создает сокет UNIX и осуществляет ожидание на нем входящих соединений если не установлена опция `--nounixsocket`
`--fork`	Разрешает `mongos` работать в режиме службы, переключая процесс в фоновый режим. Данный режим функционирования является обычным, за исключением тестирования
`--configdb <config1>, <config2><:port>,<config3>`	Опция задает конфигурационную БД для кластера шардинга. Необходимо указать один или три конфигурационных сервера в списке, разделенном запятыми. **ВНИМАНИЕ!** Экземпляр `mongos` выполняет чтение с первого конфигурационного сервера списка. Все экземпляры `mongos` должны для опции `--configdb` использовать одинаковый порядок узлов. Нельзя удалять конфигурационный сервер (серверы) из значения параметра `--configdb`, даже если конфигурационный сервер (серверы) недоступны по сети (выключены)
`--test`	Опция используется только для внутреннего тестирования и запускает набор тестов без запуска экземпляра `mongos`
`--upgrade`	При необходимости изменяет формат мета-данных, используемый в конфигурацинной БД
`--chunkSize <value>`	Значение определяет размер в мегабайтах каждой порции данных, передаваемых между узлами кластера шардинго. Значение по умолчанию 64 МБ, является идеальным размером порции данных для большинства конфигураций. Увеличение может привести к неравномерной передаче данных, уменьшение к неэффективной передаче данных между узлами. **ВНИМАНИЕ!** Опция используется только при первоначальной инициализации кластера

Окончание таблицы 7

Параметр	Описание
`--ipv6`	Включает поддержку IPv6. Позволяет клиентам подключаться к `mongod` с использованием сетей IPv6. По умолчанию в `mongod` поддержка IPv6 отключена в mongod и утилитах, указанных в 8
`--jsonp`	Разрешает JSONP-доступ через интерфейс HTTP
`--noscripting`	Отключает обработчика сценариев
`--nohttpinterface`	Отключает интерфейс HTTP
`--localThreshold`	Управляет алгоритмом, который `mongos` использует при выборе члена набора реплики для выполнения операций чтения клиентов. Значение указывается в миллисекундах. Значение по умолчанию, равное 15, соответствует значению по умолчанию во всех клиентских драйверах
`--noAutoSplit`	Опция `--noAutoSplit` предотвращает автоматическую вставку метаданных с распределением по узлам шардинга

5.2.2. Опции SSL

Использование опций SSL в `mongos` аналогично `mongod` (см. 5.1.5).

5.2.3. Использование mongod

В общем случае вызов `mongos` в контексте сценария инициализации или управления будет выглядеть следующим образом:

```
mongos --config /etc/mongodb.conf
```

6. РЕПЛИКАЦИЯ

Репликация обеспечивает избыточность, резервное копирование и восстановление после сбоев и осуществляется в группах серверов, известных как наборы реплики.

6.1. Концепция репликации ДОСУБД

В ДОСУБД набором реплики является кластер экземпляров `mongod`, выполняющих репликацию между собой и обеспечивающих автоматическое восстановление после сбоев. Как правило, набор реплики состоит из двух или более экземпляров `mongod`, из которых как минимум один является первичным, а остальные вторичными членами. Клиенты выполняют запись непосредственно на первичный член группы, в то время как вторичные выполняют с него асинхронную репликацию. Набор реплики ДОСУБД обеспечивает автоматическое восстановление после сбоев: если возникает сбой первичного члена, то оставшиеся автоматически производят выбор нового первичного из числа вторичных. Модель репликации приведена на рис. 1.

Все операции выполняются на первичном члене набора реплики и регистрируются в журнале операций репликации, расположенном в системной коллекции локальной БД `local.oplog.rs` Вторичные члены периодически обращаются к журналу операций первичного и определяют по метке времени список операций для исполнения.

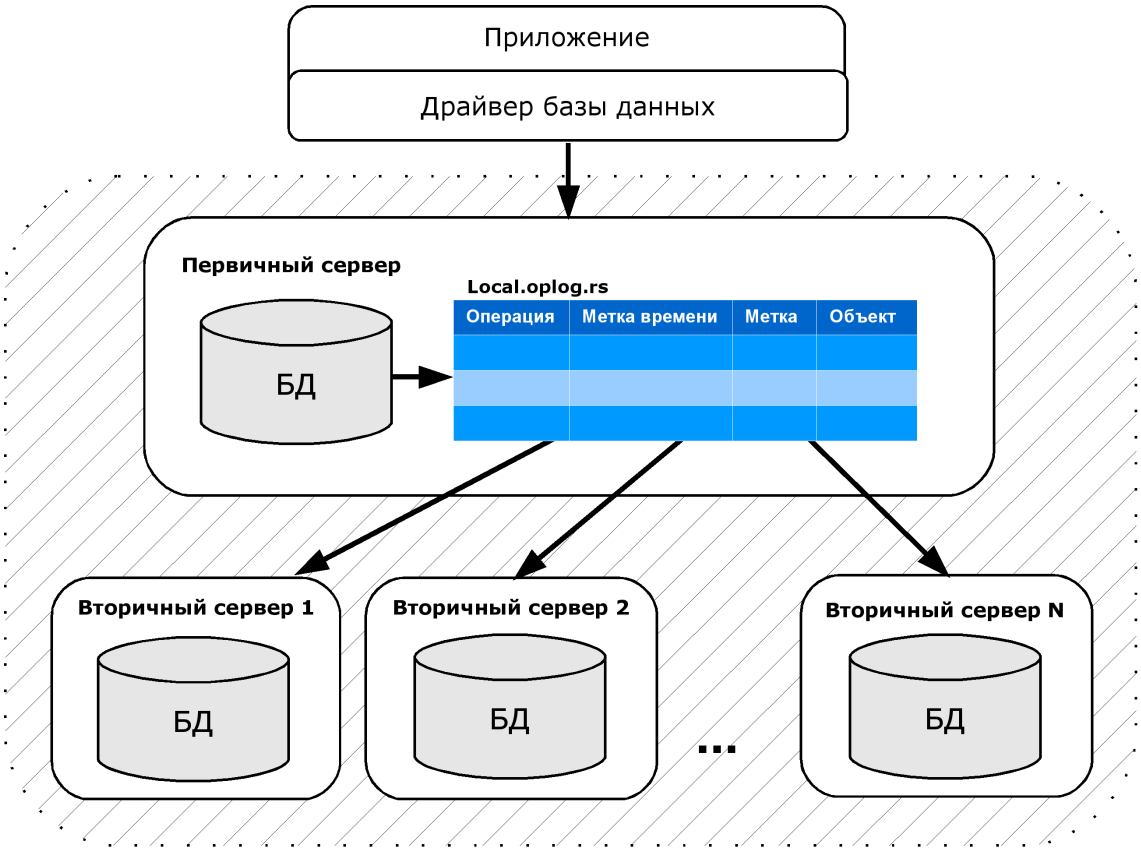

Рис. 1 – Модель репликации

Примечание. ДОСУБД поддерживает и традиционную модель репликации ведущий/ведомый (master/slave), которая функционирует схожим образом с набором реплики, но не обеспечивает автоматического восстановления после сбоев. Набор реплики является рекомендуемым решением, но может содержать не более 12 членов, из которых только 7 имеют право голоса. Если решение требует наличия более чем 11 вторичных членов, то следует выбирать модель ведущий/ведомый.

6.2. Настройка набора реплики

Набор реплики может содержать от 2 до 12 серверов ДОСУБД (экземпляров mongod). Каждый набор реплики имеет свое имя, которое указывается для всех членов набора, например rs0. Для указания имени набора реплики используется опция командной строки --replSet или одноименный параметр конфигурационого файла mongo.conf (см. 4.10.

Примечание. В целях тестирования возможно создание набора реплики на одном сервере запуском нескольких экземпляров службы mongod. При этом для каждого экземпляра должен быть задан и существовать свой путь к месту хранения баз данных и указан свой порт для принятия входящих соединений.

Рассмотрим процесс создания набора реплики на примере трех экземпляров mongod, запущенных на одном сервере.

Для работы с набором реплики в командной оболочке ДОСУБД mongo предусмотрен объект rs, методами которого осуществляется конфигурирование, управление и просмотр состояния набора реплики.

Пример

1) Перед созданием набора реплики необходимо обеспечить надежное сетевое соединение между всеми членами набора. Конфигурация сети должна позволять все возможные соединения между любыми двумя членами набора.

2) Запускаются три сервера, при этом должны быть выполнены следующие условия:

а) для каждого сервера должны быть созданы необходимые пути к месту хранения баз данных и журналов, например:

```
mkdir -p /srv/mongodb/rs0-0  /srv/mongodb/rs0-1  /srv/mongodb/rs0-2
```

б) запускаются службы mongod, например:

```
mongod --port 27017 --dbpath /srv/mongodb/rs0-0 --replSet rs0
mongod --port 27018 --dbpath /srv/mongodb/rs0-1 --replSet rs0
mongod --port 27019 --dbpath /srv/mongodb/rs0-2 --replSet rs0
```

3) Запускается командная оболочка ДОСУБД mongo с установкой соединения к первому экземпляру mongod:

```
mongo --port 27017
```

4) С помощью командной оболочки ДОСУБД `mongo` создается следующий объект для инициализации набора реплики:

```
rsconf = {
  _id: "rs0",
  members: [
    {
      _id: 0,
      host: "<hostname>:27017"
    }
   ]
}
```

5) Используя команду `rs.initiate()`, инициализируется набор реплики с одним текущим членом и конфигурацией по умолчанию:

```
rs.initiate(rsconf)
```

6) Просматривается текущая конфигурация набора реплики:

```
rs.conf()
```

7) В конфигурацию набора реплики добавляются второй и третий сервер с помощью команды `rs.add()`:

```
rs.add("<hostname>:27018")
rs.add("<hostname>:27019")
```

8) В любой момент времени может быть получен статус набора реплики:

```
rs.status()
```

ВНИМАНИЕ! Для организации аутентификации между серверами набора реплики используется файл ключей, в котором хранится аутентификационная информация. Файл ключей задается параметром `keyFile` (см. 4.1.)

7. ШАРДИНГ

Одной из технологий обеспечения масштабируемости для повышения производительности доступа к данным и параллельных вычислений над очень большими наборами данных в компьютерных системах является шардинг, который подразумевает распределение единой базы данных между группой серверов, составляющих один кластер. При этом используется горизонтальное разбиение наборов документов на основании заданного ключа (условия) разбиения.

7.1. Концепция шардинга ДОСУБД

ДОСУБД поддерживает технологию шардинга на уровне баз данных и коллекций документов. Модель шардинга представлена на рис. 2.

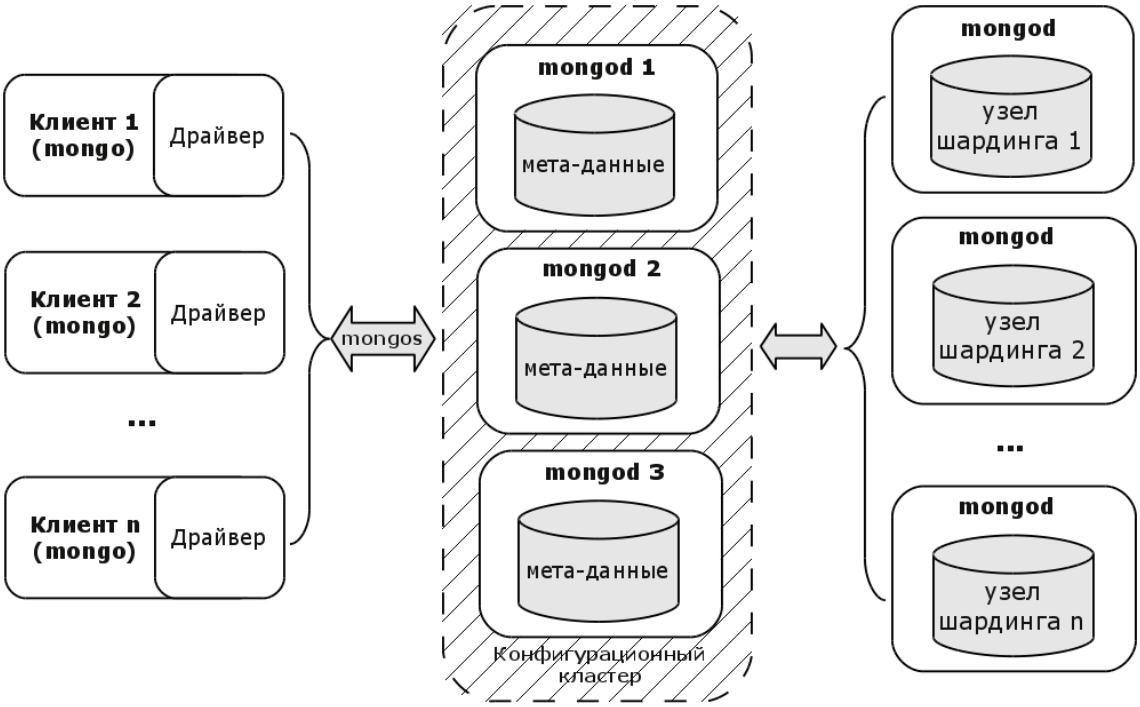

Рис. 2 – Шардинг

После создания кластера для каждой базы данных задается режим разделения. В разделяемой базе данных для каждой коллекции, которая требует разделения, определяется ключ разделения, разбивающий все документы коллекции по диапазонам своего значения. Каждый сервер кластера, являющийся узлом шардинга (шард — от английского Shard — осколок), обрабатывает документы, попадающие в определенный диапазон значений ключа. Ключ может определяться как одним, так и несколькими полями документов.

Ключ разбиения сильно влияет на производительность, возможности и функционирование БД и кластера в целом, что определяет важность его правильного выбора. Идеальный ключ разбиения должен обладать следующими свойствами:

– легкость разбиения диапазонов своих значений для облегчения распределения

данных между шардами кластера, поскольку ключ содержащий недостаточное количество возможных значений может не позволить произвести необходимое число разбиений;

– значения ключа должны по возможности распределяться равномерно в зависимости от свойств документов для обеспечения равномерного распределения операций записи при добавлении документов по шардам, в противном случае какой-нибудь из них может стать «узким местом»;

– являться первичным условием запрашиваемых данных для минимизации используемых для сбора результата шардов, т.к. использование в качестве условия полей с высокой степенью случайности своих значений может сильно снизить производительность за счет опроса большого количества серверов кластера.

Зачастую в существующих свойствах документов коллекции может не оказаться подходящего поля. В этом случае рекомендуется использовать либо составной ключ по нескольким полям, либо специальным образом созданный суррогатный ключ.

В пределах шарда коллекция документов также разбивается на части (Chunk), каждая из которых представляет небольшой поддиапазон значений из диапазона значений ключа конкретного шарда. Когда часть коллекции документов начинает превышать по объему заданное граничное значение, она разбивается на меньшие части также на основании ключа шардинга.

Механизм шардинга обеспечивает автоматическое распределение данных и нагрузки между серверами кластера (шардами). В случае завершения места на шарде, в состав кластера достаточно добавить еще один сервер, после чего данные в нем перераспределяться автоматически.

В случае использования шардинга взаимодействие клиента с ДОСУБД осуществляется через специальную службу шардинга (экземпляр сервера `mongos`), который обеспечивает перенаправление запроса на соответствующий сервер шарда. Аутентификация и разграничения доступа к базам данных и коллекциям осуществляется на сервере шардинга.

7.2. Настройка шардинга

Шардинг реализуется в пределах кластера шардинга, который состоит из следующих компонентов:

1) Шарды. Каждый шард представляет собой или отдельный экземпляр сервера ДОСУБД `mongod` или набор реплики, содержащие часть коллекций базы данных.

2) Сервера конфигурации. Каждый сервер конфигурации представляет собой отдельный экземпляр сервера ДОСУБД `mongod` с конфигурационной базой данных, в

которой содержится информация о метаданных кластера. Функционирование кластера шарда обеспечивается тремя серверами конфигурации, при этом они организованы не как набор реплики, а взаимодействуют с помощью двухфазных транзакций, обеспечивающих согласованность и целостность метаданных.

Примечание. В целях тестирования возможно использование одного сервера конфигурации. При реальном использовании это категорически не рекомендуется.

3) Сервер шардинга. Сервер шардинга представляет отдельный экземпляр службы шардинга mongos, который обеспечивает перенаправление запросов на соответствующий сервер шарда.

Рассмотрим процесс создания шардинга с помощью служб, запущенных на одном сервере. При этом для каждого экземпляра сервера ДОСУБД mongod должен быть задан и существовать свой путь к месту хранения баз данных и указан свой порт для принятия входящих соединений.

Для работы с кластером шардинга в командной оболочке ДОСУБД mongo предусмотрен объект sh, методами которого осуществляется конфигурирование, управление и просмотр состояния кластера шардинга.

Пример

1) Подготовительные действия:

а) перед созданием необходимо обеспечить надежное сетевое соединение между всеми серверами кластера шардинга;

б) для каждого сервера должны быть созданы необходимые пути к месту хранения баз данных и журналов, например:

```
mkdir -p /srv/mongodb/sh-cfg  /srv/mongodb/sh-1  /srv/mongodb/sh-2
```

2) Создание кластера шардинга:

а) запускается служба mongod конфигурационного сервера (для конфигурационного сервера используется порт по умолчанию 27019):

```
mongod --port 27019 --dbpath /srv/mongodb/sh-cfg --configsvr
```

б) запускается служба шардинга mongos, при этом должны быть указаны адреса всех конфигурационных серверов:

```
mongos --configdb <адреса конфигурационных серверов>
```

в рассматриваемом примере:

```
mongos --configdb <hostname>:27019
```

3) Добавление шарда к кластеру:

а) запускается служба mongod сервера конкретного шарда, например:

```
mongod --port 27021 --dbpath /srv/mongodb/sh-1
mongod --port 27022 --dbpath /srv/mongodb/sh-2
```

б) запускается командная оболочка ДОСУБД mongo с установкой соединения к

служба шардинга mongos:

```
mongo --port 27017
```

в) с помощью команды sh.addShard() к кластеру добавляются шарды:

```
sh.addShard( "<hostname>:27021" )
sh.addShard( "<hostname>:27022" )
```

4) Включение разделения базы данных:

а) запускается командная оболочка ДОСУБД mongo с установкой соединения к служба шардинга mongos:

```
mongo --port 27017
```

б) с помощью команды sh.enableSharding() выполняется включение разделения указанной базы данных:

```
sh.enableSharding( "<база данных>" )
```

5) Включение разделения коллекции:

а) запускается командная оболочка ДОСУБД mongo с установкой соединения к служба шардинга mongos:

```
mongo --port 27017
```

б) с помощью команды sh.shardCollection() выполняется включение разделения указанной коллекции:

```
sh.shardCollection( "<база данных>.<коллекция>", ключ_разбиения )
```

например:

```
sh.shardCollection( "records.people", { "zipcode": 1, "name": 1 } )
```

в данном случае коллекция people базы данных records разделяется с использованием значения поля zipcode. Для документов, имеющих одинаковое значение названного поля, применяется разделение с использованием значения поля name.

ВНИМАНИЕ! Для эффективной работы механизма шардинга ключ разбиения должен обладать свойствами, описанными в 7.1

ВНИМАНИЕ! Для организации аутентификации между серверами используется файл ключей, в котором хранится аутентификационная информация. Файл ключей задается параметром keyFile (см. 4.1).

8. УТИЛИТЫ КОМАНДНОЙ СТРОКИ

Для начала работы пользователя с ДОСУБД необходимо наличие установленного и настроенного сервера ДОСУБД.

Работа с ДОСУБД требует установки соединения с сервером ДОСУБД, что при использовании клиентских утилит командной строки обеспечивается заданием свойств соединения с помощью аргументов (опций) командной строки, приведенных в таблице 8.

Таблица 8

Опция	Описание
`--host <hostname><:port>`	Указывает имя сервера ДОСУБД. По умолчанию осуществляется попытка подключения к серверу ДОСУБД, функционирующему на локальном хосте с номером порта `27017`. Дополнительно может указываться порт сервера ДОСУБД для подключения к экземпляру ДОСУБД, функционирующему с номером порта отличным от `27017`. Для подключения к набору реплики можно указать имя набора реплики и перечень членов реплики в следующем формате: `<replica_set_name>/<hostname1><:port>, <hostname2:<port>,...`
`--port <port>`	Указывает порт сервера ДОСУБД для подключения к экземпляру ДОСУБД, функционирующему с номером порта отличным от `27017`. Номер порта может быть также указан с использованием опции `--host`
`--username <username>, -u <username>`	Указывает имя пользователя для аутентификации при подключении к серверу ДОСУБД. Опция используется в сочетании с опцией `--password`. Если опция `--username` использована без опции `--password`, то будет выдано приглашение для ввода пароля
`--password <password>, -p <password>`	Указывает пароль для аутентификации подключении к серверу ДОСУБД. Опция используется в сочетании с опцией `--username`

Информацию о версии и способе вызова утилит и допустимых аргументов можно получить с помощью аргументов:

`--help` — показать справку по вызову команды;

`--version` — показать версию.

8.1. Командная оболочка ДОСУБД

Основным средством пользователя для взаимодействия с ДОСУБД является командная оболочка ДОСУБД `mongo`, которая представляет собой интерактивный командный интерфейс доступа к ДОСУБД, реализованный на JavaScript. Подробная информация по работе с командной оболочкой приведена на страницах справочного руководства `man`.

8.1.1. Опции

Командная оболочка ДОСУБД имеет опции, приведенные в таблице 9.

Таблица 9

Опция	Описание
`--shell`	Предоставляет командный интерфейс после выполнения JavaScript-файла. Если командный интерфейс вызван с указанием JavaScript-файла в качестве аргумента, или использована опция `--eval` для передачи JavaScript командной строки, то опция `--shell` выводит приглашение для ввода команд после завершения выполнения файла
`--nodb`	Предотвращает подключение командной оболочки к какому-либо экземпляру БД. Позднее подключение к БД может быть выполнено внутри командной оболочки
`--norc`	Предотвращает открытие и выполнение командной оболочкой файла `~/.mongorc.js` при запуске
`--quiet`	Отключает вывод из командной оболочки в процессе подключения
`--port <port>`	Указывает порт, на котором экземпляр `mongod` или `mongos` осуществляет прослушивание, в случае если номер порта отличается от `27017`, который является номером, используемым по умолчанию
`--host <hostname>`	Указывает хост, на котором функционирует экземпляр `mongod` или `mongos` и к которому осуществляется подключение как к `<hostname>`. По умолчанию `mongo` осуществляет попытку подключения к процессу ДОСУБД, выполняющемуся на локальном хосте
`--eval <javascript>`	Выполняет JavaScript-выражение, указанное в качестве аргумента для данной опции. Оболочка командной строки не выполняет загрузку собственного окружения при выполнении кода. В результате многие опции окружения оболочки недоступны
`--username <username>`, `-u <username>`	Указывает имя пользователя для аутентификации в ДОСУБД. Используется в сочетании с опцией `--password`, задающей пароль. Если указаны имя пользователя и пароль, но БД по умолчанию или указанная БД не требуют аутентификации, то `mongo` завершится с исключением
`--password <password>`, `-p <password>`	Указывает пароль для аутентификации в экземпляре ДОСУБД. Используется в сочетании с опцией `--username`, задающей имя пользователя. Если опция `--username` использована без опции `--password`, `mongo` выведет приглашение для интерактивного ввода пароля, в случае если `mongod` или `mongos` требуют аутентификации

Продолжение таблицы 9

Опция	Описание
`--authenticationDatabase <dbname>`	Указывает БД, в которой содержится аутентификационная информация и полномочия пользователей. По умолчанию `mongo` полагает, что БД, указанная в качестве адреса БД, содержит аутентификационную информацию и полномочия пользователей, если не использована опция `--authenticationDatabase`
`--authenticationMechanism <name>`	Указывает аутентификационный механизм. По умолчанию аутентификационным механизмом является MONGODB-CR, который является механизмом типа «запрос/ответ». В ДОСУБД реализована поддержка GSSAPI для поддержки Kerberos-аутентификации
`--ssl`	Предоставляет возможность подключения к `mongod` или `mongos` с использованием SSL
`--sslPEMKeyFile <filename>`	Указывает pem-файл, содержащий сертификат и ключ SSL. Имя pem-файла задается с использованием относительного или абсолютного пути. Опция требуется при использования опции `--ssl`, если `mongod` или `mongos` имеют `sslCAFile`, доступный без `sslWeakCertificateValidation`
`--sslPEMKeyPassword <value>`	Указывает пароль для декодирования корневого сертификата цепочки, заданного `--sslPEMKeyFile`. Требуется только в случае, когда файл, содержащий пару «сертификат-ключ», закодирован
`--sslCAFile <filename>`	Указывает pem-файл, который содержит сертификат из центра сертификации. Имя pem-файла задается с использованием относительного или абсолютного пути
`--help, -h`	Выводит текст базовой справки о командной оболочке ДОСУБД
`--version`	Выводит информацию о версии командной оболочки ДОСУБД
`--verbose`	Увеличивает количество информации, выводимой в командной оболочке ДОСУБД в процессе подключения
`--ipv6`	Включает поддержку IPv6, которая позволяет `mongo` подключаться к экземпляру ДОСУБД, используя сети IPv6. Во всех программах и процессах ДОСУБД, включая `mongo`, поддержка IPv6 по умолчанию отключена

Окончание таблицы 9

Опция	Описание

<db address>	Указывает адрес БД для подключения. Например:
	`mongo admin`
	
	Данная команда будет осуществлять подключение командной оболочки `mongo` к БД admin на локальной машине. Можно указать удаленный экземпляр БД, используя разрешаемое имя хоста или IP-адрес. Имя БД отделяется от имени хоста с использованием символа `/`. Примеры команд для подключения:
	`mongo mongodb1.example.net`
	`mongo mongodb1/admin`
	`mongo 10.8.8.10/test`

<file.js>	Указывает JavaScript-файл для выполнения с последующим выходом. Данная опция должна быть последней в списке. Для возвращения в командный режим после завершения выполнения файла необходимо использовать опцию `--shell`

8.1.2. Файлы

Командная оболочка ДОСУБД использует файлы, приведенные в таблице 10.

Таблица 10

Файл	Описание
~/.dbshell	Содержит историю выполнения команд. Следует отметить, что `mongo` не записывает в файл истории действия, относящиеся к аутентификации, включая `authenticate` и `db.addUser()`
~/.mongorc.js	`mongo` читает файл `.mongorc.js` из домашней директории пользователя вызвавшего `mongo`. В файле пользователь может определить переменные, настроить приглашение командой оболочки или выполнить обновление информации, используемой при каждом запуске оболочки. Если оболочка используется для выполнения JavaScript-файла или выражения, заданного опцией `--eval`, то `mongo` будет выполнять чтение файла `.mongorc.js` после завершения выполнения JavaScript. Для отключения чтения файла `.mongorc.js` используется опция `--norc`
/tmp/mongo_edit<time_t>.js	Создается `mongo` при редактировании файла. Если файл существует, то `mongo` будет добавлять целое число от 1 до 10 к значению времени для выполнения попытки создания уникального файла

8.1.3. Окружение

Командная оболочка ДОСУБД использует переменные окружения, приведенные в таблице 11.

Таблица 11

Переменная	Описание
`EDITOR`	Определяет путь к редактору, вызываемому из оболочки. JavaScript-переменная `EDITOR` переопределяет значение переменной `EDITOR`
`HOME`	Определяет путь к домашнему каталогу, из которого `mongo` будет выполнять чтение файла `.mongorc.js` и запись файла `.dbshell`

8.1.4. Горячие клавиши

Командная оболочка `mongo` поддерживает «горячие клавиши», приведенные в таблице 12.

Таблица 12

Клавиша	Функция
Стрелка вверх	Выбор предыдущей команды из истории
Стрелка вниз	Выбор следующей команды из истории
Home	Переход в начало строки
End	Переход в конец строки
Tab	Команда автозавершения
Стрелка влево	Переход назад на один символ
Стрелка вправо	Переход вперед на один символ
Ctrl и стрелка влево	Переход назад на одно слово
Ctrl и стрелка вправо	Переход вперед на одно слово
Клавиша Windows и стрелка влево	Переход назад на одно слово
Клавиша Windows и стрелка вправо	Переход вперед на одно слово
Ctrl-A	Переход в начало строки
Ctrl-B	Переход в конец строки
Ctrl-C	Выход из командной оболочки
Ctrl-D	Удалить символ (или Выход из командной оболочки)
Ctrl-E	Переход в конец строки
Ctrl-F	Переход вперед на один символ
Ctrl-G	Прервать
Ctrl-J	Обработка/выполнение командой строки
Ctrl-K	Очистка/стирание командой строки
Ctrl-L или соманда cls	Очистка экрана

Окончание таблицы 12

Клавиша	Функция
Ctrl-M	Обработка/выполнение командой строки
Ctrl-N	Выбор следующей команды из истории
Ctrl-P	Выбор предыдущей команды из истории
Ctrl-R	Реверсивный поиск в истории команд
Ctrl-S	Прямой поиск в истории команд
Ctrl-T	Перестановка символов
Ctrl-U	Стирание строки
Ctrl-W	Стирание слова
Ctrl-Y	Вставка
Ctrl-Z	Приостановка (управляет задачами)
Ctrl-H	Обратное удаление символа
Ctrl-I	Аналогично Tab
Клавиша Windows-B	Переход назад на одно слово
Клавиша Windows-C	Писать слово прописными буквами
Клавиша Windows-D	Удалить слово
Клавиша Windows-F	Переход вперед на одно слово
Клавиша Windows-L	Изменить регистр слова на нижний
Клавиша Windows-U	Изменить регистр слова на верхний
Клавиша Windows-Y	Вставка
Клавиша Windows-Backspace	Обратное стирание слова
Клавиша Windows-<	Выбор первой команды из истории
Клавиша Windows->	Выбор последней команды из истории

8.1.5. Подключение к БД

Подключение к БД на удаленном хосте с использованием механизма аутентификации MONGODB-CR и номера порта не по умолчанию может быть выполнено следующей командой:

```
mongo --username <user> --password <pass> --hostname <host> --port 28015
```

или при помощи ее сокращенной формы:

```
mongo -u <user> -p <pass> --host <host> --port 28015
```

Необходимо заменить `<user>`, `<pass>` и `<host>` соответствующими значениями для имени пользователя, пароля и хоста, и указать номер порта или не использовать опцию `--port`.

8.1.6. Выполнение файла JavaScript

Для выполнения JavaScript-файла без выполнения файла `~/.mongorc.js` перед запуском сессии оболочки можно использовать с помощью следующей команды:

```
mongo --shell --norc alternate-environment.js
```

Вывод результатов запроса `JSON` из командной строки с использованием опции `--eval` можно выполнить с помощью следующей команды:

```
mongo --eval 'db.collection.find().forEach(printjson)'
```

Необходимо заключить JavaScript в одинарные кавычки `'`, а также дополнительный JavaScript, который требуется для генерации выходных данных.

8.2. Средство импорта mongoimport

Утилита `mongoimport` предоставляет возможность импорта содержимого файлов форматов JSON, CSV или TSV, созданных утилитой экспорта `mongoexport` или иной утилитой.

ВНИМАНИЕ! Не следует использовать утилиты `mongoimport` и `mongoexport` для работы с полной копией БД. Данные утилиты не обеспечивают надежную обработку типов данных. Необходимо использовать утилиты `mongodump` и `mongorestore`.

8.2.1. Опции mongoimport

Опции утилиты `mongoimport` представлены в таблице 13.

Таблица 13

Опция	Описание
`--help, -h`	Выводит текст базовой справки об утилите
`--version`	Выводит информацию о версии утилиты
`--verbose, -v`	Увеличивает количество информации, возвращаемое утилитой. Можно увеличить детализацию вывода опцией `-v`, включаемой несколько раз, например, `-vvvvv`

Продолжение таблицы 13

Опция	Описание

`--host <hostname><:port>, -h`	Указывает разрешаемое имя хоста, на котором функционирует экземпляр `mongod`, в который необходимо выполнить восстановление данных. По умолчанию `mongoimport` осуществляет попытку подключения к процессу ДОСУБД, выполняющемуся на локальном хосте с номером порта `27017`. Дополнительно можно указать номер порта для подключения к экземпляру ДОСУБД, использующему номер порта, отличный от `27017`. Для подключения к набору реплики опция `--host` используется с именем набора, за которым следует символ `/` и список имен хостов с номерами портов, разделенных запятыми. Утилита `mongoimport` будет выполнять подключение к первому включенному члену набора реплики. Например:
	`--host repl0/mongo0.example.net,`
	`mongo0.example.net:27018,`
	`mongo1.example.net,`
	`mongo2.example.net`
	
	Возможно подключение непосредственно к экземпляру ДОСУБД посредством указания конкретных имени хоста и номера порта

`--port <port>`	Указывает порт, используемый экземпляром ДОСУБД в случае, если номер порта отличается от `27017`. Возможно также указать номер порта в опции `--host`
`--ipv6`	Включает поддержку IPv6, которая позволяет `mongoimport` подключаться к экземпляру ДОСУБД, используя сети IPv6. Во всех программах и процессах ДОСУБД, включая `mongoimport`, поддержка IPv6 по умолчанию отключена
`--ssl`	Предоставляет возможность подключения к экземплярам `mongod` с использованием SSL
`--username <username>, -u <username>`	Указывает имя пользователя для аутентификации в ДОСУБД, если БД требует аутентификации. Используется в сочетании с опцией `--password`, задающей пароль
`--password <password>, -p <password>`	Указывает пароль для аутентификации в экземпляре ДОСУБД. Используется в сочетании с опцией `--username`, задающей имя пользователя. Если опция `--username` использована без опции `--password`, то `mongoimport` выведет приглашение для интерактивного ввода пароля
`--authenticationDatabase <dbname>`	Указывает БД, в которой содержится аутентификационная информация и полномочия пользователей. По умолчанию `mongoimport` полагает, что БД, указанная в качестве адреса БД, содержит аутентификационную информацию и полномочия пользователей, если не использована опция `--authenticationDatabase`

Продолжение таблицы 13

Опция	Описание
--authenticationMechanism <name>	Указывает аутентификационный механизм. По умолчанию аутентификационным механизмом является MONGODB-CR, который является механизмом типа «запрос/ответ». В ДОСУБД реализована поддержка GSSAPI для поддержки Kerberos-аутентификации
--dbpath <path>	Указывает путь к каталогу с файлами ДОСУБД. При использовании опция --dbpath позволяет mongoimport вставлять данные непосредственно в локальные файлы данных, не используя mongod. Для использования опции --dbpath mongoimport должна заблокировать доступ к каталогу данных. В результате ни один экземпляр mongod не может получить доступ к каталогу с тем же путем
--directoryperdb	Используется в сочетании с соответствующей опцией mongod, которая позволяет mongoimport импортировать данные в экземпляр ДОСУБД, который хранит каждый файл данных в каталоге на диске. Данная опция имеет смысл только при указании опции --dbpath
--journal	Позволяет использовать записанный утилитой mongoexport специальный журнал для обеспечения согласованности файлов данных в процессе записи. Данная опция имеет смысл только при указании опции --dbpath
--db <db>, -d <db>	Указывает БД для импорта данных
--collection <collection>, -c <collection>	Указывает коллекцию для импорта
--fields <field1<,field2>>, -f <field1[,field2]>	Задает список разделенных запятыми имен полей при импортировании файлов формата CSV или TSV, в которых в первой строке (пример заголовка) не указаны имена полей
--fieldFile <filename>	Является альтернативой для опции --fields и позволяет указать файл, который содержит список имен полей, в случае когда файл CSV или TSV, в которых не содержит в первой строке (пример заголовка) имена полей
--ignoreBlanks	Игнорирует пустые поля в файлах формата CSV или TSV. Если не указано поле, то mongoimport создает поля без значений в импортированных документах
--type <json|csv|tsv>|	Указывает импортируемый формат. По умолчанию формат JSON, но возможно импортировать файлы в формате CSV и TSV
--file <filename>	Указывает путь к файлу, содержащему импортируемые данные. Если файл не указан, то mongoimport будет читать данные из стандартного потока ввода
--drop	Модифицирует процедуру импортирования таким образом, чтобы целевой экземпляр ДОСУБД удалял каждую коллекцию перед ее восстановлением из резервной копии

Окончание таблицы 13

Опция	Описание
`--headerline`	В сочетании с `--type csv` или `--type tsv` использует первую строку для извлечения имен файлов. В противном случае, `mongoimport` будет импортировать первую строку как уникальный документ
`--upsert`	Модифицирует процедуру импортирования таким образом, чтобы выполнялось обновление существующих объектов в БД, если они совпадают с импортируемым объектом, и вставка всех прочих объектов. Если не указано поле или поля с использованием опции `--upsertFields`, то утилита `mongoimport` будет использовать опцию `--upsert` на основе поля `_id`
`--upsertFields <field1[,field2]>`	Указывает список полей для частичного использования опции `--upsert`. Опция используется, если поля `_id` в существующем документе не совпадают с полем в документе, но другое поле или комбинация полей могут уникально идентифицировать документы при использовании опции `--upsert`. Для обеспечения приемлемой производительности должны существовать индексы для данного поля или комбинации полей
`--stopOnError`	Принуждает `mongoimport` остановить операцию импорта при возникновении первой ошибки вместо продолжения операции с игнорированием ошибок
`--jsonArray`	Прием импортированных данных, выраженных множеством документов в одном JSON-массиве. Используется в сочетании с опцией `--jsonArray` утилиты `mongoexport` для импортирования данных, записанных как один JSON-массив. Размер импортируемых данных ограничен 16 MB

8.2.2. Использование mongoimport

Примеры:

1. В данном примере осуществляется импорт данных утилитой `mongoimport` в формате CSV из файла `/opt/backups/contacts.csv` в коллекцию «contacts» в БД пользователей в экземпляре ДОСУБД, выполняющемся на локальном хосте с номером порта `27017`.

```
mongoimport --db users --collection contacts --type csv
    --file /opt/backups/contacts.csv
```

2. В данном примере осуществляется импорт данных утилитой `mongoimport` в формате JSON из файла `contacts.json` в коллекцию «contacts» в экземпляре ДОСУБД, выполняющемся на локальном хосте с номером порта `27017`. Ведение журнала явно включено.

```
mongoimport --collection contacts --file contacts.json --journal
```

3. В данном примере mongoimport берет данные, переданные на стандартный поток ввода (например, через канал «|»), и импортирует данные в коллекцию «contacts» в БД «sales», в которой файлы данных размещены в каталоге /srv/mongodb/. Если при выполнении импорта возникает ошибка, то mongoimport останавливает работу использованием опции --stopOnError.

```
mongoimport --db sales --collection contacts --stopOnError
    --dbpath /srv/mongodb/
```

4. В данном примере mongoimport импортирует данные из файла /opt/backups/mdb1-examplenet.json в коллекцию «contacts» в удаленную БД «marketing». Mongoimport подключается к экземпляру mongod, выполняющемуся на хосте mongodb1.example.net с номером порта 37017, который запрашивает имя пользователя user с паролем pass.

```
mongoimport --host mongodb1.example.net --port 37017 --username user
    --password pass --collection contacts --db marketing
    --file /opt/backups/mdb1-examplenet.json
```

8.3. Средство экспорта mongoexport

Утилита mongoexport предоставляет возможность экспорта содержимого БД в файлы форматов JSON, CSV или TSV для последующего импорта утилитой mongoimport или другой утилитой.

ВНИМАНИЕ! Не следует использовать утилиты mongoimport и mongoexport для работы с полной копией БД. Данные утилиты не обеспечивают надежную обработку типов данных. Необходимо использовать утилиты mongodump и mongorestore.

8.3.1. Опции mongoexport

Опции утилиты mongoexport представлены в таблице 14.

Таблица 14

Опция	Описание
--help, -h	Выводит текст базовой справки об утилите
--version	Выводит информацию о версии утилиты
--verbose, -v	Увеличивает количество информации, возвращаемое утилитой. Можно увеличить детализацию вывода опцией -v, включаемой несколько раз, например, -vvvvv

Продолжение таблицы 14

Опция	Описание

`--host <hostname><:port>, -h`	Указывает разрешаемое имя хоста, с которого будут экспортироваться данные. По умолчанию `mongoexport` осуществляет попытку подключения к процессу ДОСУБД, выполняющемуся на локальном хосте с номером порта `27017`. Дополнительно можно указать номер порта для подключения к экземпляру ДОСУБД, использующему номер порта, отличный от `27017`. Для подключения к набору реплики опция `--host` используется с именем набора, за которым следует символ `/` и список разделенных запятой имен хостов с номерами портов. Утилита `mongoexport` будет выполнять подключение к первом включенному члену набора реплики. Например:
	`--host repl0/mongo0.example.net,`
	`mongo0.example.net:27018,`
	`mongo1.example.net,`
	`mongo2.example.net`
	
	Возможно подключение непосредственно к экземпляру ДОСУБД посредством указания конкретных имени хоста и номера порта

`--port <port>`	Указывает порт, используемый экземпляром ДОСУБД в случае, если номер порта отличается от `27017`. Возможно также указать номер порта в опции `--host`
`--ipv6`	Включает поддержку IPv6, которая позволяет `mongoimport` подключаться к экземпляру ДОСУБД, используя сети IPv6. Во всех программах и процессах ДОСУБД, включая `mongoimport`, поддержка IPv6 по умолчанию отключена
`--ssl`	Предоставляет возможность подключения к экземплярам `mongod` с использованием SSL
`--username <username>, -u <username>`	Указывает имя пользователя для аутентификации в ДОСУБД, если БД требует аутентификации. Используется в сочетании с опцией `--password`, задающей пароль
`--password <password>, -p <password>`	Указывает пароль для аутентификации в экземпляре ДОСУБД. Используется в сочетании с опцией `--username`, задающей имя пользователя. Если опция `--username` использована без опции `--password`, то `mongoimport` выведет приглашение для интерактивного ввода пароля
`--authenticationDatabase <dbname>`	Указывает БД, в которой содержится аутентификационная информация и полномочия пользователей. По умолчанию `mongoimport` полагает, что БД, указанная в качестве адреса БД, содержит аутентификационную информацию и полномочия пользователей, если не использована опция `--authenticationDatabase`
`--authenticationMechanism <name>`	Указывает аутентификационный механизм. По умолчанию аутентификационным механизмом является MONGODB-CR, который является механизмом типа «запрос/ответ». В ДОСУБД реализована поддержка GSSAPI для поддержки Kerberos-аутентификации

Продолжение таблицы 14

Опция	Описание
`--dbpath <path>`	Указывает путь к каталогу с файлами ДОСУБД. При использовании опция `--dbpath` позволяет `mongoexport` получать данные непосредственно из локальных файлов, не используя `mongod`. Для использования опции `--dbpath` `mongoexport` должна заблокировать доступ к каталогу данных. В результате ни один экземпляр `mongod` не может получить доступ к каталогу с тем же путем
`--directoryperdb`	Используется в сочетании с соответствующей опцией `mongod`, которая позволяет `mongoexport` экспортировать данные из экземпляра ДОСУБД, который хранит каждый файл данных в каталоге на диске. Данная опция имеет смысл только при указании опции `--dbpath`
`--journal`	Позволяет использовать специальный журнал для обеспечения согласованности экспортируемых данных. Данная опция имеет смысл только при указании опции `--dbpath`
`--db <db>, -d <db>`	Указывает БД, которая содержит коллекцию для экспорта данных
`--collection <collection>, -c <collection>`	Указывает коллекцию для экспорта
`--fields <field1<,field2>>, -f <field1[,field2]>`	Задает поле или поля для включения в экспорт. Используется список имен полей, разделенных запятыми, для задания множества полей. Для выходного формата CSV `mongoexport` включает только указанные поля, которые могут быть полями в пределах поддокумента. Для выходного формата JSON `mongoexport` включает только указанные поля и поле `_id`. Если указанное поле является полем в пределах поддокумента, то `mongoexport` включает поддокумент со всеми его полями
`--fieldFile <filename>`	Является альтернативой для опции `--fields` и позволяет указать в файле поле или поля для включения в экспорт. Опция действительна только в сочетании с опцией `--csv`. Файл должен иметь только одно поле в строке и строки должны заканчиваться символом перевода строки (`0x0A`). `Mongoexport` включает только указанные поля, которые могут быть полями в пределах поддокумента

Продолжение таблицы 14

Опция	Описание

`--query <JSON>, -q <JSON>`	Выполняет запрос документов в формате JSON, дополнительно ограничивая возвращаемые в экспорте документы. Существует коллекция с именем «test» в БД «records» со следующими документами:
	`{"_id":ObjectId("51f0188846a64a1ed98fde7c"), "a":1}`
	`{"_id":ObjectId("520e61b0c6646578e3661b59"), "a":1,"b":2}`
	`{"_id":ObjectId("520e642bb7fa4ea22d6b1871"), "a":2,"b":3,"c":5}`
	`{"_id":ObjectId("520e6431b7fa4ea22d6b1872"), "a":3,"b":3,"c":6}`
	`{"_id":ObjectId("520e6445b7fa4ea22d6b1873"), "a":5,"b":6,"c":8}`
	
	И следующая команда `mongoexport`:
	`mongoexport -d test -c records -q "{\$or:[{a:{\$gte:3}},{b:{\$lte:2}}]}"`
	
	Результирующая коллекция включает следующие документы:
	`{"_id":{"$oid":"520e61b0c6646578e3661b59"}, "a":1,"b":2}`
	`{"_id":{"$oid":"520e6431b7fa4ea22d6b1872"}, "a":3,"b":3,"c":6}`
	`{"_id":{"$oid":"520e6445b7fa4ea22d6b1873"}, "a":5,"b":6,"c":8}`

`--csv`	Изменяет формат экспорта на CSV. По умолчанию `mongoexport` записывает данные используя один JSON-документ для каждого документа ДОСУБД. Если указана опция `--csv`, то необходимо также использовать опцию `--fields` или `--fieldFile` для объявления экспортируемых из коллекции полей
`--jsonArray`	Модифицирует формат выходных данных `mongoexport` для записи всего экспортируемого содержимого как одного JSON-массива. По умолчанию `mongoexport` записывает данные, используя один JSON-документ для каждого документа ДОСУБД
`--slaveOk, -k`	Позволяет `mongoexport` читать данные вторичного или подчиненного узлов, используя `mongoexport` с набором реплики. Данная опция доступна, только если подключение осуществляется к `mongod` или `mongos`, и опция не доступна, если используется опция `--dbpath`. Данное поведение является поведением по умолчанию

Окончание таблицы 14

Опция	Описание
`--out <file>, -o <file>`	Указывает файл для записи экспортируемых данных. Если не указано имя файла, то `mongoexport` записывает данные в стандартный поток вывода
`--forceTableScan`	Заставляет `mongoexport` сканировать непосредственно хранилище данных: в общем случае, `mongoexport` сохраняет записи в порядке их следования в индексном поле `_id`. Опция `--forceTableScan` используется для пропуска индекса и непосредственного сканирования данных. Обычно существует два случая, в которых данное поведение предпочтительнее по умолчанию: если есть ключ размером более 800 байт, который не будет присутствовать в индексе `_id`; БД использует настраиваемое поле `_id`. При использовании опции `--forceTableScan` `mongoexport` не использует `$snapshot`. В результате экспорт, выполняемый `mongoexport`, может отражать состояние БД в множестве разных моментов времени. **ВНИМАНИЕ!** Необходимо использовать опцию `--forceTableScan` в особой осторожностью и вниманием

8.3.2. Использование mongoexport

Примеры:

1. В данном примере утилита `mongoexport` экспортирует коллекцию `contacts` из БД `users` экземпляра `mongod`, выполняющегося на локальном хосте с номером порта `27017`. Команда записывает экспортируемые данные в формате CSV в файл `/opt/backups/contacts.csv`. Файл `fields.txt` содержит разделенный построчно список экспортируемых полей:

```
mongoexport --db users --collection contacts --csv
    --fieldFile fields.txt --out /opt/backups/contacts.csv
```

2. В данном примере экспортируется коллекция `contacts` из экземпляра ДОСУБД, выполняющегося на локальном хосте с номером порта `27017`, с явно включенным журналированием. Команда записывает экспортируемые данные в файл `contacts.json` в формате JSON:

```
mongoexport --db sales --collection contacts --out contacts.json
    --journal
```

3. В данном примере экспортируется коллекция `contacts` из БД `sales`, размещенной в файлах данных ДОСУБД в каталоге `/srv/mongodb/`. Команда записывает экспортируемые данные в стандартный выходной поток в формате JSON:

```
mongoexport --db sales --collection contacts --dbpath /srv/mongodb/
```

ВНИМАНИЕ! Данный пример успешно выполняется только в случае если ни

один экземпляр mongod не использует файлы данных, размещенные в каталоге /srv/mongodb/.

4. В данном примере экспортируется коллекция contacts из БД marketing. Данные расположены в экземпляре ДОСУБД, выполняющемся на хосте mongodb1.example.net с номером порта 37017, и требующем имя пользователя user с паролем pass:

```
mongoexport --host mongodb1.example.net --port 37017 --username user
    --password pass --collection contacts
    --db marketing --out mdb1-examplenet.json
```

8.4. Средство работы с файлами mongofiles

Утилита mongofiles предоставляет возможность выполнения из командной строки действий для работы с файлами, хранимыми в экземпляре ДОСУБД в объектах GridFS. Это особенно полезно, поскольку предоставляет интерфейс между объектами, хранимыми в файловой системе и GridFS.

Все команды mongofiles имеют следующий вид:

```
mongofiles <options> <commands> <filename>
```

Команда включает следующие компоненты:

– опции — options. Существует возможность использовать одну или несколько опций для управления поведением утилиты mongofiles;

– команды — commands. Используется одна из команд для указания действия утилите mongofiles;

– имя файла — filename. Указывает на объект в файловой системе или объект GridFS.

Утилита mongofiles, подобно утилитам mongodump, mongoexport, mongoimport, и mongorestore, может получать доступ к данным, хранимым в каталоге данных ДОСУБД, без использования выполняющегося экземпляра mongod.

ВНИМАНИЕ! Для набора реплики mongofiles может читать только с первичного сервера.

8.4.1. Опции mongofiles

Опции утилиты mongofiles представлены в таблице 15.

Таблица 15

Опция	Описание
--help, -h	Выводит текст базовой справки об утилите
--version	Выводит информацию о версии утилиты

Продолжение таблицы 15

Опция	Описание
`--verbose, -v`	Увеличивает количество информации, возвращаемое утилитой. Можно увеличить детализацию вывода опцией `-v`, включаемой несколько раз, например, `-vvvvv`
`--host <hostname><:port>, -h`	Указывает разрешаемое имя хоста, с который содержит GridFS. По умолчанию `mongofiles` осуществляет попытку подключения к процессу ДОСУБД, выполняющемуся на локальном хосте с номером порта `27017`. Дополнительно можно указать номер порта для подключения к экземпляру ДОСУБД, использующему номер порта, отличный от `27017`
`--port <port>`	Указывает порт, используемый экземпляром ДОСУБД в случае, если номер порта отличается от `27017`. Возможно также указать номер порта в опции `--host`
`--ipv6`	Включает поддержку IPv6, которая позволяет `mongofiles` подключаться к экземпляру ДОСУБД, используя сети IPv6. Во всех программах и процессах ДОСУБД, включая `mongofiles`, поддержка IPv6 по умолчанию отключена
`--ssl`	Предоставляет возможность подключения к экземплярам `mongod` с использованием SSL
`--username <username>, -u <username>`	Указывает имя пользователя для аутентификации в ДОСУБД, если БД требует аутентификации. Используется в сочетании с опцией `--password`, задающей пароль
`--password <password>, -p <password>`	Указывает пароль для аутентификации в экземпляре ДОСУБД. Используется в сочетании с опцией `--username`, задающей имя пользователя. Если опция `--username` использована без опции `--password`, то `mongofiles` выведет приглашение для интерактивного ввода пароля
`--authenticationDatabase <dbname>`	Указывает БД, в которой содержится аутентификационная информация и полномочия пользователей. По умолчанию `mongofiles` полагает, что БД, указанная в качестве адреса БД, содержит аутентификационную информацию и полномочия пользователей, если не использована опция `--authenticationDatabase`
`--authenticationMechanism <name>`	Указывает аутентификационный механизм. По умолчанию аутентификационным механизмом является MONGODB-CR, который является механизмом типа «запрос/ответ». В ДОСУБД реализована поддержка GSSAPI для поддержки Kerberos-аутентификации
`--dbpath <path>`	Указывает путь к каталогу с файлами ДОСУБД. При использовании опция `--dbpath` позволяет `mongofiles` получать данные GridFS непосредственно из локальных файлов, не используя `mongod`. Для использования опции `--dbpath` `mongofiles` должна заблокировать доступ к каталогу данных. В результате ни один экземпляр `mongod` не может получить доступ к каталогу с тем же путем

Окончание таблицы 15

Опция	Описание
`--directoryperdb`	Используется в сочетании с соответствующей опцией `mongod`, которая позволяет `mongofiles` в сочетании опцией `--dbpath` хранить каждый файл данных в каталоге на диске. Данная опция имеет смысл только при указании опции `--dbpath`
`--journal`	Позволяет использовать специальный журнал, обеспечивающий при указании опции `--dbpath` нахождение БД в состоянии, пригодном для восстановления. Данная опция имеет смысл только при указании опции `--dbpath`
`--db <db>, -d <db>`	Указывает БД, в которой хранятся или будут храниться файлы GridFS
`--collection <collection>, -c <collection>`	Данная опция не используется
`--local <filename>, -l <filename>`	Задает имя файла в локальной файловой системе для выполнения операций `put` и `get`. В командах `mongofiles put` и `mongofiles get` требуется модификатор `<filename>`, ссылающийся на имя, которое будет иметь объект в GridFS
`--type <MIME>, t <MIME>`	Предоставляет возможность указать тип MIME для описания файла, вставленного в хранилище GridFS. `mongofiles` по умолчаню в операциях опускает данную опцию. Опция должна использоваться только в операциях `put` утилиты `mongofiles`
`--replace, -r`	Изменяет поведение операции `put` утилиты `mongofiles` для замены существующего объекта GridFS из указанного локального файла вместо добавления объекта с таким же именем. По умолчанию объект не будет перезаписываться операцией `put` утилиты `mongofiles`

8.4.2. Команды mongofiles

Команды утилиты `mongofiles` представлены в таблице 16.

Таблица 16

Команда	Описание
`list <prefix>`	Выводит список файлов в хранилище GridFS. Символы, указанные после `list` (например, `<prefix>`), дополнительно ограничивают список возвращаемых элементов для файлов, которые начинаются с указанной строки символов
`search <string>`	Выводит список файлов в хранилище GridFSc именами, совпадающими с любой частью `<string>`

Окончание таблицы 16

Команда	Описание
`put <filename>`	Копирует указанный файл из локальной файловой системы в хранилище GridFS. Здесь `<filename>` ссылается на имя, которое будет иметь объект в GridFS, и `mongofiles` полагает, что указано имя, которое файл имеет в локальной файловой системе. Если имя локального файла отличается, то необходимо использовать опцию `--local`.
`get <filename>`	Копирует указанный файл из хранилища GridFS в локальную файловую систему. Здесь `<filename>` ссылается на имя, которое будет иметь файл в локальной файловой системе и `mongofiles` полагает, что указано имя объекта, которое имеет объект в GridFS. Если имя локального файла отличается, то необходимо использовать опцию `--local`
`delete <filename>`	Удаляет указанный файл из хранилища

8.4.3. Использование mongofiles

Примеры:

1. Для получения списка всех файлов в коллекции GridFS в БД `records` используется следующее обращение к файловой системе:

```
mongofiles -d records list
```

В данном примере `mongofiles` будет подключаться к экземпляру `mongod`, выполняющемуся на интерфейсе `27017 localhost`.

2. Для выполнения операции, аналогичной предыдущему примеру с другим номером порта или именем хоста, используется одна из следующих команд:

```
mongofiles --port 37017 -d records list
mongofiles --hostname db1.example.net -d records list
mongofiles --hostname db1.example.net --port 37017 -d records list
```

3. Для загрузки файла с именем `32-corinth.lp` в коллекцию GridFS в БД `records` можно выполнить следующую команду:

```
mongofiles -d records put 32-corinth.lp
```

4. Для удаления файла с именем `32-corinth.lp` из коллекции GridFS в БД `records` можно выполнить следующую команду:

```
mongofiles -d records delete 32-corinth.lp
```

5. Для поиска файлов в коллекции GridFS в БД `records`, которые имеют строку `corinth` в их именах можно выполнить следующую команду:

```
mongofiles -d records search corinth
```

6. Для вывода списка всех файлов в коллекции GridFS в БД `records`, которые начинаются со строки `32` в их именах, можно выполнить следующую команду:

```
mongofiles -d records list 32
```

7. Для извлечения файла с именем 32-corinth.lp из коллекции GridFS в БД records можно выполнить следующую команду:

```
mongofiles -d records get 32-corinth.lp
```

8.5. Средство резервного копирования mongodump

Утилита mongodump предназначена для бинарного экспорта содержимого БД. Использование данной утилиты является частью эффективной стратегии резервного копирования. Необходимо использовать утилиту mongodump в сочетании с утилитой mongorestore для обеспечения возможности восстановления БД. mongodump может читать данные из экземпляра mongod, mongos или осуществлять прямой доступ к файлам данных ДОСУБД без использования активного mongod.

ВНИМАНИЕ! Утилита mongodump не выполняет экспорт из БД local.

8.5.1. Опции mongodump

Опции утилиты mongodump представлены в таблице 17.

Таблица 17

Опция	Описание
--help, -h	Выводит текст базовой справки об утилите
--version	Выводит информацию о версии утилиты
--verbose, -v	Увеличивает количество информации, возвращаемое утилитой. Можно увеличить детализацию вывода опцией -v, включаемой несколько раз, например, -vvvvv

--host <hostname><:port>, -h	Указывает разрешаемое имя хоста, на котором функционирует экземпляр mongod, который будет использоваться для создания резервной копии БД. По умолчанию mongodump осуществляет попытку подключения к процессу ДОСУБД, выполняющемуся на локальном хосте с номером порта 27017. Дополнительно можно указать номер порта для подключения к экземпляру ДОСУБД, использующему номер порта, отличный от 27017. Для подключения к набору реплики опция --host используется с именем набора, за которым следует символ / и список имен хостов с номерами портов, разделенных запятыми. Утилита mongodump будет выполнять подключение к первом включенному члену набора реплики. Например:
	`mongodump --host repl0/mongo0.example.net,`
	`mongo0.example.net:27018,mongo1.example.net,`
	`mongo2.example.net`
	Возможно подключение непосредственно к экземпляру ДОСУБД посредством указания конкретных имени хоста и номера порта

Продолжение таблицы 17

Опция	Описание
`--port <port>`	Указывает порт, используемый экземпляром ДОСУБД в случае, если номер порта отличается от `27017`. Возможно также указать номер порта в опции `--host`
`--ipv6`	Включает поддержку IPv6, которая позволяет `mongodump` подключаться к экземпляру ДОСУБД, используя сети IPv6. Во всех программах и процессах ДОСУБД, включая `mongodump`, поддержка IPv6 по умолчанию отключена
`--ssl`	Предоставляет возможность подключения к экземплярам `mongod` с использованием SSL
`--username <username>, -u <username>`	Указывает имя пользователя для аутентификации в ДОСУБД, если БД требует аутентификации. Используется в сочетании с опцией `--password`, задающей пароль
`--password <password>, -p <password>`	Указывает пароль для аутентификации в экземпляре ДОСУБД. Используется в сочетании с опцией `--username`, задающей имя пользователя. Если опция `--username` использована без опции `--password`, то `mongodump` выведет приглашение для интерактивного ввода пароля
`--authenticationDatabase <dbname>`	Указывает БД, в которой содержится аутентификационная информация и полномочия пользователей. По умолчанию `mongodump` полагает, что БД, указанная в качестве адреса БД, содержит аутентификационную информацию и полномочия пользователей, если не использована опция `--authenticationDatabase`
`--authenticationMechanism <name>`	Указывает аутентификационный механизм. По умолчанию аутентификационным механизмом является MONGODB-CR, который является механизмом типа «запрос/ответ». В ДОСУБД реализована поддержка GSSAPI для поддержки Kerberos-аутентификации
`--dbpath <path>`	Указывает путь к каталогу с файлами ДОСУБД. При использовании опция `--dbpath` позволяет `mongodump` копировать данные непосредственно из локальных файлов данных, не используя `mongod`. Для использования опции `--dbpath` `mongodump` должна заблокировать доступ к каталогу данных. В результате ни один экземпляр `mongod` не может получить доступ к каталогу с тем же путем
`--directoryperdb`	Используется в сочетании с соответствующей опцией `mongod`, которая позволяет `mongodump` читать файлы данных связанные с БД, размещенной в отдельном каталоге. Данная опция имеет смысл только при указании опции `--dbpath`
`--journal`	Позволяет операциям `mongodump` использовать специальный журнал для обеспечения согласованности экспортируемых данных. Данная опция имеет смысл только при указании опции `--dbpath`

Продолжение таблицы 17

Опция	Описание
`--db <db>, -d <db>`	Указывает БД для резервного копирования данных. Если БД не указана, то `mongodump` копирует все БД в экземпляре ДОСУБД. Опция используется для уменьшения объема данных, подлежащих резервному копированию
`--collection <collection>, -c <collection>`	Указывает коллекцию для резервного копирования данных. Если коллекция не указана, то `mongodump` копирует все коллекции в указанной БД или экземпляре ДОСУБД. Опция используется для уменьшения объема данных, подлежащих резервному копированию.
`--out <path>, -o <path>`	Задает каталог, в котором `mongodump` сохраняет выходные данные при резервном копировании БД. По умолчанию `mongodump` сохраняет выходные данные в каталоге с именем резервной копии в текущем рабочем каталоге. Для направления резервной копии БД в стандартный поток вывода указывается – вместо пути. Запись в стандартный поток вывода выполняется при необходимости обработать результаты резервного копирования перед их сохранением. Например, выполнить сжатие с использованием утилиты `gzip`. При направлении в стандартный поток вывода `mongodump` не записывает метаданные, которые записываются в файл `<dbname>.metadata.json` при выводе непосредственно в файлы
`--query <json>, -q <json>`	Формирует запрос для дополнительного ограничения набора документов, включаемых в выходные данные `mongodump`.
`--oplog`	Используется для создания резервной копии БД, включающей коллекцию `oplog` для создания снимка состояния экземпляра ДОСУБД. Для восстановления резервной копии снимка состояния используется резервная копия, созданная с данной опцией в сочетании с опцией `--oplogReplay` утилиты `mongorestore`. Без использования опции `--oplog` при наличии операций записи во время выполнения резервного копирования выходные данные не будут отражать состояние в определенный момент времени. Изменения, сделанные в БД в процессе обновления, могут повлиять на выходные данные резервного копирования. Опция `--oplog` не используется при выполнении `mongodump` для экземпляра `mongos` в целях резервного копирования всего содержимого кластера шардинга. Однако опция `--oplog` используется для резервного копирования данных индивидуальных узлов шардинга. **ВНИМАНИЕ!** Опция `--oplog` используется только для узлов, которые поддерживают коллекцию `oplog`: все члены набора реплики и узлы `master` в схемах репликации `master-slave`

Окончание таблицы 17

Опция	Описание
`--repair`	Используется для выполнения операции восстановления совместно с операцией резервного копирования. С опцией `repair` выполняется попытка восстановления БД, которая может находиться в несогласованном состоянии в результате сбоя `mongod` или некорректного завершения работы ОС СН. **ВНИМАНИЕ!** Опция `--repair` использует агрессивные алгоритмы восстановления данных, что может привести к возникновению значительного количества дубликатов
`--forceTableScan`	Принуждает mongodump к прямому сканированию хранилища данных: по умолчанию `mongodump` сохраняет документы в порядке их появления в поле `_id`. Опция `--forceTableScan` используется для пропуска индекса и непосредственного сканирования данных. В общем случае существует два варианта, в которых указанное поведение предпочтительнее поведения по умолчанию: – при использовании ключей, размер которых превышает 800 байт, данные ключи не будут присутствовать в индексе `_id`; – при использовании в БД настраиваемого поля `_id`. При выполнении с опцией `--forceTableScan` утилита `mongodump` не использует снимок `$snapshot`. В результате созданная `mongodump` резервная копия может отражать состояние БД в множестве разных моментов времени. **ВНИМАНИЕ!** Необходимо использовать опцию `--forceTableScan` с особой осторожностью и вниманием

8.5.2. Поведение mongodump

При выполнении `mongodump` для экземпляра `mongos` в случае когда кластер шардинга состоит из наборов реплик, операции чтения будут осуществляется в первую очередь со вторичных членов набора реплики.

ВНИМАНИЕ! При использовании в сочетании с `fsync` или `db.fsyncLock()` `mongod` может блокировать некоторые операции чтения, включая те операции `mongodump`, за которыми следуют операции записи, ожидающие снятия блокировки `fsync`.

8.5.3. Необходимые привилегии пользователя

Пользователь должен иметь соответствующие привилегии для чтения данных с использованием `mongodump` из БД, содержащих коллекции. Привилегии, необходимые для выполнения операций `mongodump`, приведены в таблице 18.

Таблица 18

Задача	Необходимые привилегии
Резервное копирование всех коллекций в БД за исключением system.users	read Может потребоваться readWrite вместо read
Резервное копирование всех коллекций в БД, включая system.users	read и userAdmin Может потребоваться readWrite вместо read
Резервное копирование всех БД	readAnyDatabase, userAdminAnyDatabase и clusterAdmin Привилегия clusterAdmin предоставляет возможность выполнения команд listDatabases для получения списка существующих БД. Если БД используется с включенным профилированием, то mongodump может потребоваться привилегия dbAdminAnyDatabase для резервного копирования коллекции system.profile

8.5.4. Использование mongodump

Примеры:

1. Следующая команда создает резервную копию данных, содержащую только коллекцию с именем collection в БД с именем test. В данном случае ДОСУБД выполняется на локальном интерфейсе с номером порта 27017:

```
mongodump --collection collection --db test
```

2. В данном примере утилита mongodump создает резервную копию данных экземпляра БД, хранимого в каталоге /srv/mongodb на локальной машине. Необходимо, чтобы ни один экземпляр mongod не использовал каталог /srv/mongodb.

```
mongodump --dbpath /srv/mongodb
```

3. В данном примере утилита mongodump создает резервную копию данных БД и сохраняет ее в /opt/backup/mongodump-2011-10-24 из экземпляра ДОСУБД, выполняющегося с номером порта 37017 на узле mongodb1.example.net, используя для аутентификации имя пользователя user и пароль pass:

```
mongodump --host mongodb1.example.net --port 37017 --username user
    --password pass --out /opt/backup/mongodump-2011-10-24
```

8.6. Средство восстановления резервных копий mongorestore

Утилита mongodump предназначена для записи данных из бинарной резервной копии, созданной утилитой mongodump для экземпляра БД. Утилита mongorestore может создать новую БД или добавить данные в существующую БД.

Утилита mongorestore может записывать данные в экземпляры mongod или

mongos в дополнение к записи непосредственно в файлы данных ДОСУБД без использования активного mongod.

Использование данной утилиты является частью эффективной стратегии резервного копирования. Необходимо использовать утилиту mongodump в сочетании с утилитой mongorestore для обеспечения возможности восстановления БД. mongodump может читать данные из экземпляра mongod, mongos или осуществлять прямой доступ к файлам данных ДОСУБД без использования активного mongod.

Для восстановления данных в существующую БД утилита mongorestore будет осуществлять вставку в существующую БД и не будет выполнять какие-либо обновления. Если существующий документ имеет тоже значение поля _id в целевой БД и коллекции, утилита mongorestore не будет перезаписывать такие документы.

ВНИМАНИЕ! Необходимо помнить следующие особенности поведения mongorestore:

– утилита mongorestore создает индексы, записанные утилитой mongodump;

– утилита mongorestore не ожидает ответа от mongod для обеспечения гарантии, что процесс ДОСУБД получил и записал данные;

– экземпляр mongod будет записывать в журнал любые ошибки, которые возникают при выполнении операции восстановления, mongorestore но не будет получать данные об ошибках.

8.6.1. Опции mongorestore

Опции утилиты mongorestore представлены в таблице 19.

Таблица 19

Опция	Описание
--help, -h	Выводит текст базовой справки об утилите
--version	Выводит информацию о версии утилиты
--verbose, -v	Увеличивает количество информации, возвращаемое утилитой. Можно увеличить детализацию вывода опцией -v, включаемой несколько раз, например, -vvvvv

Продолжение таблицы 19

Опция	Описание

`--host <hostname><:port>, -h`	Указывает разрешаемое имя хоста, на котором функционирует экземпляр `mongod`, который будет использования для создания резервной копии БД. По умолчанию `mongorestore` осуществляет попытку подключения к процессу ДОСУБД, выполняющемуся на локальном хосте с номером порта `27017`. Дополнительно можно указать номер порта для подключения к экземпляру ДОСУБД, использующему номер порта, отличный от `27017`. Для подключения к набору реплики опция `--host` используется с именем набора, за которым следует символ `/` и список имен хостов с номерами портов, разделенных запятыми. Утилита `mongorestore` будет выполнять подключение к первому включенному члену набора реплики. Например:
	`mongorestore --host repl0/mongo0.example.net,`
	`mongo0.example.net:27018,mongo1.example.net,`
	`mongo2.example.net`
	Возможно подключение непосредственно к экземпляру ДОСУБД посредством указания конкретных имени хоста и номера порта

`--port <port>`	Указывает порт, используемый экземпляром ДОСУБД в случае, если номер порта отличается от `27017`. Возможно также указать номер порта в опции `--host`
`--ipv6`	Включает поддержку IPv6, которая позволяет `mongorestore` подключаться к экземпляру ДОСУБД, используя сети IPv6. Во всех программах и процессах ДОСУБД, включая `mongodump`, поддержка IPv6 по умолчанию отключена
`--ssl`	Предоставляет возможность подключения к экземплярам `mongod` с использованием SSL
`--username <username>, -u <username>`	Указывает имя пользователя для аутентификации в ДОСУБД, если БД требует аутентификации. Используется в сочетании с опцией `--password`, задающей пароль
`--password <password>, -p <password>`	Указывает пароль для аутентификации в экземпляре ДОСУБД. Используется в сочетании с опцией `--username`, задающей имя пользователя. Если опция `--username` использована без опции `--password`, то `mongorestore` выведет приглашение для интерактивного ввода пароля
`--authenticationDatabase <dbname>`	Указывает БД, в которой содержится аутентификационная информация и полномочия пользователей. По умолчанию `mongorestore` полагает, что БД, указанная в качестве адреса БД, содержит аутентификационную информацию и полномочия пользователей, если не использована опция `--authenticationDatabase`
`--authenticationMechanism <name>`	Указывает аутентификационный механизм. По умолчанию аутентификационным механизмом является MONGODB-CR, который является механизмом типа «запрос/ответ». В ДОСУБД реализована поддержка GSSAPI для поддержки Kerberos-аутентификации

Продолжение таблицы 19

Опция	Описание
`--dbpath <path>`	Указывает путь к каталогу с файлами ДОСУБД. При использовании опция `--dbpath` позволяет `mongorestore` вставлять данные непосредственно в локальные файлы данных, не используя `mongod`. Для использования опции `--dbpath` `mongorestore` должна заблокировать доступ к каталогу данных. В результате ни один экземпляр `mongod` не может получить доступ к каталогу с тем же путем
`--directoryperdb`	Используется в сочетании с соответствующей опцией `mongod`, которая позволяет `mongorestore` импортировать данные в экземпляры ДОСУБД, в которых файлы каждой БД хранятся в отдельных каталогах на диске. Данная опция имеет смысл только при указании опции `--dbpath`
`--journal`	Позволяет `mongorestore` осуществлять запись в специальный журнал для обеспечения согласованности экспортируемых данных в процессе восстановления данных. Данная опция имеет смысл только при указании опции `--dbpath`
`--db <db>, -d <db>`	Указывает БД для восстановления резервной копии данных. Если указанная БД не существует, то `mongorestore` создаст указанную БД. Если БД не указана, то `mongorestore` создаст новую копию БД, соответствующую БД, данные из которой восстанавливаются, и данные могут быть перезаписаны. Опция используется для восстановления данных в экземпляр ДОСУБД, в котором уже содержатся данные. Опция `--db` не контролирует какие файлы BSON восстанавливает утилита `mongorestore`. Необходимо использовать опцию `path` утилиты `mongorestore` для ограничения объема восстанавливаемых данных
`--collection <collection>, -c <collection>`	Указывает коллекцию для восстановления резервной копии данных. Если коллекция не указана, то `mongorestore` импортирует все созданные коллекции. Существующие данные могут быть перезаписаны. Опция используется для восстановления данных в экземпляр ДОСУБД, в котором уже содержатся данные, или для восстановления некоторой части указанного набора импортируемых данных
`--objcheck`	Принуждает `mongorestore` проверять все запросы для обеспечения невозможности вставки недействительных документов в БД. Для объектов с высокой степенью вложенности поддокументов опция `--objcheck` может привести к небольшому снижению производительности. Можно использовать опцию `--noobjcheck` для отключения проверки. ДОСУБД использует `--objcheck` по умолчанию для предотвращения вставки некорректного или недействительного файла BSON в БД
`--noobjcheck`	Отключает проверку, выполняемую ДОСУБД по умолчанию, для всех входящих документов BSON

Окончание таблицы 19

Опция	Описание
`--filter '<JSON>'`	Ограничивает набор импортируемых `mongorestore` документов теми, которые соответствуют документу JSON, указанному как `'<JSON>'`. Необходимо заключить документ в одинарные кавычки во избежание взаимодействия с системным окружением оболочки
`--drop`	Модифицирует процедуру восстановления таким образом, чтобы при восстановлении данных из резервной копии все коллекции из целевой БД были пропущены
`--oplogReplay`	Использует коллекцию `oplog` после восстановления резервной копии для обеспечения гарантии, что текущее состояние БД отражает состояние в определенный момент времени, зафиксированное командой `mongodump --oplog`
`--keepIndexVersion`	Предотвращает модернизацию утилитой `mongorestore` индексов до следующей версии в процессе восстановления
`--w <number of replicas per write>`	Определяет, что необходимо контролировать каждую операцию записи, выполняемую утилитой `mongorestore` в целевую БД. По умолчанию mongorestore не ожидает ответа для подтверждения записи
`--noOptionsRestore`	Предотвращает восстановление утилитой `mongorestore` опций для восстанавливаемых коллекций
`--noIndexRestore`	Предотвращает создание и построение утилитой `mongorestore` индексов, указанных в соответствующем выводе `mongodump`
`--oplogLimit <timestamp>`	Предотвращает применение утилитой `mongorestore` записей `oplog` более новых чем `<timestamp>`. Значение `<timestamp>` указывается в форме `<time_t>:<ordinal>`, где `<time_t>` задается в секундах с начала UNIX-эпохи и `<ordinal>` представляет собой число операций в `oplog`, которые выполнены с указанного числа секунд. Необходимо использовать опцию `--oplogLimit` в сочетании с опцией `--oplogReplay`
`<path>`	Финальный аргумент команды `mongorestore` является путем к каталогу, в котором размещается резервная копия БД для восстановлени

8.6.2. Использование mongorestore

Примеры:

1. Следующая команда читает резервную копию данных в подкаталоге `dump/` текущего каталога и восстанавливает только документы в коллекции с именем `people` из БД с именем `accounts`. Утилита `mongorestore` восстанавливает данные в экземпляр ДОСУБД выполняющийся на локальном интерфейсе с номером порта `27017`:

```
mongorestore --collection people --db accounts dump/accounts/people.bson
```

2. В данном примере утилита `mongorestore` восстанавливает резервную копию в БД, хранимую в каталоге `/srv/mongodb` на локальной машине. Необходимо, чтобы ни один экземпляр mongod не использовал каталог `/srv/mongodb`.

```
mongorestore --dbpath /srv/mongodb
```

3. В данном примере утилита `mongorestore` восстанавливает резервную копию данных БД, расположеную в `/opt/backup/mongodump-2011-10-24` в экземпляр ДОСУБД, выполняющийся с номером порта `37017` на узле `mongodb1.example.net`, используя для аутентификации имя пользователя `user` и пароль `pass`:

```
mongorestore --host mongodb1.example.net --port 37017 --username user
    --password pass /opt/backup/mongodump-2011-10-24
```

9. ВЗАИМОДЕЙСТВИЕ АДМИНИСТРАТОРА С СЗИ

Взаимодействие администратора с СЗИ ДОСУБД состоит из обязательного прохождения процедуры аутентификации на сервере ДОСУБД и работы в условиях применения дискреционных и мандатных правил разграничения доступа.

Также предусмотрена возможность санкционированного изменения правил разграничения доступа и регистрации событий.

9.1. Управление учетными записями пользователей

В ДОСУБД применяется два способа аутентификации:

– MONGODB-CR — базовый механизм аутентификации на основе пары (имя пользователя, пароль);

– SASL GSSAPI — механизм GSSAPI средства аутентификации SASL для аутентификации по протоколу Kerberos.

ВНИМАНИЕ! Перед включением режима аутентификации необходимо наличие хотя бы одного зарегистрированного пользователя, имеющего доступ к администрированию.

Если ЕПП не используется, то по умолчанию в ДОСУБД используется базовый механизм аутентификации (MONGODB-CR) на основе пары (имя пользователя, пароль). Использование указанного механизма требует ввода пользователем пароля при каждой установке сессии с ДОСУБД.

Документ, содержащий привилегии пользователя и аутентификационную информацию, хранится в системной коллекции `<database>.system.users` в следующем виде:

```
{
 user: "<username>",
 pwd: "<hash>",
 roles: []
}
{
 user: "<username>",
 userSource: "<database>",
 roles: []
}
```

Когда аутентификационная информация находится непосредственно в БД, то указывается параметр `pwd`, значением которого является хеш пароля. В случае когда аутентификационная информация находится в другой БД, указывается параметр `userSource`, значением которого является имя БД, содержащей аутентификационную информацию пользователя.

ВНИМАНИЕ! Аутентификация пользователя должна выполнятся в той БД, в которой хранится аутентификационная информация пользователя.

Аутентификация может выполняться указанием опций командной строки утилит, описанных в разделе 8, или выполнением команды `db.auth` в командной оболочке ДОСУБД `mongo`:

```
db.auth( <username>, <pwd> )
```

При использовании ЕПП аутентификация пользователей осуществляется централизованно по протоколу Kerberos. Для интеграции ДОСУБД с Astra Linux Directory и аутентификации через Kerberos используется механизм GSSAPI средства аутентификации SASL. При этом в качестве аутентификационной БД `userSource` используется специальное значение `$external`.

В этом случае для подключения к экземпляру ДОСУБД пользователя ДОСУБД необходимо наличие билета Kerberos, полученного при успешном входе одноименного ALD-пользователя ОС СН (см. раздел 6 документа РУСБ.10015-01 95 01).

Аутентификация может выполняться указанием опций командной строки утилит, описанных в разделе 8, например:

```
mongo --authenticationMechanism=GSSAPI
      --authenticationDatabase='$external'
      --username application/reporting@EXAMPLE.NET
```

Или выполнением команды `db.auth` в командной оболочке ДОСУБД `mongo` для специальной аутентификационной БД `$external`, например:

```
use $external
db.auth( { mechanism: "GSSAPI", user: "application/reporting" } )
```

Данная операция аутентифицирует принципала Kerberos с именем `application/reporting@EXAMPLE.NET` при подключении к `mongod`. Все доступные привилегии при необходимости будут приобретаться автоматически.

Примечание. Особенностью реализации аутентификации с помощью SASL является то, что при выполнении команд аутентификации следует опускать имя домена для домена «по умолчанию». Например, команда:

```
> db.auth( { mechanism: 'GSSAPI', user: 'user@REALM' } )
```

вызовет ошибку. Правильной в этом случае будет команда:

```
> db.auth( { mechanism: 'GSSAPI', user: 'user' } )
```

Добавление, удаление и модификация учетных записей пользователей осуществляется операциями с документами системной коллекции `<database>.system.users`.

Для удобства работы в командном интерфейсе `mongo` предусмотрена команда добавления учетной записи для доступа к базе данных `db.addUser`, в качестве аргументов

которой может выступать либо пара (имя пользователя, пароль), либо документ привилегий пользователя:

```
db.addUser(<имя пользователя>, <пароль>)
db.addUser({<документ привилегий пользователя>})
```

Примечание. В случае невозможности использования метки соединения, может быть выбран режим задания метки сессии по максимально допустимому уровню доступа пользователя с помощью поля признака игнорирования мандатной метки сетевого соединения `ignoreSocketMAC` документа привилегий пользователя указанием значения `true`.

ВНИМАНИЕ! Признак игнорирования мандатной метки сетевого соединения `ignoreSocketMAC` должен использоваться только в случае крайней необходимости для реализации специальных процедур обработки данных.

9.2. Дискреционное разграничение доступа в ДОСУБД

В ДОСУБД обеспечивается дискреционное разграничение доступа на уровне БД. При этом предусматривается два вида доступа: на «чтение» и «чтение-запись». Кроме этого, отдельно рассматривается доступ к системным БД для администрирования. Наборы привилегий объединяются в роли.

В ДОСУБД представлены следующие предопределенные роли:

1) Пользовательские роли:
   а) `read` — чтение любой коллекции определенной БД с предоставлением доступа к набору команд, использующих чтение или поиск документов;
   б) `readWrite` — чтение и запись любой коллекции заданной БД с предоставлением доступа ко всем командам роли `read`, базовым командам `insert()`, `update()`, `remove()` и командам, использующих запись документов.
2) Роли администратора БД:
   а) `dbAdmin` — выполнение служебных операций в пределах определенной БД, включая создание новых коллекций;
   б) `userAdmin` — выполнение служебных операций в системной БД пользователей, включая управление ими и изменение их прав доступа.
3) Административные роли:
   а) `clusterAdmin` — управление кластерами и шардами;
   б) `userAdmin` — выполнение служебных операций в системной БД пользователей, включая управление ими и изменение их прав доступа.
4) Системные роли:
   а) `readAnyDatabase` — аналогично роли `read`, но распространяется на все БД;

б) readWriteAnyDatabase — аналогично роли readWrite, но распространяется на все БД;

в) userAdminAnyDatabase — аналогично роли userAdmin, но распространяется на все БД;

г) dbAdminAnyDatabase — аналогично роли dbAdmin, но распространяется на все БД.

Роли пользователя задаются в массиве {roles} документа привилегий, описанного в 9.1.

9.3. Мандатное разграничение доступа в ДОСУБД

В ДОСУБД реализована работа с мандатными атрибутами (метками) объектов доступа. Метка по составу соответствует метке ОС СН и содержит два поля: иерархический мандатный атрибут – уровень конфиденциальности и набор неиерархических мандатных атрибутов – категорий.

Мандатная метка документа размещается в специальном поле _mac в текстовом виде "{Уровень,Категории}", где первым элементом является уровень конфиденциальности в десятичном выражении, а вторым — набор мандатных категорий в шестнадцатеричном выражении (например, “{2,1}”). Символ подчеркивания в имени поля отличает его как системное поле (аналогично полю _id). Использование текстового поля не нарушает стандарта BSON и позволяет использовать существующие правила преобразования в JSON.

При работе с контейнерами (БД и коллекциями) используется признак применения мандатных атрибутов контейнера _ccr. Если значение признака _ccr равно FALSE, то разрешается читать содержимое контейнера. При этом применяются мандатные атрибуты содержимого контейнера. Если значение признака _ccr равно TRUE, то применяются мандатные атрибуты самого контейнера.

Для получения и санкционированного изменения мандатных атрибутов контейнеров и документов существует соответствующий набор команд сервера ДОСУБД. Доступ к этим командам возможен как с помощью интерактивного командного интерфейса доступа к ДОСУБД mongo, так и с помощью прикладного программного интерфейса.

Команды для работы с мандатными правилами разграничения доступа:

– macid — получение текущей метки конфиденциальности сессии;

– getMacLabel — получение метки конфиденциальности контейнера;

– chmac — изменения мандатных атрибутов (метки) контейнера;

– $chmac — операция команды update для изменения мандатных атрибутов документов.

ВНИМАНИЕ! Изменение мандатных атрибутов документов операцией `$set` команды `update` запрещено.

С помощью прикладного программного интерфейса ДОСУБД доступ к указанным командам возможен следующим образом:

– вызовом методов класса `DBClientWithCommands` библиотеки C++ драйвера (`mongo/client/dbclientinterface.h`);

– вызовом функций библиотеки C драйвера (`mongo.h`).

ВНИМАНИЕ! Помимо соответствующих дискреционных прав доступа на модифицируемый объект БД, изменение мандатных атрибутов требует наличия у пользователя мандатной привилегии `PARSEC_CAP_CHMAC`.

Команды получения метки конфиденциальности собственной сессии `macid` и получения мандатных атрибутов объектов `getMacLabel` не требуют специальных привилегий пользователя. Команды модификации мандатных атрибутов объектов требуют прав управления базой данных или коллекцией.

9.3.1. Установка мандатных атрибутов данных

При входе в ОС СН пользователь пройдет аутентификацию, и если для него установлены мандатные уровни и категории, отличные от нуля, то ему будет предложено установить конкретный мандатный уровень и конкретную категорию для данной сессии в пределах разрешенных диапазонов. Выбранные значения этих параметров можно будет проверить с помощью индикатора в виде кружка с числом внутри, расположенного в системном лотке в правом нижнем углу рабочего стола (рис. 3). Для получения информационного сообщения следует навести курсор на этот индикатор.

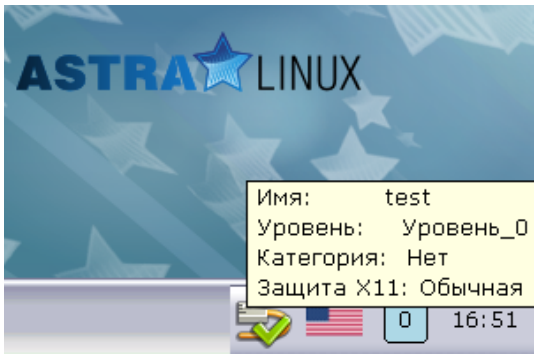

Рис. 3

Также для просмотра своих мандатных атрибутов пользователь ОС СН может воспользоваться консольной утилитой `macid`.

Запускаемые пользователем ОС СН в контексте текущей сессии процессы будут наследовать мандатные атрибуты (мандатную метку) текущей сессии. Создаваемые пользовательским процессом объекты (например, файлы и каталоги) будут наследовать ман-

датные атрибуты процесса. Непривилегированному пользователю ОС СН не предоставляются права на изменение мандатных атрибутов объектов доступа в ОС СН.

ДОСУБД является сетевым сервисом, функционирующим под управлением ОС СН, который должен обеспечивать обработку пользователями информации с различными уровнями конфиденциальности. Сетевой сервис ДОСУБД для обработки информации с различными мандатными уровнями, используя привилегии подсистемы безопасности PARSEC из состава ОС СН, открывает привилегированный слушающий сокет, имеющий возможность принимать входящие сетевые соединения с любыми мандатными метками.

При подключении к ДОСУБД клиентский процесс создаст сокет, который унаследует мандатные атрибуты процесса и обеспечит их указание в поле данных IP-пакетов, передаваемых на сетевой сервис ДОСУБД. Таким образом, при доступе пользователя ДОСУБД к объектам в БД будет использована мандатная метка сессии пользователя ОС СН.

ДОСУБД выполняет съем с сокета, обслуживающего соединение с клиентом, передаваемой по сети мандатной метки процесса, в котором выполнен запрос пользователя ДОСУБД на доступ к данным.

После проверки корректности полученной мандатной метки ДОСУБД будет выполнять мандатное разграничение доступа пользователя ДОСУБД к объектам БД с использованием указанной метки. Все объекты, создаваемые пользователем ДОСУБД в БД, будут наследовать мандатную метку сетевого соединения.

9.3.2. Получение метки конфиденциальности сессии

Для получения текущей метки конфиденциальности сессии в командном интерфейсе `mongo` необходимо выполнить:

```
> db.macid()
{0,0}
```

Примечание. Для получения метки конфиденциальности сессии непринципиально, какая именно база данных активна в командном интерфейсе.

Для получения текущей метки конфиденциальности сессии с помощью прикладного программного интерфейса C++ используется метод `macid`:

```
string DBClientWithCommands::macid();
```

Метка возвращается в текстовом виде.

Для получения текущей метки конфиденциальности сессии с помощью прикладного программного интерфейса C используется функция `mongo_cmd_macid`:

```
int mongo_cmd_macid( mongo *conn, bson *out );
```

Аргументы:

– `conn` — указатель на экземпляр установленного соединения с сервером ДОСУБД;

– `out` — BSON-документ с результатами выполнения запроса (полем `maclabel`,

содержащем метку конфиденциальности сессии).

В результате вызова функции возвращается соответствующий код ошибки.

9.3.3. Получение мандатных атрибутов базы данных или коллекции

Для получения мандатных атрибутов текущей базы данных в командном интерфейсе mongo необходимо выполнить:

```
> db.getMacLabel()
{ "_mac" : "{0,0}", "_ccr": true, "ok" : 1 }
```

Результат возвращается в виде объекта с полями _mac и _ccr, содержащих, соответственно, метку конфиденциальности и признак применения мандатных атрибутов контейнера.

Аналогично могут быть получены мандатные атрибуты выбранной коллекции текущей базы данных:

```
> db.coll0.getMacLabel()
{ "_mac" : "{0,0}", "_ccr": true, "ok" : 1 }
```

Для получения мандатных атрибутов контейнера с помощью прикладного программного интерфейса C++ используется метод getMacLabel:

```
bool DBClientWithCommands::getMacLabel(const string &ns, BSONObj* info = NULL);
```

Аргументы:

– ns — контейнер в виде "БД.коллекция", при этом если указана коллекция, то возвращаются мандатные атрибуты коллекции, в противном случае возвращаются мандатные атрибуты базы данных;

– info — возвращаемые мандатные атрибуты в виде объекта с полями _mac и _ccr, содержащих, соответственно, метку конфиденциальности и признак применения мандатных атрибутов контейнера.

В случае успеха метод возвращает true; если получение мандатных атрибутов невозможно, то метод возвращает false.

Для получения мандатных атрибутов контейнера с помощью прикладного программного интерфейса C используется метод mongo_cmd_get_maclabel:

```
int mongo_cmd_get_maclabel( mongo *conn, const char *db,
                            const char *collection, bson *out );
```

Аргументы:

– conn — указатель на экземпляр установленного соединения с сервером ДОСУБД;

– db — база данных;

– collection — коллекция, при этом если коллекция не указана, возвращаются мандатные атрибуты базы данных;

– out — BSON документ с результатами выполнения запроса (возвращаемые ман-

датные атрибуты расположены в полях _mac и _ccr, содержащих, соответственно, метку конфиденциальности и признак применения мандатных атрибутов контейнера).

В результате вызова функции возвращается соответствующий код ошибки.

9.3.4. Изменение мандатных атрибутов

Для изменения мандатных атрибутов текущей базы данных в командном интерфейсе mongo используется команда db.chmac(label, ccr), где первым аргументом указывается новая метка конфиденциальности в текстовом виде, а вторым — признак применения мандатных атрибутов, например:

```
> db.chmac("{0,0}", true)
true
```

Аналогично могут быть изменены мандатные атрибуты выбранной коллекции текущей базы данных:

```
> db.coll0.chmac("{0,0}", true)
true
```

Для изменения метки конфиденциальности документов используется операция {$chmac: {'_mac':label}} команды update. В качестве аргумента указывается текстовое представление новой метки конфиденциальности:

```
> db.coll0.update({}, {$chmac: {'_mac': "{0,0}"}}, {multi: true})
true
```

ВНИМАНИЕ! Изменение мандатных атрибутов документов операцией $set команды update запрещено.

Для изменения мандатных атрибутов контейнера с помощью прикладного программного интерфейса C++ используется метод chmac:

```
bool DBClientWithCommands::chmac( const string &ns, const string &label,
                                  bool ccr, BSONObj* info = NULL );
```

Аргументы:

– ns — контейнер в виде "БД.коллекция", при этом, если указана коллекция, то изменяются мандатные атрибуты коллекции, в противном случае изменяются мандатные атрибуты базы данных;

– label — устанавливаемая мандатная метка в текстовом виде;

– ccr — устанавливаемый признак применения мандатных атрибутов контейнера;

– info — указатель на объект для возвращения результата выполнения операции и диагностических сообщений об ошибке.

В случае успеха метод возвращает true, иначе — false.

Для изменения мандатных атрибутов контейнера с помощью прикладного программного интерфейса С используется функция mongo_cmd_chmac:

```
int mongo_cmd_chmac( mongo *conn, const char *db, const char *collection,
                     const char *label, bson_bool_t ccr, bson *out );
```

Аргументы:

– conn — указатель на экземпляр установленного соединения с сервером ДОСУБД;

– db — база данных;

– collection — коллекция, при этом если коллекция указана, возвращаются мандатные атрибуты коллекции, в противном случае возвращаются мандатные атрибуты базы данных;

– label — устанавливаемая мандатная метка в текстовом виде;

– ccr — устанавливаемый признак применения мандатных атрибутов контейнера;

– out — BSON-документ с результатами выполнения запроса.

В результате вызова функции возвращается соответствующий код ошибки.

Для изменения метки конфиденциальности документов с помощью прикладного программного интерфейса используются стандартные вызовы для модификации с указанием операции {$chmac: {'_mac':label}} команды update. В качестве аргумента указывается текстовое представление новой метки конфиденциальности.

ВНИМАНИЕ! Санкционированное изменение мандатных атрибутов контейнеров и документов требует особых привилегий пользователя. В случае отсутствия необходимых привилегий или недопустимости указанной метки генерируется соответствующая ошибка.

Примечание. Все действия по санкционированному изменению мандатных атрибутов контейнеров и документов регистрируются в подсистеме регистрации событий.

9.4. Регистрация событий

ДОСУБД обеспечивает регистрацию событий в расширенной подсистеме протоколирования ОС СН с использованием прикладного программного интерфейса подсистемы безопасности PARSEC. При этом регистрируются как попытки доступа к объектам типа «база данных», «коллекция документов» и «документ», так и попытки изменения классификационных меток субъектов и объектов доступа.

В ОС СН реализована расширенная подсистема протоколирования, осуществляющая регистрацию событий в двоичные файлы с использованием сервиса parlogd. Описание указанного сервиса приведено в эксплуатационной документации на ОС СН и в руководстве man parlogd.

В библиотеках подсистемы безопасности PARSEC реализован программный интерфейс для протоколирования событий с использованием расширенной подсистемы прото-

колирования. Описание интерфейса приведено в руководстве `man parsec-aud`.

В соответствии с руководством `man parsec-aud` для подсистемы безопасности PARSEC для для каждого пользователя используются списки успешных (success events mask) и неуспешных (failure events mask) типов запросов на доступ, которые регистрируются в журнале ДОСУБД и подсистеме регистрации событий ОС СН. В случае, когда указанные списки явно не заданы, используются списки типов запросов по умолчанию.

Примечание. Регистрации событий установки соединения пользователя с БД (`CONNECT`) и разъединении с ней (`DISCONNECT`) регистрируются всегда.

Списки типов запросов на доступ реализованы в виде маски. При этом каждому типу запроса соответствует установленный (для регистрируемых запросов) или сброшенный (для нерегистрируемых запросов) элемент маски.

```
//! псевдонимы регистрируемых событий
const char eacAudAliases[33] = "crudsa!!!!!!!!leCRUDSA!!!!!!!!LE";

//! Список регистрируемых событий
enum eacEvent {
    /**
     * @name Регистрация выполнения запросов к данным
     *
     * Эта группа флагов использует самые младшие 4-е бита (первую справа
     * шестнадцатеричную цифру).
     */
    /** @{*/
    //! CREATE
    EAC_EVENT_CREATE = (1 << 0), // c C
    //! READ
    EAC_EVENT_READ = (1 << 1), // r R
    //! UPDATE
    EAC_EVENT_UPDATE = (1 << 2), // u U
    //! DELETE
    EAC_EVENT_DELETE = (1 << 3), // d D
    /** @}*/
    /**
     * @name Регистрация запросов на модификацию прав доступа
     *
     * Эта группа использует следующие 4-ре бита
     */
    /** @{*/
    //! Запрос на модификацию субъектов
    EAC_EVENT_SUBJECT = (1 << 4), // s S
    //! Запрос на модификацию прав доступа к объектам БД
```

```
    EAC_EVENT_RIGHTS = (1 << 5), // a A (access)
    /** @}*/
    /**
      * @name Регистрация использования механизма идентификации//
         аутентификации
      *
      * Для этой группы флагов используются старшие 4 бита
      * Эти флаги включены всегда, вне зависимости от настроек аудита,
      * поэтому при конфигурировании аудита их можно не учитывать.
      */
    /** @{*/
    //! Соединение с базой данных
    EAC_EVENT_CONNECT = (1 << 14), // l L (login)
    //! Расоединение с базой данных
    EAC_EVENT_DISCONNECT = (1 << 15) // e E (exit)
    /** @}*/
};
```

Список регистрируемых событий включает в себя события доступа к документам согласно парадигме CRUD MongoDB:

– **C**reate — создание документов;

– **R**ead — чтение документов;

– **U**pdate — модификация документов;

– **D**elete — удаление документов.

Все операции по манипуляции данными в ДОСУБД приводятся к соответствующим событиям доступа.

Действия по регистрации событий выполняются экземпляром мандатного контекста пользователя в процессе проверки доступа к объекту БД. При этом используются текущие мандатные атрибуты контекста с учетом привилегий пользователя.

При инициализации мандатного контекста для указанного пользователя выполняется установка маски регистрации событий. В дальнейшем, в ходе работы функций мандатного контекста пользователя, по этой маске проверяется необходимость регистрации каждого конкретного события. Указанная маска регистрации событий формируется из маски регистрации событий, заданной для пользователя (см. 9.4.1), и маски регистрации событий текущей базы данных (см. 9.4.2).

Согласно требованиям подсистемы безопасности PARSEC для корректной работы сервиса `parlogd`, для ДОСУБД в конфигурационном файле `events_mongo.conf` задан шаблон для подсистемы регистрации событий PARSEC, описывающий структуру сообщения ДОСУБД.

При установке ДОСУБД указанный файл копируется в соответствующее место кон-

фигурационных файлов подсистемы регистрации событий PARSEC (/etc/parsec/mlog).

9.4.1. Настройка регистрации событий пользователя

Для указания масок аудита пользователя используется поле audit документа привилегий пользователя (PrivilegeDocument) (см. 9.1). Данное поле содержит в текстовом представлении маску регистрации событий с использованием псевдонимов регистрируемых событий eacAudAliases, где символы в нижнем регистре задают список успешных, а в верхнем — неуспешных типов запросов на доступ.

9.4.2. Настройка регистрации событий базы данных

Для настройки регистрации событий текущей базы данных в командном интерфейсе mongo используется команда db.chaud(audit), где аргументом является текстовое представление маски регистрации событий с использованием псевдонимов регистрируемых событий eacAudAliases, где символы в нижнем регистре задают список успешных, а в верхнем — неуспешных типов запросов на доступ, например:

```
> db.chaud("crudsaCRUDSA")
true
```

Для настройки регистрации событий текущей базы данных с помощью прикладного программного интерфейса C++ используется метод chaud:

```
bool DBClientWithCommands::chaud( const string &db, const string &audit,
                                  BSONObj* info = NULL );
```

Аргументы:

– db — база данных;

– audit — текстовое представление маски регистрации событий;

– info — указатель на объект для возвращения результата выполнения операции и диагностических сообщений об ошибке.

В случае успеха метод возвращает true, иначе — false.

Для настройки регистрации событий текущей базы данных с помощью прикладного программного интерфейса C используется функция mongo_cmd_chaud:

```
int mongo_cmd_chaud( mongo *conn, const char *db, const char *audit,
                     bson *out );
```

Аргументы:

– conn — указатель на экземпляр установленного соединения с сервером ДОСУБД;

– db — база данных;

– audit — текстовое представление маски регистрации событий;

– out — BSON-документ с результатами выполнения запроса.

В результате вызова функции возвращается соответствующий код ошибки.

10. ТЕСТИРОВАНИЕ МАНДАТНОГО РАЗГРАНИЧЕНИЯ ДОСТУПА ДОСУБД

ДОСУБД обеспечивает реализацию следующих функций по защите информации от НСД:

– дискреционный принцип контроля доступа;

– мандатный принцип контроля доступа;

– взаимодействие пользователя с комплексом средств защиты (КСЗ);

– идентификацию и аутентификацию пользователей;

– регистрацию событий безопасности информации;

– восстановление данных, обрабатываемых изделием, после сбоев и отказов оборудования;

– автоматизированное тестирование средств мандатного разграничения доступа.

В состав ДОСУБД входит пакет `mongodb-se-test`, содержащий тесты функциональных возможностей по мандатному разграничению доступа.

Тестирование осуществляется выполнением сценариев с запросами, относящимся к тестируемой части функциональности, с помощью интерактивного командного интерфейса доступа к ДОСУБД `mongo`.

Результат выполнения сохраняется в выходном файле и в дальнейшем сравнивается с эталонным файлом результатов выполнения. Решение об успешности прохождения теста принимается по результату сравнения. Тест считается выполненным успешно при отсутствии расхождения результатов с эталоном.

Также в составе тестов представлены тесты мандатного разграничения доступа ДОСУБД, реализованные с помощью прикладного программного интерфейса ДОСУБД на языках C и C++.

При выполнении каждого теста создаются необходимые объекты БД, изменяются правила разграничения доступа и выполняются запросы от пользователей с разными атрибутами безопасности и наборами привилегий. При этом проверяется как успешность выполнения запросов, так и отказы доступа. В тестах проверки мандатного разграничения доступа проверяется доступ пользователей с разными мандатными метками и наборами привилегий к защищенным мандатными метками данным (базам данных, коллекциям и документам). По завершении теста все созданные объекты удаляются.

Функции защиты, такие как взаимодействие пользователя с СЗИ, идентификация и аутентификация и регистрация событий проверяются косвенным образом, т. к. в каждом тесте осуществляется доступ разных пользователей, используется их взаимодействие с СЗИ и регистрируются все попытки доступа к защищаемым объектам, создания и уничтожения объектов и действия по изменению правил разграничения доступа. Также проверя-

ются системные функции ДОСУБД, такие как индексирование, резервное копирование и восстановление, экспорт и импорт.

10.1. Структура тестов

При установке пакета ДОСУБД `mongodb-se-test` в каталог `/usr/share/mongodb/setests/` помещаются необходимые сценарии тестов и эталоны результатов для выполнения тестов.

Каталог `extac` имеет следующую структуру:

– `cTest` — каталог, содержащий тест прикладного программного интерфейса ДОСУБД на языке С;

– `cppTest` — каталог, содержащий тест прикладного программного интерфейса ДОСУБД на языке С++;

– `expected` — каталог, содержащий файлы эталонов результатов теста;

– `tests` — каталог, содержащий сценарии тестов;

– `support` — каталог, содержащий вспомогательные сценарии общих частей тестов. Также в этом каталоге располагается скрипт создания пользователей в ОС с назначением им соответствующих атрибутов безопасности;

– `runtests` — вспомогательный скрипт для запуска процесса тестирования;

– `runsetests` — вспомогательный скрипт для запуска процесса тестирования дополнительных функциональных возможностей.

Описание тестов приведено в таблице 20.

Таблица 20

№ п/п	Тест	Описание
10.1.1	`aggregate`	Проверка мандатного разграничения доступа при агрегации документов
10.1.2	`chmac`	Проверка изменения мандатных атрибутов
10.1.3	`copyto`	Проверка мандатного разграничения доступа при копировании коллекций
10.1.4	`create`	Проверка мандатного разграничения доступа при создании документов
10.1.5	`database`	Проверка мандатного разграничения доступа для баз данных
10.1.6	`delete`	Проверка мандатного разграничения доступа при удалении документов
10.1.7	`dump-restore`	Проверка мандатного разграничения доступа при резервном копировании/восстановлении
10.1.8	`import-export`	Проверка мандатного разграничения доступа при экспорте/импорте документов
10.1.9	`index`	Проверка мандатного разграничения доступа при работе с индексами

Окончание таблицы 20

№ п/п	Тест	Описание
10.1.10	`macid`	Проверка функции получения текущей метки сессии
10.1.11	`namespace`	Проверка мандатного разграничения доступа для коллекций
10.1.12	`read`	Проверка мандатного разграничения доступа при чтении документов
10.1.13	`stats`	Проверка мандатного разграничения доступа при получении статистики
10.1.14	`update`	Проверка мандатного разграничения доступа при модификации документов

10.1.1. aggregate

Проверка мандатного разграничения доступа при агрегации документов заключается в создании коллекций и документов пользователями с разным уровнем доступа с контролем корректности мандатных атрибутов, назначенных при создании объектов доступа. Далее производятся попытки выполнения операций агрегации от имени пользователей с разными мандатными уровнями доступа. Затем выполняется сброс признака применения мандатных атрибутов контейнера `CCR` у коллекций и повторяются попытки выполнения операций агрегации от имени пользователей с разными мандатными уровнями.

При проведении проверок оценивается соответствие результата предоставления доступа диспетчером доступа ДОСУБД (успех или отказ) установленным мандатным атрибутам объектов доступа и мандатным уровням доступа и привилегиям пользователей.

10.1.2. chmac

Проверка функции изменения мандатных атрибутов `chmac` заключается в создании в БД, имеющей определенные мандатные атрибуты и сброшенный признак применения мандатных атрибутов контейнера `CCR`, коллекций, имеющих разные мандатные атрибуты, с последующим созданием в коллекциях документов, имеющих разные мандатные атрибуты, и сброс признака применения мандатных атрибутов контейнера `CCR` у коллекций. Далее производятся следующие проверки:

– проверка невозможности изменения мандатных атрибутов документов непривилегированным пользователем;

– проверка возможности изменения мандатных атрибутов документов привилегированным пользователем (при этом пользователь без привилегии игнорирования мандатного разграничения доступа сможет изменить метку только доступных ему документов);

– проверка невозможности изменения мандатных атрибутов коллекции непривилегированными пользователями;

– проверка изменения мандатных атрибутов коллекции привилегированными поль-

зователями.

При проведении проверок оценивается соответствие результата изменения мандатных атрибутов диспетчером доступа ДОСУБД (успех или отказ) установленным мандатным атрибутам объектов доступа и мандатным уровням доступа и привилегиям пользователей.

10.1.3. copyto

Проверка мандатного разграничения доступа при копировании коллекций заключается в создании коллекций, имеющих разные мандатные атрибуты, с последующим созданием в коллекциях документов, имеющих разные мандатные атрибуты, и сброс признака применения мандатных атрибутов контейнера `CCR` у коллекций. Далее производится копирование коллекций, при котором новые коллекции и документы в них должны приобрести метку создающего их пользователя. При этом скопированы будут только доступные пользователю документы.

При проведении проверок оценивается:

– соответствие результата предоставления доступа диспетчером доступа ДОСУБД (успех или отказ) установленным мандатным атрибутам объектов доступа и мандатным уровням доступа и привилегиям пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов новых коллекций и документов в них и мандатных уровней доступа пользователей.

10.1.4. create

Проверка мандатного разграничения доступа при создании документов заключается в последовательных попытках создания коллекций, имеющих разные мандатные атрибуты, с последующим созданием в коллекциях документов пользователями с разными мандатными уровнями доступа и наборами привилегий.

При проведении проверки оценивается соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов новых коллекций и документов в них и мандатными уровнями доступа и наборами привилегий пользователей.

10.1.5. database

Проверка мандатного разграничения доступа для баз данных заключается в установке для БД определенных мандатных атрибутов и признака применения мандатных атрибутов контейнера `CCR` с последующим созданием в БД коллекций, имеющих разные мандатные атрибуты, и созданием в коллекциях документов, имеющих разные мандатные атрибуты. Также проверяется доступ при выполнении операций с базой данных, таких как: создание, удаление, модификация параметров.

При проведении проверок оценивается:

– соответствие результата предоставления доступа диспетчером доступа ДОСУБД (успех или отказ) установленным мандатным атрибутам объектов доступа и мандатным уровням доступа и привилегиям пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов новых коллекций и документов в них и мандатных уровней доступа пользователей.

10.1.6. delete

Проверка мандатного разграничения доступа при удалении документов заключается в установке для БД определенных мандатных атрибутов и сброса признака применения мандатных атрибутов контейнера `CCR` с последующим созданием в БД коллекций, имеющих разные мандатные атрибуты, и созданием в коллекциях документов, имеющих разные мандатные атрибуты. Далее выполняются последовательные попытки просмотра и удаления пользователями с разными мандатными уровнями доступа и наборами привилегий документов в коллекциях. Затем осуществляется сброс признака применения мандатных атрибутов контейнера `CCR` у коллекций и вновь выполняются последовательные попытки просмотра и удаления пользователями с разными мандатными уровнями доступа и наборами привилегий документов в коллекциях.

При проведении проверок оценивается:

– соответствие результата предоставления диспетчером доступа ДОСУБД (успех или отказ) доступа к документам для просмотра и удаления установленным мандатным атрибутам БД, коллекций, документов в них и мандатных уровней доступа и наборами привилегий пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов новых коллекций и документов в них и мандатных уровней доступа пользователей.

10.1.7. dump-restore

Проверка мандатного разграничения доступа при резервном копировании/восстановлении заключается в создании коллекций и документов пользователями с разным уровнем доступа с контролем корректности мандатных атрибутов, назначенных при создании объектов доступа. После этого производятся попытки выполнения резервного копирования данных с помощью утилиты командой строки `mongodump` от имени пользователей с разными мандатными уровнями доступа и наборами привилегий. В резервные копии должны попасть только документы, доступные пользователям для чтения. После очистки тестовой БД производятся попытки восстановления данных с помощью утилиты командой строки `mongorestore` от имени пользователей с разными мандатными уровнями доступа и на-

борами привилегий. При этом в случае непривилигированного пользователя восстанавливаемые данные должны приобрести метку пользователя, а в случае привилегированного пользователя сохранить свою исходную мандатную метку.

При проведении проверок оценивается:

– соответствие результата предоставления диспетчером доступа ДОСУБД (успех или отказ) доступа к документам для резервирования установленным мандатным атрибутам БД, коллекций, документов в них и мандатных уровней доступа и наборами привилегий пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов восстанавливаемых коллекций и документов в них и мандатных уровней доступа пользователей.

10.1.8. import-export

Проверка мандатного разграничения доступа при экспорте/импорте документов заключается в создании коллекций и документов пользователями с разным уровнем доступа с контролем корректности мандатных атрибутов, назначенных при создании объектов доступа. После этого производятся попытки выполнения экспорта данных с помощью утилиты командой строки `mongoexport` от имени пользователей с разными мандатными уровнями доступа и наборами привилегий. Экспортированы должны быть только документы, доступные пользователям для чтения. После очистки тестовой БД производятся попытки импорта заранее подготовленных данных (как с мандатными атрибутами, так и без них) с помощью утилиты командой строки `mongoimport` от имени пользователей с разными мандатными уровнями доступа и наборами привилегий. При этом в случае непривилегированного пользователя импортируемые данные должны приобрести метку пользователя, а в случае привилегированного пользователя сохранить свою исходную мандатную метку.

При проведении проверок оценивается:

– соответствие результата предоставления диспетчером доступа ДОСУБД (успех или отказ) доступа к документам для экспорта установленным мандатным атрибутам БД, коллекций, документов в них и мандатных уровней доступа и наборами привилегий пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов импортируемых коллекций и документов в них и мандатных уровней доступа пользователей.

10.1.9. index

Проверка мандатного разграничения доступа при использовании индексов заключается в создании коллекций и документов пользователями с разным уровнем доступа с

контролем корректности мандатных атрибутов, назначенных при создании объектов доступа. После создания индексов для коллекций производятся попытки выборки документов по индексу от имени пользователей с разными мандатными уровнями доступа. Затем выполняется сброс признака применения мандатных атрибутов контейнера `CCR` у коллекций и повторяются попытки выборки документов по индексу от имени пользователей с разными мандатными уровнями доступа и наборами привилегий. Также проверяется доступ при выполнении операций с индексами.

При проведении проверок оценивается:

– соответствие результата предоставления диспетчером доступа ДОСУБД (успех или отказ) доступа к документам для выборки установленным мандатным атрибутам БД, коллекций, документов в них и мандатных уровней доступа и наборами привилегий пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов новых коллекций и документов в них и мандатных уровней доступа пользователей.

10.1.10. macid

Проверка функции получения текущей метки сессии `macid` заключается в последовательной установке соединений от имени пользователей с разными мандатными уровнями доступа с последующим выполнением вызова проверяемой функции `macid`.

При проведении проверки оценивается соответствие результата вызова функции получения текущей метки сессии `macid` реальным уровням доступа пользователей.

10.1.11. namespace

Проверка мандатного разграничения доступа для коллекций заключается в установке для БД определенных мандатных атрибутов и признака применения мандатных атрибутов контейнера `CCR`. Далее производятся попытки выполнения операций создания, удаления и переименования коллекций, имеющих разные мандатные атрибуты, пользователями с разными мандатными уровнями доступа и наборами привилегий.

При проведении проверок оценивается:

– соответствие результата предоставления доступа диспетчером доступа ДОСУБД (успех или отказ) установленным мандатным атрибутам объектов доступа и мандатным уровням доступа и привилегиям пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов новых коллекций и документов в них и мандатных уровней доступа пользователей.

10.1.12. read

Проверка мандатного разграничения доступа при выборке документов заключается в создании коллекций и документов пользователями с разным уровнем доступа с контролем корректности мандатных атрибутов, назначенных при создании объектов доступа. Далее производятся попытки выборки документов от имени пользователей с разными мандатными уровнями доступа. Затем выполняется сброс признака применения мандатных атрибутов контейнера `CCR` у коллекций и повторяются попытки выборки документов от имени пользователей с разными мандатными уровнями.

При проведении проверок оценивается:

– соответствие результата предоставления диспетчером доступа ДОСУБД (успех или отказ) доступа к документам для выборки установленным мандатным атрибутам БД, коллекций, документов в них и мандатных уровней доступа и наборами привилегий пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов новых коллекций и документов в них и мандатных уровней доступа пользователей.

10.1.13. stats

Проверка мандатного разграничения доступа при операциях получения статистики заключается в создании коллекций и документов пользователями с разным уровнем доступа с контролем корректности мандатных атрибутов, назначенных при создании объектов доступа. Далее производятся попытки получения статистики коллекций и базы данных от имени пользователей с разными мандатными уровнями доступа. Затем выполняется сброс признака применения мандатных атрибутов контейнера `CCR` у коллекций и повторяются попытки получения статистики коллекций и базы данных от имени пользователей с разными мандатными уровнями.

При проведении проверок оценивается:

– соответствие результата предоставления диспетчером доступа ДОСУБД (успех или отказ) доступа к статистике установленным мандатным атрибутам БД, коллекций, документов в них и мандатных уровней доступа и наборами привилегий пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов новых коллекций и документов в них и мандатных уровней доступа пользователей.

10.1.14. update

Проверка мандатного разграничения доступа при модификации документов заключается в создании коллекций и документов пользователями с разным уровнем доступа с контролем корректности мандатных атрибутов, назначенных при создании объектов доступа. Далее производятся попытки модификации документов от имени пользователей с разными мандатными уровнями доступа. Затем выполняется сброс признака применения мандатных атрибутов контейнера `CCR` у коллекций и повторяются попытки модификации документов от имени пользователей с разными мандатными уровнями.

При проведении проверок оценивается:

– соответствие результата предоставления диспетчером доступа ДОСУБД (успех или отказ) доступа к документам для модификации установленным мандатным атрибутам БД, коллекций, документов в них и мандатных уровней доступа и наборами привилегий пользователей;

– соответствие результата установки диспетчером доступа ДОСУБД мандатных атрибутов новых коллекций и документов в них и мандатных уровней доступа пользователей.

10.2. Проведение тестирования

Для проведения тестирования необходимо выполнить следующие действия:

1) включить компьютер, дождаться загрузки ОС СН;
2) войти в систему как суперпользователь `root`;
3) проверить, используя менеджер пакетов `fly-admin-package`, наличие в системе установленных пакетов ДОСУБД;
4) открыть окно терминала;
5) установить пакет `mongodb-se-test` в соответствии с 3.5;
6) перейти в каталог `/usr/share/mongodb/setests/` командой:

```
cd /usr/share/mongodb/setests/
```

7) запустить тесты командой:

```
./runtests
```

В ходе тестирования будут осуществлены необходимые подготовительные действия и запуск тестов функциональных возможностей по разграничению доступа.

Успешность выполнения каждого теста подтверждается сообщением:

```
успех
```

Проверка считается успешной, если после выполнения программы на экране появится сообщение:

```
Всего = 14, запущено = 14, ошибочных = 0
```

11. СООБЩЕНИЯ ОБ ОШИБКАХ

При возникновении проблем в процессе начальной установки, настройки или функционирования ДОСУБД может выдавать следующие классы ошибок (см. таблицу 21).

Таблица 21

Код ошибки	Сообщение/значение
0	"OK" Успех
1	"InternalError" Внутренняя ошибка
2	"BadValue" Неверное значение
3	"DuplicateKey" Повторяющееся значение ключа
4	"NoSuchKey" Документ с указанным значением ключа не найден
5	"GraphContainsCycle" Граф содержит циклы
6	"HostUnreachable" Хост недоступен
7	"HostNotFound" Хост не найден
8	"UnknownError" Неопределенная ошибка
9	"FailedToParse" Не удалось разобрать строку
10	"CannotMutateObject" Не удалось изменить объект
11	"UserNotFound" Пользователь не найден
12	"UnsupportedFormat" Неподдерживаемый формат данных
13	"Unauthorized" Несанкционированный доступ
14	"TypeMismatch" Несоответствие типа
15	"Overflow" Переполнение
16	"InvalidLength" Неверная длина
17	"ProtocolError" Ошибка протокола обмена

Окончание таблицы 21

Код ошибки	Сообщение/значение
18	"AuthenticationFailed" Ошибка аутентификации
19	"CannotReuseObject" Не удалось повторно использовать объект
20	"IllegalOperation" Недопустимая операция
21	"EmptyArrayOperation" Операция над пустым массивом
22	"InvalidBSON" Неверный бинарный документ BSON
23	"AlreadyInitialized" Ошибка повторной инициализации
24	"LockTimeout" Истекло время ожидания освобождения блокировки
25	"RemoteValidationError" Ошибка несовместимости

Классы ошибок могут быть расширены более детальной информацией о причине возникновения конкретной ошибки (см. таблицу 22).

Таблица 22

Код ошибки	Сообщение/значение
20000	"error refreshing server Kerberos TGT: ..." Не удалось обновить билет Kerberos для сервера
20001	"error obtaining MAC configuration for user ’<username>’: ..." Не удалось получить мандатные атрибуты пользователя ’имя пользователя’
20002	"invalid MAC server side configuration for user ’<username>’: ..." Неверные мандатные атрибуты пользователя ’<имя пользователя>’
20003	"invalid MAC label for user ’<username>’: ..." Метка входящего соединения не соответствует допустимым мандатным атрибутам пользователя ’<имя пользователя>’
20004	"internal MAC init allowed only to ’local’ database" Системная инициализация мандатных атрибутов доступна только при доступе к локальной базе данных
20005	"insufficient privilege or MAC capabilities" Отсутствуют необходимые мандатные привилегии
20006	"invalid MAC label" Недопустимое значение мандатной метки
20006	"invalid audit alias ’<alias>’" Недопустимое значение псевдонима регистрируемого события
20006	"failed to obtain connection MAC label" Не удалось определить метку входящего соединения

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД — база данных
ДОСУБД — документо-ориентированная система управления базами данных
ЕПП — единое пространство пользователей
ОС СН — операционная система специального назначения
ПРД — правила разграничения доступа
РД — руководящий документ
СВТ — средства вычислительной техники
СЗИ — средства защиты информации
СУБД — система управления базами данных
ФС — файловая система
ФСТЭК — Федеральная служба по техническому и экспортному контролю

ACL — Access Control List (список контроля доступа)
ALD — Astra Linux Directory (единое пространство пользователей)
HTTP — HyperText Transfer Protocol (протокол передачи гипертекстовых файлов)
LDAP — Lightweight Directory Access Protocol (легковесный протокол доступа к сервисам каталогов)

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					