

# vinum(8)

## НАЗВАНИЕ

**vinum** - программа управления диспетчером логических томов (Logical Volume Manager)

## СИНТАКСИС

`vinum [команда] [-опции]`

## КОМАНДЫ

**attach** *сплетение том* [**rename**]

**attach** *поддиск сплетение* [**смещение**] [**rename**]

Подключает *сплетение* (plex) к *тому* (volume) или *поддиск* (subdisk) к сплетению.

**checkparity** *сплетение* [-f] [-v]

Проверяет блоки четности в **сплетении** RAID-4 или RAID-5.

**concat** [-f] [-n *имя*] [-v] *диски*

Создает конкатенированный том из указанных **дисков**.

**create** [-f] *файл\_описания*

Создает том как указано в **файле описания**.

**debug**

Вызывает вход диспетчера томов в режим отладчика ядра.

**debug** *флаги*

Устанавливает флаги отладки.

**detach** [-f] [*сплетение* | *поддиск*]

Отключает сплетение или поддиск от тома или сплетения, к которому он подключен.

**dumpconfig** [*диск ...*]

Выдает информацию о конфигурации, хранящуюся на указанных дисках или на всех дисках в системе, если имена дисков не указаны.

**info** [-v] [-V]

Выдает информацию о состоянии диспетчера томов.

**init** [-S *размер*] [-w] *сплетение* | *поддиск*

Инициализирует нулями содержимое **поддиска** или всех поддисков **сплетения**.

**label** *том*

Создает метку тома.

**l** | **list** [-r] [-s] [-v] [-V] [*том* | *сплетение* | *поддиск*]

Выдает информацию об указанных объектах.

**ld** [-r] [-s] [-v] [-V] [*том*]

Выдает информацию о дисках.

**ls** [-r] [-s] [-v] [-V] [*поддиск*]

Выдает информацию о **поддисках**.

**lp [-r] [-s] [-v] [-V] [сплетение]**

Выдает информацию о **сплетениях**.

**lv [-r] [-s] [-v] [-V] [том]**

Выдает информацию о **томах**.

**makedev**

Пересоздает файлы устройств в каталоге **/dev/vinum**.

**mirror [-f] [-n имя] [-s] [-v] диски**

Создает зеркальный том из указанных **дисков**.

**move | mv -f диск объект ...**

Переносит объекты на указанный диск.

**printconfig [файл]**

Записывает копию текущей конфигурации в **файл**.

**quit**

Завершает работу утилиты **vinum** в интерактивном режиме. Обычно это происходит при вводе символа конца файла (EOF).

**read диск ...**

Читает конфигурацию **vinum** с указанных **дисков**.

**rename [-r] [диск | поддиск | сплетение | том] новое\_имя**

Переименовывает указанный объект.

**rebuildparity сплетение [-f] [-v] [-V]**

Пересоздает блоки четности сплетения типа RAID-4 или RAID-5.

**resetconfig**

Сбрасывает полную конфигурацию **vinum**.

**resetstats [-r] [том | сплетение | поддиск]**

Сбрасывает статистические счетчики для указанных объектов или для всех объектов, если ни один не указан.

**rm [-f] [-r] том | сплетение | поддиск**

Удаляет объект.

**saveconfig**

Сохраняет конфигурацию **vinum** на диск после сбоя конфигурации.

**setdaemon [значение]**

Устанавливает конфигурацию демона.

**setstate состояние [том | сплетение | поддиск | диск]**

Устанавливает **состояние**, не влияя на другие объекты, только в диагностических целях.

**start**

Читает конфигурацию со всех дисков **vinum**.

**start [-i интервал [-S размер] [-w] том | сплетение | поддиск]**

Позволяет системе обращаться к объектам

**stop [-f] [том | сплетение | поддиск]**

Прекращает доступ к объектам или останавливает **vinum**, если параметры не указаны.

**stripe [-f] [-n имя] [-v] диски**

Создает том с чередованием из указанных **дисков**.

## ОПИСАНИЕ

Утилита **vinum** взаимодействует с компонентом ядра диспетчера логических томов Vinum. Утилита предназначена либо для интерактивного использования, если запущена без аргументов командной строки, либо для выполнения одной переданной в командной строке команды. В интерактивном режиме утилита **vinum** сохраняет хронологию выполнения команд.

## ОПЦИИ

После команд **vinum** могут указываться необязательные опции. Любую из следующих опций можно указывать для любой команды, но в некоторых случаях опции игнорируются. Например, команда **stop** игнорирует опции **-v** и **-V**.

### **-f**

Опция **-f** ("force") отменяет проверки защиты. Эту опцию надо использовать крайне осторожно. Эта опция предназначена только для критических случаев. Например, команда `rm -f myvolume` удаляет том **myvolume**, даже если он открыт. Любой последующий доступ к этому тому почти наверняка вызовет панику в системе.

### **-i** миллисекунд

При выполнении команд **init** и **start**, ожидать указанное количество миллисекунд между копированиями каждого блока. Это уменьшает нагрузку на систему.

### **-n** имя

Используйте опцию **-n** для указания имени тома в простых командах конфигурации **concat**, **mirror** и **stripe**.

### **-r**

Опция **-r** ("recursive") используется в командах **list** для выдачи информации не только об указанных объектах, но и о входящих в них объектах. Например, при указании в команде **lv**, опция **-r** позволит также получить информацию о сплетениях и поддисках, принадлежащих тому.

### **-s**

Опция **-s** ("statistics") используется в командах **list** для выдачи статистической информации. В команде **mirror** эта опция используется для указания того, что необходимо создавать сплетения с чередованием.

### **-S** размер

Опция **-S** задает размер (transfer size) для команд **init** и **start**.

### **-v**

Опция **-v** ("verbose") может использоваться для запроса более подробной информации.

### **-V**

Опция **-V** ("Very verbose") может использоваться для запроса еще более подробной информации, чем выдает опция **-v**.

### **-w**

Опция **-w** ("wait") требует от утилиты **vinum** дожидаться завершения команд, которые обычно работают в фоновом режиме, например, **init**.

## ДЕТАЛЬНОЕ ОПИСАНИЕ КОМАНД

Команды **vinum** выполняют следующие функции:

### **attach** сплетение том [rename]

#### **attach** поддиск сплетение [смещение] [rename]

Команда **vinum attach** вставляет указанное сплетение или поддиск в том или сплетение, соответственно. В случае поддиска, можно указать **смещение** в сплетении. Если смещение не задано, поддиск будет добавлен в первое подходящее место. После подключения сплетения к непустому тому, утилита **vinum** реинтегрирует сплетение.

Если указано ключевое слово **rename**, утилита **vinum** переименовывает объект (и, в случае сплетения, любые входящие в него поддиски) в соответствии со стандартными соглашениями по именованию **vinum**. Для переименования объекта с присвоением нестандартного имени используйте команду **rename**.

При подключении поддисков следует учитывать несколько соображений:

- Поддиски обычно можно подключать только к конкатенированным сплетениям.
- Если в сплетении RAID-5 с чередованием не хватает поддиска (например, после сбоя диска), его следует заменять только поддиском такого же размера.
- Для добавления дополнительных поддисков к сплетению с чередованием или типа RAID-5, используйте опцию **-f**. В результате, данные в сплетении будут повреждены.
- Для конкатенированных сплетений параметр **смещение** задает смещение в блоках от начала сплетения. Для сплетений типа RAID-5 с чередованием он задает смещение первого блока поддиска: другими словами, смещение представляет собой числовую позицию поддиска, умноженную на размер полосы. Например, в сплетении с размером полосы 271 Кбайт первый поддиск будет иметь смещение 0, второй - смещение 271 Кбайт, третий - 542 Кбайта и т.д. В этом вычислении блоки четности в сплетениях типа RAID-5 игнорируются.

#### **checkparity** сплетение [-f] [-v]

Проверяет блоки четности в указанном сплетении типа RAID-4 или RAID-5. Эта операция поддерживает указатель в сплетении, так что проверку, при необходимости, можно остановить и затем возобновить с той же позиции. Кроме того, этот указатель используется командой **rebuildparity**, поэтому пересоздание блоков четности нужно будет начинать только с места, где выявлена первая проблема четности. Если указан флаг **-f**, команда **checkparity** начинает проверку с начала сплетения. Если указан флаг **-v**, **checkparity** выдает отчет о работе процесса.

#### **concat** [-f] [-n имя] [-v] диски

Команда **concat** предлагает упрощенную альтернативу команде **create** при создании томов из одного конкатенированного сплетения. Наибольший фрагмент свободного места на каждом диске используется для создания поддисков, составляющих сплетение.

Обычно команда **concat** устанавливает для тома и его компонентов произвольные имена. Имя состоит из слова "**vinum**" и небольшого целого числа, например, "**vinum3**". Можно задать **имя** для тома явно с помощью опции **-n имя**. Имена сплетений и поддисков строятся стандартно на базе имени соответствующего тома.

Для дисков имена задавать нельзя. Если диски были проинициализированы как диски **vinum**, имя остается. В противном случае, диски получают имена, состоящие из слова "**vinumdrive**" и небольшого целого числа, например, "**vinumdrive7**". Как и для команды **create**, опцию **-f** можно использовать для переопределения ранее данного имени. Опция **-v** используется для получения подробных результатов.

Примеры использования этой команды см. в разделе "[УПРОЩЕННОЕ КОНФИГУРИРОВАНИЕ](#)" ниже.

#### **create [-f] файл\_описания**

Команда **vinum create** используется для создания любого объекта. С учетом достаточно сложных взаимосвязей и потенциальных опасностей при создании объекта **vinum**, в интерактивном интерфейсе эта функция недоступна. Если имя файла не указано, команда **vinum** открывает временный файл в редакторе. Если установлена переменная среды **EDITOR**, утилита **vinum** запускает указанный в ней редактор. Если же нет, запускается стандартный редактор **vi**. Подробнее о формате **файла описания** см. в разделе "[ФАЙЛ КОНФИГУРАЦИИ](#)" ниже.

Учтите, что функция **vinum create** - аддитивна: если выполнить ее несколько раз, будет создано несколько экземпляров всех неименованных объектов.

Обычно команда **create** не будет изменять имена существующих дисков **vinum**, чтобы случайно не стереть их. Корректный способ удалить не нужные более диски **vinum** - сбросить конфигурацию с помощью команды **resetconfig**. В некоторых случаях, однако, может потребоваться создать новые данные на дисках **vinum**, которые не удастся запустить. В этом случае используйте команду **create -f**.

#### **debug**

Команда **vinum debug** без параметров используется для входа в удаленный отладчик ядра (remote kernel debugger). Она срабатывает только если **vinum** собран с опцией **VINUMDEBUG**. Эта опция остановит работу операционной системы, пока не завершится работа отладчика ядра. Если установлена удаленная отладка, а удаленное подключение к отладчику ядра организовать не получается, для выхода из режима отладки придется перезагрузить систему.

#### **debug флаги**

Устанавливает битовую маску внутренних флагов отладки. Эти флаги могут меняться без уведомления по ходу развития продукта; уточните их значения в заголовочном файле `<sys/dev/vinumvar.h>`. Битовая маска строится из следующих значений:

##### **DEBUG\_ADDRESSES (1)**

Показывать информацию буфера в ходе запросов.

##### **DEBUG\_RESID (4)**

Перейти в отладчик в **complete\_rqe()**.

##### **DEBUG\_LASTREQS (8)**

Поддерживать циклический буфер последних запросов.

##### **DEBUG\_REVIVECONFLICT (16)**

Выдавать информацию о *конфликтах восстановления* (revive conflicts).

##### **DEBUG\_EOFINFO (32)**

Выдавать информацию о внутреннем состоянии при возврате символа конца файла (EOF) со сплетения с чередованием (striped plex).

##### **DEBUG\_MEMFREE (64)**

Поддерживать циклический список последних освобожденных подсистемой выделения памяти (memory allocator) областей памяти.

##### **DEBUG\_REMOTEGDB (256)**

Перейти в удаленный отладчик **gdb** при выполнении команды **debug**.

##### **DEBUG\_WARNINGS (512)**

Выдавать ряд предупреждений о несущественных проблемах в реализации.

#### **detach [-f] сплетение**

#### **detach [-f] поддиск**

Команда **vinum detach** удаляет (отключает) указанное **сплетение** или **поддиск** из тома или сплетения, к которому он подключен. Если удаление объекта приводит к нарушению целостности данных тома, операция не работает, если только не указана опция **-f**. Если имя объекта строится на

базе объекта, в который он входит (например, поддиск **vol1.p7.s0**, подключенный к сплетению **vol1.p7**), имя будет изменено путем добавления префикса "ex-" (например, **ex-vol1.p7.s0**). При необходимости, имя в процессе будет усечено.

Команда **detach** не уменьшает количество поддисков в сплетении с чередованием или типа RAID-5. Вместо этого поддиск помечается как отсутствующий и может быть в дальнейшем заменен с помощью команды **attach**.

### dumpconfig [диск ...]

Команда **vinum dumpconfig** выдает информацию о конфигурации, хранящуюся на указанных дисках. Если имена дисков не указаны, команда **dumpconfig** просматривает все диски в системе в поисках разделов Vinum, и сбрасывает находящуюся на них информацию. Если изменения конфигурации отключены, может оказаться, что эта информация не совпадает с выдаваемой командой **list**. Эта команда используется, в основном, для сопровождения и отладки.

### info

Команда **vinum info** выдает информацию об использовании памяти драйвером **vinum**. Она предназначена, в основном, для отладки. При указании опции **-v** она будет выдавать детальную информацию об используемых областях памяти.

При указании опции **-V** команда **info** выдает информацию о последних (до 64) запросах ввода-вывода, обработанных драйвером **vinum**. Эта информация собирается, только если установлен флаг отладки 8. Информация выдается в следующем формате:

```
vinum -> info -V
Flags: 0x200      1 opens
Total of 38 blocks malloced, total memory: 16460
Maximum allocs:   56, malloc table at 0xf0f72dbc
```

Time	Bytes	SD	SDoff	Event	Do	Off	Buf	Dev	Offset
					ffset	ffset			
14:40:00.637758	1VS	Write	0xf2361f40	91.3	0x10				16384
14:40:00.639280	2LR	Write	0xf2361f40	91.3	0x10				16384
14:40:00.639294	3RQ	Read	0xf2361f40	4.39	0x104109			8192	19
0			0						
14:40:00.639455	3RQ	Read	0xf2361f40	4.23	0xd2109			8192	17
0			0						
14:40:00.639529	3RQ	Read	0xf2361f40	4.15	0x6e109			8192	16
0			0						
14:40:00.652978	4DN	Read	0xf2361f40	4.39	0x104109			8192	19
0			0						
14:40:00.667040	4DN	Read	0xf2361f40	4.15	0x6e109			8192	16
0			0						
14:40:00.668556	4DN	Read	0xf2361f40	4.23	0xd2109			8192	17
0			0						
14:40:00.669777	6RP	Write	0xf2361f40	4.39	0x104109			8192	19
0			0						
14:40:00.685547	4DN	Write	0xf2361f40	4.39	0x104109			8192	19
0			0						
11:11:14.975184	Lock		0xc2374210	2		0x1f8001			
11:11:15.018400	7VS	Write	0xc2374210			0x7c0		32768	10
11:11:15.018456	8LR	Write	0xc2374210	13.39	0xcc0c9			32768	
11:11:15.046229	Unlock		0xc2374210	2		0x1f8001			

Поле **Buf** всегда содержит адрес заголовка пользовательского буфера. Его можно использовать для идентификации запроса, связанного с пользовательским запросом, хотя это и не на 100% надежно: теоретически, два последовательных запроса могут использовать один и тот же заголовок буфера, хотя это и нетипично. Начало запроса идентифицируется значениями **1VS** или **7VS** в поле **Event**. Первый представленный выше пример показывает запросы драйвера, связанные с выполнением запроса пользователя. Второй пример - запрос ввода-вывода поддиска с блокировкой.

Поле **Event** содержит информацию, связанную с последовательностью событий в цепочке выполнения запроса (request chain). Цифры от 1 до 6 показывают приблизительную последовательность событий, а двухбуквенное сокращение описывает действие:

**1VS**

(**vinumstrategy**) показывает информацию о пользовательском запросе при входе в **vinumstrategy()**. Номер устройства определяет устройство **vinum**, а смещение и длина - это пользовательские параметры. С такой записи всегда начинается последовательность выполнения запроса.

## 2LR

(**launch\_requests**) показывает пользовательский запрос непосредственно перед выполнением низкоуровневых запросов **vinum** в функции **launch\_requests()**. Параметры должны быть такими же, как и в записи **1VS**.

В следующих запросах **Dev** - номер устройства соответствующего раздела диска, **Offset** - смещение от начала раздела, **SD** - индекс поддиска в **vinum\_conf**, **SDoff** - смещение от начала поддиска, **Doffset** - смещение соответствующего запроса данных, а **Goffset** - смещение соответствующего запроса группы, если оно необходимо.

## 3RQ

(**request**) показывает один из, возможно, нескольких низкоуровневых запросов **vinum**, выполненных для удовлетворения запроса верхнего уровня. Эта информация также регистрируется в **launch\_requests()**.

## 4DN

(**done**) вызывается из **complete\_rq()**, показывая завершение запроса. Это завершение должно соответствовать запросу, инициированному либо на стадии **4DN** из функции **launch\_requests()**, либо из функции **complete\_raid5\_write()** на стадии **5RD** или **6RP**.

## 5RD

(**RAID-5 data**) вызывается из функции **complete\_raid5\_write()** и представляет данные, записанные на полосу данных массива RAID-5 после вычисления четности.

## 6RP

(**RAID-5 parity**) вызывается из функции **complete\_raid5\_write()** и представляет данные, записанные на *полосу четности* (parity stripe) массива RAID-5 после вычисления четности.

## 7VS

Показывает запрос ввода/вывода для поддиска. Эти запросы, обычно, являются внутренними для драйвера **vinum** и используются для операций типа инициализации или пересоздания сплетений.

## 8LR

Показывает низкоуровневую операцию, сгенерированную для запроса ввода/вывода для поддиска.

## Lockwait

Указывает, что процесс ждет *блокировки диапазона* (range lock). Параметрами являются заголовок буфера, соответствующего запросу, номер сплетения и номер блока. Из внутренних соображений номер блока - на единицу больше, чем адрес начала полосы.

## Lock

Указывает, что была получена блокировка диапазона. Параметры такие же, как и для ожидания блокировки диапазона.

## Unlock

Указывает, что блокировка диапазона была снята. Параметры такие же, как и для ожидания блокировки диапазона.

## init [-S размер] [-w] сплетение | поддиск

Команда **vinum init** инициализирует поддиск путем записи на него нулей. Можно инициализировать все поддиски в сплетении, указав имя сплетения. Это - единственный способ гарантировать согласованность данных в сплетении. Эту инициализацию необходимо выполнять до начала использования сплетения типа RAID-5. Ее также рекомендуется выполнять и для других новых сплетений. Утилита **vinum** инициализирует все поддиски сплетения параллельно. Поскольку это действие может потребовать много времени, оно обычно выполняется в фоновом режиме. Если необходимо подождать завершения команды, используйте опцию **-w** (wait).

Укажите опцию **-S**, если необходимо записывать блоками нестандартного (отличающегося от 16 Кбайт) размера. Утилита **vinum** выдает сообщение на консоль по завершении инициализации.

## label том

Команда **label** записывает на том метку тома в стиле **ufs**. Это - простая альтернатива соответствующему вызову утилиты **disklabel**. Ее выполнение необходимо потому, что некоторые команды **ufs** по-прежнему читают диск в поисках метки вместо того, чтобы выполнять соответствующий вызов **ioctl(2)** для доступа к ней. Драйвер **vinum** хранит метку тома отдельно от данных тома, так что эта команда не нужна для **newfs(8)**. Использовать эту команду не рекомендуется.

**list [-r] [-V] [том | сплетение | поддиск]**

**l [-r] [-V] [том | сплетение | поддиск]**

**ld [-r] [-s] [-v] [-V] [том]**

**ls [-r] [-s] [-v] [-V] [поддиск]**

**lp [-r] [-s] [-v] [-V] [сплетение]**

**lv [-r] [-s] [-v] [-V] [том]**

Команда **list** используется для получения информации об указанном объекте. Если аргумент не указан, выдается информация о всех объектах, известных драйверу **vinum**. Команда **l** - синоним для **list**.

Опция **-r** относится к томам и сплетениям: если она указана, то рекурсивно выдает информацию для поддисков и (для тома) сплетений, составляющих объекты. Команды **lv**, **lp**, **ls** и **ld** выдают только тома, сплетения, поддиски и диски, соответственно. Они особенно полезны при вызове без параметров.

Опция **-s** вызывает выдачу утилитой **vinum** статистической информации по устройствам, опция **-v** (verbose) приводит к выдаче определенной дополнительной информации, а опция **-V** обеспечивает выдачу существенного объема дополнительной информации.

## makedev

Команда **makedev** удаляет каталог **/dev/vinum** и пересоздает его вместе со специальными файлами устройств, отражающими текущую конфигурацию. Эта команда не предназначена для общего использования и должна использоваться только в критических ситуациях.

**mirror [-f] [-n имя] [-s] [-v] диски**

Команда **mirror** является упрощенной альтернативой команде **create** при создании зеркалированных томов. При вызове без опций она создает том типа RAID-1 (зеркалированный) из двух конкатенированных сплетений. Самые большие фрагменты свободного места на каждом диске используются для создания поддисков для сплетений. Первое сплетение строится из **дисков** в списке с нечетными номерами, а второе - из четных. Если диски - разного размера, то и сплетения получатся разного размера.

Если указана опция **-s**, команда **mirror** строит сплетения с чередованием, с размером полосы 279 Кбайт. Размер поддисков в каждом сплетении соответствует размеру наименьшего фрагмента подряд идущего свободного места на всех дисках, формирующих сплетение. И в этом случае размеры сплетений могут отличаться.

Обычно команда **mirror** автоматически создает имя для тома и его компонентов. Это имя состоит из слова "**vinum**" и небольшого целого числа, например, "**vinum3**". Это можно изменить с помощью опции **-n имя**, которая присваивает указанное **имя** тому. Имена сплетений и поддисков строятся на основе имени тома стандартным образом.

Имена дисков выбирать нельзя. Если диски уже были проинициализированы как диски **vinum**, их имена неизменны. В противном случае, диски получают имена, состоящие из слова "**vinumdrive**" и небольшого целого числа, например, "**vinumdrive7**". Как и в случае команды **create**, опция **-f** может использоваться для указания того, что прежнее имя надо изменить. Опция **-v** используется для выдачи более детальной информации.



Ряд примеров использования этой команды представлен в разделе "**УПРОЩЕННОЕ КОНФИГУРИРОВАНИЕ**" ниже.

**mv -f диск объект ...**

**move -f диск объект ...**

Переносит все поддиски указанных **объектов** на новый **диск**. Объектами могут быть поддиски, диски или сплетения. Если указаны диски или сплетения, переносятся все поддиски, связанные с этими объектами.

Опция **-f** для этой функции необходима, поскольку в настоящее время она не сохраняет данные на поддиске. Эта возможность будет добавлена в дальнейшем. Но и в нынешнем виде эта команда подходит для восстановления после сбоя диска.

**printconfig [файл]**

Записывает копию текущей конфигурации в файл в формате, который можно использовать для воссоздания конфигурации **vinum**. В отличие от хранящейся на диске конфигурации, она включает определения дисков. Если **файл** не указан, утилита **vinum** выдает информацию в стандартный выходной поток.

**quit**

Завершает работу утилиты **vinum** в интерактивном режиме. Обычно для этого достаточно ввести символ конца файла (EOF).

**read диск ...**

Команда **read** сканирует указанные диски в поисках разделов **vinum**, содержащих ранее созданную информацию о конфигурации. Она читает информацию о конфигурации, начиная с последних изменений и до самых давних. Утилита **vinum** поддерживает актуальную копию всей конфигурационной информации на каждом разделе диска. Необходимо указать все *секции* (slices) в конфигурации в качестве параметра этой команды.

Команда **read** предназначена для избирательной загрузки конфигурации **vinum** в системе, где есть другие разделы **vinum**. Если необходимо запустить все разделы в системе, проще использовать команду **start**.

Если по ходу выполнения этой команды утилитой **vinum** возникают любые ошибки, она отключает автоматическое обновление конфигурации, чтобы предотвратить повреждение копий на диске. Это также происходит, если в конфигурации на диске обнаруживается ошибка (например, поддиск с недопустимой спецификацией занимаемого места). Изменения можно возобновить с помощью команд **setdaemon** и **saveconfig**. Сбросьте бит 2 (числовое значение - 4) в маске опций демона, чтобы снова можно было сохранять изменения конфигурации.

**rebuildparity сплетение [-f] [-v] [-V]**

Пересоздает блоки четности для указанного сплетения типа RAID-4 или RAID-5. Это действие поддерживает указатель на сплетение, так что его можно остановить и в дальнейшем перезапустить с того же места, при необходимости. Кроме того, этот указатель используется командой **checkparity**, поэтому пересоздание блоков четности можно будет начать с того места, где выявлена первая проблема четности.

Если указан флаг **-f**, команда **rebuildparity** начинает пересоздание с начала сплетения. Если указан флаг **-v**, команда **rebuildparity** сначала проверяет существующие блоки четности и выдает информацию о тех из них, которые оказались некорректными до пересоздания. Если указан флаг **-V**, команда **rebuildparity** выдает отчет о ходе работы.

**rename [-r] [диск | поддиск | сплетение | том] новое\_имя**

Изменяет имя указанного объекта. Если указана опция **-r**, подчиненные объекты будут переименовываться по стандартным правилам: имена сплетений будут строиться путем добавления **номер\_сплетения** к имени тома, а имена поддисков - путем добавления **номер\_поддиска** к имени сплетения.

**resetconfig**

Команда **resetconfig** полностью сбрасывает конфигурацию **vinum** в системе. Используйте эту команду только при необходимости полностью удалить конфигурацию. Утилита **vinum** попросит подтверждения; необходимо ввести слова **NO FUTURE** как показано ниже:

```
# vinum resetconfig
```

```
WARNING! This command will completely wipe out your vinum
configuration. All data will be lost. If you really want
to do this, enter the text
```

```
NO FUTURE
Enter text -> NO FUTURE
Vinum configuration obliterated
```

Как и сказано в сообщении, эта команда - жест отчаяния. Используйте ее, только если хотите никогда больше не видеть существующую конфигурацию.

#### **resetstats [-r] [том | сплетение | поддиск]**

Драйвер **vinum** поддерживает ряд статистических счетчиков для каждого объекта. Подробнее см. в заголовочном файле `<sys/dev/vinumvar.h>`. Команда **resetstats** используется для сброса этих счетчиков. При указании опции **-r**, **vinum** также сбрасывает счетчики подчиненных объектов.

#### **rm [-f] [-r] том | сплетение | поддиск**

Команда **rm** удаляет объект из конфигурации **vinum**. После удаления объекта восстановить его нельзя. Обычно перед удалением объекта утилита **vinum** выполняет множество проверок согласованности. Опция **-f** требует от **vinum** пропустить эти проверки и удалить объект в любом случае. Используйте эту опцию очень осторожно: она может привести к полной потере данных тома.

Обычно команда **vinum** отказывается удалять том или сплетение, если у него есть подчиненные сплетения или поддиски, соответственно. Можно потребовать от **vinum** безусловного удаления объекта с помощью опции **-f** или просто удалить все подчиненные объекты тоже с помощью опции **-r** (recursive). При удалении тома с опцией **-r** будут удалены как сплетения, так и поддиски, принадлежащие сплетениям.

#### **saveconfig**

Сохраняет текущую конфигурацию на диск. Обычно это не нужно, поскольку утилита **vinum** автоматически сохраняет любые изменения в конфигурации. Если при запуске возникает ошибка, обновления будут отключены. При их повторном включении с помощью команды **setdaemon**, **vinum** не сохраняет автоматически конфигурацию на диск. Используйте эту команду для сохранения конфигурации.

#### **setdaemon [значение]**

Команда **setdaemon** устанавливает переменную битовую маску для демона **vinum**. Эта команда - временная и будет заменена. В настоящее время битовая маска может содержать биты 1 (регистировать каждое действие с помощью **syslog**) и 4 (не обновлять конфигурацию). Необязательный бит 4 может пригодиться при восстановлении после ошибки.

#### **setstate состояние [том | сплетение | поддиск | диск]**

Команда **setstate** устанавливает заданное **состояние** для указанных объектов. При этом не используется обычный механизм проверки согласованности **vinum**, так что эту команду следует использовать только для восстановления. Неправильное использование этой команды может вызвать сбой системы.

#### **start [-i интервал] [-S размер] [-w] [сплетение | поддиск]**

Команда **start** запускает (переводит в работающее состояние) один или несколько объектов **vinum**.

Если имена объектов не указаны, утилита **vinum** сканирует все известные системе диски в поисках дисков **vinum**, а затем читает конфигурацию так, как было описано для команды **read**. Диск **vinum** (vinum drive) содержит заголовок со всей информацией о данных, хранящихся на диске, включая имена других дисков, которые необходимы для представления сплетений и томов.

Если по ходу выполнения этой команды утилитой **vinum** будут обнаружены ошибки, она отключит автоматическое обновление конфигурации, чтобы избежать повреждения копий на диске. То же самое произойдет, если конфигурация на диске содержит ошибку (например, поддиски с

недопустимой спецификацией занимаемого места). Изменения можно возобновить с помощью команд **setdaemon** и **saveconfig**. Сбросьте бит 2 (числовое значение - 4) в маске опций демона, чтобы снова можно было сохранять изменения конфигурации.

Если имена объектов указаны, утилита **vinum** запускает их. Обычно эту операцию имеет смысл выполнять только для поддисков. Действие зависит от текущего состояния объекта:

- Если объект уже работает, утилита **vinum** ничего не делает.
- Если объект является поддиском в состояниях **down** или **reborn**, утилита **vinum** запускает его (меняет состояние на **up**).
- Если объект является поддиском в состоянии **empty** (пуст), изменение зависит от поддиска. Если он входит в сплетение, являющееся частью тома, в котором есть и другие сплетения, утилита **vinum** переводит поддиск в состояние **reviving** и пытается скопировать данные с тома. По завершении этой операции поддиск запускается. Если же поддиск входит в сплетение, являющееся частью тома, в котором других сплетений нет, или если он вообще не входит в сплетение, утилита **vinum** запускает его немедленно.
- Если объект является поддиском в состоянии **reviving**, утилита **vinum** продолжает его "оживление" в отключенном режиме. По завершении этой операции поддиск запускается.

Когда поддиск запускается (переводится в состояние **up**), утилита **vinum** автоматически проверяет состояние сплетения и тома, к которым он может принадлежать и изменяет их состояние при необходимости.

Если объект является сплетением, команда **start** проверяет состояние подчиненных поддисков (и сплетений, в случае тома) и запускает все поддиски, которые можно запустить.

Для запуска сплетения в томе из нескольких сплетений данные должны быть скопированы с другого сплетения в томе. Поскольку зачастую для этого требуется много времени, это обычно делается в фоновом режиме. Если необходимо дождаться завершения этой операции (например, если она выполняется в сценарии), используйте опцию **-w**.

Копирование данных не только требует много времени, но и может существенно нагружать систему. Можно указать размер передаваемого блока в байтах или секторах с помощью опции **-S**, и интервал (в миллисекундах) ожидания между копированиями каждого блока с помощью опции **-i**. Обе эти опции позволяют снизить нагрузку на систему.

#### **stop [-f] [том | сплетение | поддиск]**

Если параметры не указаны, команда **stop** удаляет демон ядра **vinum** и останавливает **vinum(4)**. Это можно сделать, только если ни один из объектов не активен. В частности, опция **-f** не позволяет обойти это требование. Обычно команда **stop** записывает текущую конфигурацию обратно на диски перед завершением работы. Это будет невозможно, если изменения конфигурации отключены, поэтому **vinum** не остановится, если изменения конфигурации отключены. Это можно обойти, указав опцию **-f**.

Команда **stop** может работать только если **vinum** был загружен как демон ядра (**kld**), поскольку статически сконфигурированных драйвер выгрузить нельзя. Команда **vinum stop** не работает, если **vinum** сконфигурирован статически.

Если имена объектов указаны, команда **stop** отключает доступ к объектам. Если объекты имеют подчиненные объекты, эти подчиненные объекты должны быть либо не активны (остановлены или в состоянии ошибки), либо надо указывать опции **-r** и **-f**. Эта команда не удаляет объекты из конфигурации. К ним можно будет снова получить доступ после команды **start**.

По умолчанию утилита **vinum** не останавливает активные объекты. Например, нельзя остановить сплетение, подключенное к активному тому, и нельзя отключить открытый том. Опция **-f** требует от **vinum** пропустить эту проверку и удалить объект в любом случае. Используйте эту опцию максимально осторожно, и только помня о последствиях: при некорректном использовании она может привести к серьезному повреждению данных.

### **stripe [-f] [-n имя] [-v] диски**

Команда **stripe** является упрощенной альтернативой команде **create** для создания томов из единственного сплетения с чередованием. Размер поддисков определяется как размер максимального фрагмента свободного места, доступного на всех указанных дисках. Размер полосы фиксирован - 279 Кбайт.

Обычно команда **stripe** создает для тома и его компонентов стандартные имена. Имя состоит из слова "**vinum**" и небольшого целого числа, например, "**vinum3**". Можно задать нестандартное имя с помощью опции **-n имя**, что присваивает указанное **имя** тому. Имена сплетений и поддисков строятся на основе имени тома стандартным образом.

Имена для дисков задавать нельзя. Если диски уже были проинициализированы как диски **vinum**, имя остается. В противном случае, диски получают имена, начинающиеся со слова "**vinumdrive**", и заканчивающиеся небольшим целым числом, например, "**vinumdrive7**". Как и в случае команды **create**, можно использовать опцию **-f** для изменения прежнего имени. Опция **-v** используется для получения детальных результатов.

Несколько примеров использования этой команды см. в разделе "**УПРОЩЕННОЕ КОНФИГУРИРОВАНИЕ**" ниже.

## **УПРОЩЕННОЕ КОНФИГУРИРОВАНИЕ**

В этом разделе описан упрощенный интерфейс для конфигурирования **vinum** с помощью команд **concat**, **mirror** и **stripe**. Эти команды создают удобные конфигурации для ряда стандартных ситуаций, но не обеспечивают такую гибкость, как команда **create**.

Описание этих команд было представлено выше. Вот ряд примеров, построенных на одном и том же наборе дисков. Учтите, что первый диск, **/dev/da1h**, - меньше других. Это влияет на размеры поддисков каждого вида.

Во всех следующих примерах используется **-v**, чтобы показать команды, передаваемые в систему, а также для получения информации о структуре тома. Без опции **-v** эти команды ничего не выдают.

### **Том с одним конкатенированным сплетением**

Используйте том с одним конкатенированным сплетением для получения максимально возможного объема дисковой памяти без обеспечения защиты от сбоев диска:

```

vinum -> concat -v /dev/dal1h /dev/da2h /dev/da3h /dev/da4h
volume vinum0
  plex name vinum0.p0 org concat
  drive vinumdrive0 device /dev/dal1h
    sd name vinum0.p0.s0 drive vinumdrive0 size 0
  drive vinumdrive1 device /dev/da2h
    sd name vinum0.p0.s1 drive vinumdrive1 size 0
  drive vinumdrive2 device /dev/da3h
    sd name vinum0.p0.s2 drive vinumdrive2 size 0
  drive vinumdrive3 device /dev/da4h
    sd name vinum0.p0.s3 drive vinumdrive3 size 0
V vinum0                               State: up           Plexes:                1 Size:
2134 MB
P vinum0.p0                           C State: up         Subdisks:              4 Size:
2134 MB
S vinum0.p0.s0                        State: up           D: vinumdrive0         Size:                414 MB
S vinum0.p0.s1                        State: up           D: vinumdrive1         Size:                573 MB
S vinum0.p0.s2                        State: up           D: vinumdrive2         Size:                573 MB
S vinum0.p0.s3                        State: up           D: vinumdrive3         Size:                573 MB

```

В нашем случае использовано все место на всех четырех дисках, что дало нам в итоге 2134 Мбайта.

### Том с одним сплетением с чередованием

Том из одного сплетения с чередованием может обеспечить более высокую производительность, чем конкатенированное сплетение, но ограничения на сплетения с чередованием могут привести к уменьшению размера тома. Такой том тоже не обеспечивает защиту от сбоя диска:

```

vinum -> stripe -v /dev/dal1h /dev/da2h /dev/da3h /dev/da4h
drive vinumdrive0 device /dev/dal1h
drive vinumdrive1 device /dev/da2h
drive vinumdrive2 device /dev/da3h
drive vinumdrive3 device /dev/da4h
volume vinum0
  plex name vinum0.p0 org striped 279k
    sd name vinum0.p0.s0 drive vinumdrive0 size 849825b
    sd name vinum0.p0.s1 drive vinumdrive1 size 849825b
    sd name vinum0.p0.s2 drive vinumdrive2 size 849825b
    sd name vinum0.p0.s3 drive vinumdrive3 size 849825b
V vinum0                               State: up           Plexes:                1 Size:
1659 MB
P vinum0.p0                           S State: up         Subdisks:              4 Size:
1659 MB
S vinum0.p0.s0                        State: up           D: vinumdrive0         Size:                414 MB
S vinum0.p0.s1                        State: up           D: vinumdrive1         Size:                414 MB
S vinum0.p0.s2                        State: up           D: vinumdrive2         Size:                414 MB
S vinum0.p0.s3                        State: up           D: vinumdrive3         Size:                414 MB

```

В этом случае размер поддисков ограничен наименьшим доступным диском, так что, получившийся в результате том имеет размер всего 1659 Мбайт.

### Том с зеркалированием из двух конкатенированных сплетений

Для повышения надежности можно использовать зеркалированный и конкатенированный том:

```

vinum -> mirror -v -n mirror /dev/dal1h /dev/da2h /dev/da3h /dev/da4h
drive vinumdrive0 device /dev/dal1h
drive vinumdrive1 device /dev/da2h
drive vinumdrive2 device /dev/da3h
drive vinumdrive3 device /dev/da4h
volume mirror setupstate
  plex name mirror.p0 org concat
    sd name mirror.p0.s0 drive vinumdrive0 size 0b
    sd name mirror.p0.s1 drive vinumdrive2 size 0b
  plex name mirror.p1 org concat
    sd name mirror.p1.s0 drive vinumdrive1 size 0b
    sd name mirror.p1.s1 drive vinumdrive3 size 0b
V mirror                               State: up           Plexes:                2 Size:
1146 MB

```

```

P mirror.p0                      C State: up                      Subdisks:  2 Size:
988 MB
P mirror.p1                      C State: up                      Subdisks:  2 Size:
1146 MB
S vinum0.p0.s0                  D: vinumdrive0 Size: 414 MB
S vinum0.p0.s2                  D: vinumdrive2 Size: 414 MB
S vinum0.p0.s1                  D: vinumdrive1 Size: 414 MB
S vinum0.p0.s3                  D: vinumdrive3 Size: 414 MB

```

В этом примере явно указано имя тома, **mirror**. Поскольку один диск меньше других, два сплетения имеют разный размер, и последние 158 Мбайт тома не защищены от сбоев. Чтобы гарантировать полную надежность в такой ситуации, используйте команду **create** для создания тома размером 988 Мбайт.

## Том с зеркалированием из двух сплетений с чередованием

В качестве альтернативы, используйте опцию **-s** для создания зеркалированного тома из двух сплетений с чередованием:

```

vinum -> mirror -v -n raid10 -s /dev/dalh /dev/da2h /dev/da3h /dev/da4h
drive vinumdrive0 device /dev/dalh
drive vinumdrive1 device /dev/da2h
drive vinumdrive2 device /dev/da3h
drive vinumdrive3 device /dev/da4h
volume raid10 setupstate
plex name raid10.p0 org striped 279k
sd name raid10.p0.s0 drive vinumdrive0 size 849825b
sd name raid10.p0.s1 drive vinumdrive2 size 849825b
plex name raid10.p1 org striped 279k
sd name raid10.p1.s0 drive vinumdrive1 size 1173665b
sd name raid10.p1.s1 drive vinumdrive3 size 1173665b
V raid10                        State: up                      Plexes:  2 Size: 1146 MB
P raid10.p0                    S State: up                      Subdisks:  2 Size:
829 MB
P raid10.p1                    S State: up                      Subdisks:  2 Size:
1146 MB
S raid10.p0.s0                State: up                      PO:  0 B Size: 414 MB
S raid10.p0.s1                State: up                      PO: 279 kB Size: 414 MB
S raid10.p1.s0                State: up                      PO:  0 B Size: 573 MB
S raid10.p1.s1                State: up                      PO: 279 kB Size: 573 MB

```

В этом случае полезная часть тома - еще меньше, поскольку первое сплетение было уменьшено до размера самого маленького диска.

## ФАЙЛ КОНФИГУРАЦИИ

Утилита **vinum** требует, чтобы все параметры команд **create** были в файле конфигурации. Записи в файле конфигурации определяют тома, сплетения и поддиски, и могут быть в произвольном формате, но каждая запись должна быть в отдельной строке.

### Единицы измерения

Некоторые параметры файла конфигурации задают размер (длины, размеры полос). Эти значения можно задавать в байтах или с указанием одной из следующих единиц измерения:

- s**      Указывает, что значение задано в секторах по 512 байтов.
- k**      Указывает, что значение задано в килобайтах (1024 байта).
- m**      Указывает, что значение задано в мегабайтах (1048576 байтов).
- g**      Указывает, что значение задано в гигабайтах (1073741824 байта).

- b** Используется для совместимости с VERITAS. Указывает, что значение задано в блоках по 512 байтов. Это сокращение неоднозначно, поскольку слово "блок" используется в разных смыслах, и его использование не рекомендуется. Используйте единицу измерения 's'.

Например, значение 16777216 байтов можно также записать как **16m**, **16384k** или **32768s**.

Файл конфигурации может содержать следующие записи:

**drive имя\_имя\_устройства [опции]**

Определяет диск. Поддерживаются следующие опции:

**device имя\_устройства**

Задаёт устройство, на котором находится диск. **Имя\_устройства** должно быть именем раздела диска, например, **/dev/da1e** или **/dev/ad3s2h**, это устройство должно быть типа **vinum**. Не используйте раздел "c", который зарезервирован для представления всего диска.

**hotspare**

Определяет диск как диск "горячей замены" ("hot spare" drive), который предназначен для автоматической замены сбойного диска. Утилита **vinum** не позволяет использовать этот диск для любых других целей. В частности, на нем нельзя создавать поддиски. Эта возможность еще не полностью реализована.

**volume имя [опции]**

Определяет том с указанным **именем**. Поддерживаются следующие опции:

**plex имя\_сплетения**

Добавляет указанное сплетение к тому. Если имя сплетения указано как \*, утилита **vinum** будет искать определение сплетения в следующей записи файла конфигурации после определения тома.

**readpol правило**

Задаёт *правило чтения* (read policy) для тома. В качестве правила можно указывать значение **round** или **prefer имя\_предпочтительного\_сплетения**. Утилита **vinum** удовлетворяет запросы на чтение только с одного из сплетений. Правило чтения **round** означает, что каждое последующее чтение должно выполняться с другого сплетения, циклически. Правило чтения **prefer** требует выполнять чтение каждый раз с явно указанного сплетения.

**setupstate**

При создании тома из нескольких сплетений предполагать, что содержимое всех сплетений согласовано. Обычно это не так, поэтому стандартно утилита **vinum** устанавливает для всех сплетений, кроме первого, состояние **faulty**. Используйте команду **start**, чтобы сначала привести их в согласованное состояние. В случае конкатенированных сплетений и сплетений с чередованием, однако, оставление сплетений в несогласованном состоянии обычно не вызывает проблем: при использовании тома для создания файловой системы или раздела подкачки прежнее содержимое дисков обычно не представляет интереса, поэтому исходную согласованность можно проигнорировать. Если вы готовы пойти на этот риск, используйте ключевое слово **setupstate**. Оно применяется только к сплетениям, определенным непосредственно после соответствующего тома в файле конфигурации. При добавлении сплетений к тому в дальнейшем, их необходимо интегрировать вручную с помощью команды **start**.

Учтите, что для сплетений типа RAID-5 надо обязательно использовать команду **init**: иначе при сбое одного из поддисков произойдет существенное повреждение данных.

**plex [опции]**

Определяет сплетение. В отличие от тома, имя сплетения задавать не обязательно. Поддерживаются следующие опции:

**name имя\_сплетения**

Задаёт имя сплетения. Учтите, что ключевое слово **name** надо обязательно использовать при именовании сплетения или поддиска.

**org организация [размер\_полосы]**

Задаёт организацию сплетения. В качестве **организации** может быть задано одно из значений **concat**, **striped** или **raid5**. Для сплетений типа **striped** и **raid5** необходимо указывать параметр **размер\_полосы**, а вот для сплетения типа **concat** его указывать нельзя. Для сплетения типа **striped** значение **размер\_полосы** задаёт ширину каждой полосы. Для сплетения типа **raid5** оно задаёт



размер группы. *Группа* - это часть сплетения, хранящая биты четности на одном и том же поддиске. Размер сплетения должен быть кратен размеру группы (другими словами, результат деления размера сплетения на размер полосы должен быть целым числом), а размер группы должен быть кратен размеру сектора диска (512 байтов).

Для оптимальной производительности полосы должны быть размером минимум 128 Кбайт: меньший размер приведет к существенному увеличению количества операций ввода/вывода из-за распределения отдельных запросов по нескольким дискам. Повышение производительности из-за увеличения количества одновременных передач, вызванное таким распределением, не компенсирует падение производительности в связи с дополнительными задержками. Хорошей рекомендацией будет выбирать размер полосы в диапазоне от 256 Кбайт до 512 Кбайт. Степеней двойки при этом, однако, лучше избегать: обычно они приводят к размещению всех суперблоков на первом поддиске. Упрощенные команды используют размер полосы 279 Кбайт, что дает разумное распределение суперблоков.

Сплетение с чередованием должно иметь минимум два поддиск (иначе это просто конкатенированное сплетение), причем, одинакового размера. Сплетение типа RAID-5 должно иметь как минимум три поддиска, причем, одинакового размера. Практически сплетение типа RAID-5 должно включать не менее 5 поддисков.

#### **volume имя\_тома**

Добавляет сплетение к указанному тому. Если ключевое слово **volume** не указано, сплетение будет добавлено к последнему тому, упоминавшемуся в файле конфигурации.

#### **sd имя\_поддиска смещение**

Добавляет указанный **поддиск** к сплетению с указанного **смещения**.

#### **subdisk [опции]**

Определяет поддиск. Поддерживаются следующие опции:

##### **name имя**

Задаёт **имя** поддиска. Задавать имя для поддиска не обязательно - см. раздел "ИМЕНОВАНИЕ ОБЪЕКТОВ" выше. Учтите, что если необходимо задать имя поддиска, ключевое слово **name** указывать обязательно.

##### **plexoffset смещение**

Задаёт начальное смещение поддиска в сплетении. Если смещение не указано, утилита **vinum** выделяет место сразу после предыдущего поддиска, если он уже был определен, или с начала сплетения в противном случае.

##### **driveoffset смещение**

Задаёт начальное смещение поддиска на диске. Если **смещение** не задано, **vinum** выделяет первый же фрагмент свободного места достаточного размера (см. **length**) на диске.

##### **length длина**

Задаёт "длину" (размер) поддиска. Это ключевое слово указывать обязательно. Стандартного значения нет, но можно указать значение 0 - оно означает "использовать самый большой фрагмент свободного места на диске". Если диск пустой, это означает, что для поддиска будет использован весь диск. Ключевое слово **length** можно сокращать до **len**.

##### **plex сплетение**

Задаёт сплетение, которому принадлежит поддиск. По умолчанию, поддиск принадлежит последнему определенному сплетению.

##### **drive диск**

Задаёт диск, на котором находится поддиск. По умолчанию, поддиск находится на последнем указанном диске.

##### **retryerrors**

Указывает, что поддиск не должен отключаться в случае невосстановимых ошибок. Обычно **vinum** при возникновении таких ошибок делает весь поддиск недоступным.

## **ПРИМЕР ФАЙЛА КОНФИГУРАЦИИ**



```
# Пример файла конфигурации vinum
#
# Наши диски
drive drive1 device /dev/da1h
drive drive2 device /dev/da2h
drive drive3 device /dev/da3h
drive drive4 device /dev/da4h
drive drive5 device /dev/da5h
drive drive6 device /dev/da6h
# Том из одного сплетения с чередованием
volume tinyvol
    plex org striped 279k
        sd length 64m drive drive2
        sd length 64m drive drive4
volume stripe
    plex org striped 279k
        sd length 512m drive drive2
        sd length 512m drive drive4
# Два сплетения
volume concat
    plex org concat
        sd length 100m drive drive2
        sd length 50m drive drive4
    plex org concat
        sd length 150m drive drive4
# Том из одного сплетения с чередованием и одного конкатенированного

volume strcon
    plex org striped 279k
        sd length 100m drive drive2
        sd length 100m drive drive4
    plex org concat
        sd length 150m drive drive2
        sd length 50m drive drive4
# Том из сплетения типа RAID-5 и сплетения с чередованием
# Обратите внимание, что сплетение RAID-5 длинее на
# один поддиск
volume vol5
    plex org striped 491k
        sd length 1000m drive drive2
        sd length 1000m drive drive4
    plex org raid5 273k
        sd length 500m drive drive1
        sd length 500m drive drive2
        sd length 500m drive drive3
        sd length 500m drive drive4
        sd length 500m drive drive5
```

## ПРИНЦИПЫ РАЗМЕЩЕНИЯ ДИСКОВ

Диски **vinum** в настоящее время являются разделами дисков BSD. Они должны быть типа **vinum**, чтобы предотвратить перезапись данных, используемых для других целей. Для изменения определения типа раздела используйте команду **disklabel -e**. Ниже представлено типичное разбиение диска на разделы, выдаваемое **disklabel(8)**:

```
8 partitions:
#      size  offset  fstype  [fsize bsize bps/cpg]
a:    81920   344064   4.2BSD      0      0      0 # (Cyl.  240*- 297*)
b:    262144   81920    swap              0      0      0 # (Cyl.   57*- 240*)
c:    4226725      0   unused              0      0      0 # (Cyl.0 - 2955*)
e:     81920      0   4.2BSD      0      0      0 # (Cyl.0 - 57*)
f:   1900000   425984   4.2BSD      0      0      0 # (Cyl. 297*- 1626*)
g:   1900741   2325984   vinum        0      0      0 # (Cyl. 1626*- 2955*)
```

В этом примере, раздел **"g"** может использоваться как раздел **vinum**. Разделы **"a"**, **"e"** и **"f"** могут использоваться для создания файловых систем UFS или как разделы **ccd**. Раздел **"b"** является разделом подкачки, а раздел **"c"** представляет весь диск в целом и не должен использоваться для любых других целей.

Утилита **vinum** использует первые 265 секторов каждого раздела для информации о конфигурации, поэтому максимальный размер поддиска будет на 265 секторов меньше, чем объем диска.

## ЖУРНАЛЬНЫЙ ФАЙЛ

Утилита **vinum** поддерживает журнальный файл, по умолчанию - `/var/log/vinum_history`, в который она записывает выполненные команды **vinum**. Имя этого файла можно изменить путем установки соответствующего значения переменной среды **VINUM\_HISTORY**.

Перед каждым сообщением в журнальном файле указывается дата и время. Стандартный формат - `"%e %b %Y %H:%M:%S"`. Подробнее о строке формата см. на странице справочного руководства [strftime\(3\)](#). Нестандартный формат можно задать в переменной среды **VINUM\_DATEFORMAT**.

## КАК НАСТРОИТЬ VINUM

В этом разделе представлены практические советы по использованию **vinum** в системе.

### Где размещать данные

Прежде всего, надо решить, где размещать данные. Необходимо выделить разделы диска специально для **vinum**. Это должны быть именно разделы, а не устройства в целом; нельзя использовать также раздел `"c"`. Например, прекрасно подходят имена устройств `/dev/da0e` или `/dev/ad3s4a`. Не подходят устройства `/dev/da0` и `/dev/da0s1`, поскольку представляют диск в целом, а не раздел, а также раздел `/dev/ad1c`, представляющий диск в целом - он должен быть типа **unused**. Примеры см. выше в разделе [ПРИНЦИПЫ РАЗМЕЩЕНИЯ ДИСКОВ](#).

### Проектирование томов

Настройка томов **vinum** зависит от поставленных целей. Есть несколько вариантов:

1. Может потребоваться объединить несколько небольших дисков для получения файловой системы достаточного размера. Например, при наличии пяти небольших дисков, если необходимо использовать все дисковое пространство как один том, можно создать конфигурационный файл следующего вида:

```
2. drive d1 device /dev/da2e
3. drive d2 device /dev/da3e
4. drive d3 device /dev/da4e
5. drive d4 device /dev/da5e
6. drive d5 device /dev/da6e
7. volume bigger
8.   plex org concat
9.     sd length 0 drive d1
10.    sd length 0 drive d2
11.    sd length 0 drive d3
12.    sd length 0 drive d4
13.    sd length 0 drive d5
```

В данном случае длина поддисков задается как 0, что означает "использоваться самый большой фрагмент свободного места, который удастся найти на диске". Если поддиск - единственный на диске, под него будет использоваться все свободное место.

14. Можно настраивать **vinum** для получения дополнительной защиты от сбоев дисков. Для этого можно использовать либо конфигурацию RAID-1, которую также называют "зеркалирование", либо конфигурацию RAID-5, с битами четности.

Для настройки зеркалирования создайте несколько сплетений в томе. Например, для создания зеркалированного тома размером 2 Гбайта можно использовать следующий файл конфигурации:

```
drive d1 device /dev/da2e
drive d2 device /dev/da3e
volume mirror
  plex org concat
    sd length 2g drive d1
  plex org concat
    sd length 2g drive d2
```

При создании дисков с зеркалированием важно убедиться, что данные каждого сплетения находятся на другом физическом диске, чтобы драйвер **vinum** мог обращаться ко всему адресному пространству тома даже при сбое одного диска. Учтите, что для каждого сплетения потребуется столько же места, сколько и для всего тома: в нашем примере том имеет размер 2 Гбайта, поэтому каждое сплетение (и каждый поддиск) должны быть размером 2 Гбайта, так что всего на дисках будет использовано 4 Гбайта.

Для настройки массива RAID-5 создайте одно сплетение типа **raid5**. Например, для создания эквивалентного тома размером 2 Гбайта, остающегося работоспособным при сбое одного диска, можно использовать следующий файл конфигурации:

```
drive d1 device /dev/da2e
drive d2 device /dev/da3e
drive d3 device /dev/da4e
drive d4 device /dev/da5e
drive d5 device /dev/da6e
volume raid
  plex org raid5 433k
    sd length 512m drive d1
    sd length 512m drive d2
    sd length 512m drive d3
    sd length 512m drive d4
    sd length 512m drive d5
```

Сплетения типа RAID-5 требуют минимум трех поддисков, один из которых используется для хранения информации о четности и для хранения данных не доступен. Чем больше дисков используется, тем большая доля дискового пространства может использоваться для хранения собственно данных. В нашем примере суммарно используется 2,5 Гбайта по сравнению с 4 Гбайтами в конфигурации с зеркалированием. Если придется использовать всего три диска, для хранения той же информации потребуется 3 Гбайта, например:

```
drive d1 device /dev/da2e
drive d2 device /dev/da3e
drive d3 device /dev/da4e
volume raid
  plex org raid5 433k
    sd length 1g drive d1
    sd length 1g drive d2
    sd length 1g drive d3
```

Как и при создании дисков с зеркалированием, важно проверить, что данные каждого поддиска находятся на другом физическом диске, чтобы драйвер **vinum** мог получить доступ ко всему адресному пространству тома даже при сбое одного диска.

15. Можно настроить **vinum** для обеспечения одновременного доступа к файловой системе большего количества пользователей. Во многих случаях, доступ к файловой системе ограничивается скоростью диска. Распределяя том по нескольким дискам, можно увеличить пропускную способность при многопользовательском доступе. Этот метод дает минимальные преимущества или вообще не дает преимуществ, если процессы обращаются к дискам по одному. Утилита **vinum** использует метод под названием "чередование" (иногда соответствующий массив называют RAID-0) для увеличения одновременности при доступе. Имя RAID-0 сбивает с толку: чередование не обеспечивает ни избыточности, ни повышенной надежности. Фактически, оно снижает надежность, поскольку сбой единственного диска делает бесполезным весь том, и чем больше дисков входит в том, тем больше вероятность, что на одном из них произойдет сбой.

Для реализации чередования используйте сплетение с чередованием:

```
drive d1 device /dev/da2e
drive d2 device /dev/da3e
drive d3 device /dev/da4e
drive d4 device /dev/da5e
volume raid
    plex org striped 433k
        sd length 512m drive d1
        sd length 512m drive d2
        sd length 512m drive d3
        sd length 512m drive d4
```

Сплетение с чередованием должно состоять, как минимум, из двух поддисков, но повышение производительности будет больше при увеличении количества дисков.

16. Можно попытаться получить лучшее из двух миров и обеспечить как защиту от сбоев, так и высокую производительность. Соответствующий массив иногда называют RAID-10 (комбинация RAID-1 и RAID-0), хотя и это имя сбивает с толку. Можно реализовать такой массив **vinum** с помощью следующего файла конфигурации:

```
17. drive d1 device /dev/da2e
18. drive d2 device /dev/da3e
19. drive d3 device /dev/da4e
20. drive d4 device /dev/da5e
21. volume raid setupstate
22.  plex org striped 433k
23.      sd length 512m drive d1
24.      sd length 512m drive d2
25.      sd length 512m drive d3
26.      sd length 512m drive d4
27.  plex org striped 433k
28.      sd length 512m drive d4
29.      sd length 512m drive d3
30.      sd length 512m drive d2
31.      sd length 512m drive d1
```

В данном случае сплетения созданы с чередованием, что повышает производительность, а то, что их два, повышает надежность. Обратите внимание, что в этом примере поддиски второго сплетения идут в порядке, противоположном первому сплетению. Это сделано из соображений производительности, которые обсуждаются далее. Кроме того, определение тома включает ключевое слово **setupstate**, что гарантирует запуск всех сплетений сразу после создания.

## Создание томов

После создания необходимых файлов конфигурации запустите **vinum** и создайте тома. В следующем примере конфигурация задана в файле **configfile**:

```
# vinum create -v configfile
1: drive d1 device /dev/da2e
2: drive d2 device /dev/da3e
3: volume mirror
4:   plex org concat
5:     sd length 2g drive d1
6:   plex org concat
7:     sd length 2g drive d2
Configuration summary

Drives:      2 (4 configured)
Volumes:     1 (4 configured)
Plexes:      2 (8 configured)
Subdisks:    2 (16 configured)

Drive d1:    Device /dev/da2e
             Created on vinum.lemis.com at Tue Mar 23 12:30:31 1999
             Config last updated Tue Mar 23 14:30:32 1999
             Size:  60105216000 bytes (57320 MB)
             Used:  2147619328 bytes (2048 MB)
             Available: 57957596672 bytes (55272 MB)
```

```

State: up
Last error: none
Drive d2: Device /dev/da3e
Created on vinum.lemis.com at Tue Mar 23 12:30:32 1999
Config last updated Tue Mar 23 14:30:33 1999
Size: 60105216000 bytes (57320 MB)
Used: 2147619328 bytes (2048 MB)
Available: 57957596672 bytes (55272 MB)
State: up
Last error: none

Volume mirror: Size: 2147483648 bytes (2048 MB)
State: up
Flags:
2 plexes
Read policy: round robin

Plex mirror.p0: Size: 2147483648 bytes (2048 MB)
Subdisks:1
State: up
Organization: concat
Part of volume mirror
Plex mirror.p1: Size: 2147483648 bytes (2048 MB)
Subdisks:1
State: up
Organization: concat
Part of volume mirror

Subdisk mirror.p0.s0:
Size: 2147483648 bytes (2048 MB)
State: up
Plex mirror.p0 at offset 0
Subdisk mirror.p1.s0:
Size: 2147483648 bytes (2048 MB)
State: up
Plex mirror.p1 at offset 0

```

Опция **-v** требует от утилиты **vinum** выдавать содержимое файла по ходу конфигурирования. Далее она выдает информацию о текущей конфигурации в том же формате, что и команда **list -v**.

## Создание дополнительных томов

После создания томов **vinum**, драйвер **vinum** отслеживает их состояние в своих внутренних файлах конфигурации. Снова создавать их не придется. В частности, при повторном выполнении команды **create** будут созданы дополнительные объекты:

```

# vinum create sampleconfig
Configuration summary

Drives:      2 (4 configured)
Volumes:     1 (4 configured)
Plexes:      4 (8 configured)
Subdisks:    4 (16 configured)

D d1          State: up      Device /dev/da2e    Avail: 53224/57320 MB (92%)
D d2          State: up      Device /dev/da3e    Avail: 53224/57320 MB (92%)

V mirror State: up      Plexes:      4 Size: 2048 MB

P mirror.p0   C State: up      Subdisks:    1 Size: 2048 MB
P mirror.p1   C State: up      Subdisks:    1 Size: 2048 MB
P mirror.p2   C State: up      Subdisks:    1 Size: 2048 MB
P mirror.p3   C State: up      Subdisks:    1 Size: 2048 MB

S mirror.p0.s0 State: up      PO:0 B Size: 2048 MB
S mirror.p1.s0 State: up      PO:0 B Size: 2048 MB
S mirror.p2.s0 State: up      PO:0 B Size: 2048 MB
S mirror.p3.s0 State: up      PO:0 B Size: 2048 MB

```

Как показывает этот пример (на этот раз добавлена опция **-f**), повторное выполнение команды **create** привело к созданию четырех новых сплетений, каждое - с новым поддиск. Если необходимо создать другие тома, создайте для них новые файлы конфигурации. В них не обязательно задавать диски, о которых драйвер **vinum** уже знает. Например, для создания тома **raid** на четырех дисках, **/dev/da1e**, **/dev/da2e**, **/dev/da3e** и **/dev/da4e**, достаточно упомянуть два оставшихся:

```
drive d3 device /dev/da1e
drive d4 device /dev/da4e
volume raid
  plex org raid5 433k
  sd size 2g drive d1
  sd size 2g drive d2
  sd size 2g drive d3
  sd size 2g drive d4
```

С помощью этого файла конфигурации мы получим:

```
# vinum create newconfig
Configuration summary

Drives:                4 (4 configured)
Volumes:                2 (4 configured)
Plexes:                 5 (8 configured)
Subdisks:              8 (16 configured)

D d1                   State: up Device /dev/da2e   Avail: 51176/57320 MB (89%)
D d2                   State: up Device /dev/da3e   Avail: 53220/57320 MB (89%)
D d3                   State: up Device /dev/da1e   Avail: 53224/57320 MB (92%)
D d4                   State: up Device /dev/da4e   Avail: 53224/57320 MB (92%)

V mirror State: down   Plexes: 4 Size: 2048 MB
V raid   State: down   Plexes: 1 Size: 6144 MB

P mirror.p0 C State: init Subdisks: 1 Size: 2048 MB
P mirror.p1 C State: init Subdisks: 1 Size: 2048 MB
P mirror.p2 C State: init Subdisks: 1 Size: 2048 MB
P mirror.p3 C State: init Subdisks: 1 Size: 2048 MB
P raid.p0   R5 State: init Subdisks: 4 Size: 6144 MB

S mirror.p0.s0 State: up PO: 0 B Size: 2048 MB
S mirror.p1.s0 State: up PO: 0 B Size: 2048 MB
S mirror.p2.s0 State: up PO: 0 B Size: 2048 MB
S mirror.p3.s0 State: up PO: 0 B Size: 2048 MB
S raid.p0.s0   State: empty PO: 0 B Size: 2048 MB
S raid.p0.s1   State: empty PO: 433 kB Size: 2048 MB
S raid.p0.s2   State: empty PO: 866 kB Size: 2048 MB
S raid.p0.s3   State: empty PO: 1299 kB Size: 2048 MB
```

Обратите внимание на размер сплетения RAID-5: он составляет всего 6 Гбайт, хотя совместно все его компоненты занимают 8 Гбайт дискового пространства. Причина в том, что пространство объемом с один поддиск используется для хранения данных о четности.

## Перезапуск Vinum

После перезагрузки системы запустите **vinum** с помощью команды **start**:

```
# vinum start
```

Эта команда запустит все диски **vinum** в системе. Если, по любой причине, необходимо запустить только некоторые из них, используйте команду **read**.

## Соображения производительности

Есть ряд ошибочных представлений о том, как настраивать RAID-массив для достижения наилучшей производительности. В частности, в большинстве систем используется слишком маленький размер полосы. Следующее обсуждение применимо ко всем RAID-системам, а не только к **vinum**.

Система блочного ввода/вывода ОС FreeBSD выдает запросы на чтение от 0,5 Кбайта до 128 Кбайт; в среднем - около 8 Кбайт. Ни в одной системе чередования нельзя предотвратить разбиение запроса на два физических запроса, и если сделать размер полосы слишком маленьким, он может быть разбит и на несколько

запросов. Это приведет к существенному падению производительности: уменьшение времени передачи для каждого диска с лихвой компенсируется на порядок увеличивающимся ожиданием доступа.

При современных размерах дисков и состоянии системы ввода/вывода FreeBSD, следует ожидать достаточно небольшой доли фрагментированных запросов при использовании полос размером от 256 до 512 Кбайт; хотя при обоснованном применении RAID-массивов нет очевидных причин не увеличить размер до 2-4 Мбайт для больших дисков.

При выборе размера полосы учитывайте, что большинство современных файловых систем UFS использует группы цилиндров размером 32 Мбайта. Если размер полосы и количество дисков - степени двойки, вероятно, что все суперблоки и индексные дескрипторы будут помещены на один и тот же поддиск, что существенно повлияет на производительность. Выбирайте лучше нечетное число, например, 479 Кбайт.

Самый простой способ учесть влияние любой передачи данных в системе с множественным доступом - рассматривать ее с точки зрения потенциального узкого места, дисковой подсистемы: сколько всего времени работы диска потребует эта передача? Поскольку практически все кэшируется, время от выдачи запроса до его выполнения не так важно: важно, сколько всего времени запрос активно использует диски, время, в течение которого диски недоступны для выполнения передач других данных. В результате, не так уж важно, выполняются ли передачи одновременно или в разное время. Практически, время, которое нас интересует - это сумма общей задержки (времени на позиционирование и ожидания при вращении пластин, или времени, за которое требуемые данные окажутся под головками диска) и общего времени передачи. Для передачи данных по одному и тому же адресу на диски с той же скоростью вращения, время передачи зависит только от объема передаваемых данных.

Рассмотрим типичную статью дискуссионной группы или web-страницу размером 24 Кбайта, которая, вероятно, будет прочитана одной операцией ввода/вывода. Пусть диски обеспечивают скорость передачи 6 Мбайт/сек, среднее время позиционирования составляет 8 миллисекунд, а файловая система имеет размер блока 4 Кбайта. Поскольку прочитать надо 24 Кбайта, фрагменты можно не учитывать, так что, считаем, что файл начинается на границе очередных 4 Кбайт. Количество необходимых передач зависит от того, где начинается блок: в данном случае, оно вычисляется по формуле  $(S + F - 1) / S$ , где  $S$  - размер полосы в блоках файловой системы, а  $F$  - размер файла в блоках файловой системы.

1. При размере полосы 4 Кбайта потребуется 6 передач. Общая нагрузка на дисковую подсистему: задержка - 48 миллисекунд, передача - 2 миллисекунды, всего - 50 миллисекунд.
2. При размере полосы 8 Кбайт в среднем потребуется 3,5 передачи. Общая нагрузка на дисковую подсистему: задержка - 28 миллисекунд, передача - 2 миллисекунды, всего - 30 миллисекунд.
3. При размере полосы 16 Кбайт в среднем потребуется 2,25 передачи. Общая нагрузка на дисковую подсистему: задержка - 18 миллисекунд, передача - 2 миллисекунды, всего - 20 миллисекунд.
4. При размере полосы 256 Кбайт в среднем потребуется 1,08 передачи. Общая нагрузка на дисковую подсистему: задержка - 8,6 миллисекунд, передача - 2 миллисекунды, всего - 10,6 миллисекунд.
5. При размере полосы 4 Мбайта в среднем потребуется 1,0009 передачи. Общая нагрузка на дисковую подсистему: задержка - 8,01 миллисекунды, передача - 2 миллисекунды, всего - 10,01 миллисекунды.

Похоже, что у некоторых аппаратных RAID-систем есть проблемы с большими полосами: они, по-видимому, передают на диск и с диска только всю полосу, так что увеличение размера полосы может отрицательно влиять на производительность. Утилита **vinum** от этой проблемы не страдает: все обмены данными оптимизируются и ненужные данные не передаются.

Учтите, что ни одна из широко известных программ оценки производительности не реализует условия действительно множественного доступа (более 100 одновременно работающих пользователей), так что правильность представленных утверждений проверить трудно.

С учетом представленных соображений, на производительность тома **vinum** влияют следующие факторы:

1. Чередование повышает производительность только при множественном доступе, поскольку повышает вероятность того, что отдельные запросы будут переадресованы на разные диски.
2. Конкатенированные файловые системы UFS из нескольких дисков тоже могут повысить производительность при доступе к множеству файлов, поскольку UFS делит файловую систему на группы цилиндров и пытается поддерживать файлы в одной группе цилиндров. В общем случае, В общем случае, это не так эффективно, как чередование.
3. Зеркалирование может повысить производительность множественного доступа при чтении, поскольку по умолчанию каждое следующее чтение **vinum** выполняет со следующего сплетения.
4. Зеркалирование снижает производительность для всех видов записи, будь-то множественные или отдельные, поскольку данные надо записать на все сплетения. Это объясняет выбранное размещение дисков в представленном ранее примере конфигурации с зеркалированием: если соответствующий поддиск в каждом сплетении располагается на другом физическом диске, команды записи могут выполняться параллельно, а при размещении на одном физическом диске они будут выполняться строго последовательно.
5. Чтения с массива RAID-5 имеют, по сути, те же особенности, что и чтения с массива с чередованием, если только сплетение с чередованием не входит в том с зеркалированием, - в этом случае производительность тома с зеркалированием будет выше.
6. Скорость записи на массив RAID-5 составляет примерно 25% от скорости записи на массив с чередованием: для выполнения записи утилита **vinum** должна сначала прочитать блок данных и соответствующий блок четности, выполнить ряд вычислений и записать блок четности и блок данных, что требует в четыре раза больше передач, чем при записи на сплетение с чередованием. С другой стороны, это компенсируется отсутствием затрат на зеркалирование, поэтому запись на том с единственным сплетением типа RAID-5 выполняется примерно вдвое медленнее, чем на правильно сконфигурированный том из двух сплетений с чередованием.
7. При изменении конфигурации **vinum** (например, при добавлении или удалении объектов, или при изменении состояния одного из объектов), **vinum** записывает до 128 Кбайт обновленной информации о конфигурации на каждый диск. Чем больше дисков задействовано, тем больше времени это займет.

## Создание файловых систем на томах Vinum



Перед созданием файловой системы на томе **vinum** запускать утилиту **disklabel(8)** не нужно. Просто выполните команду **newfs(8)**. Используйте опцию **-v**, чтобы указать, что устройство не разбито на разделы. Например, чтобы создать файловую систему на томе **mirror**, выполните следующую команду:

```
# newfs -v /dev/vinum/mirror
```

При конфигурировании **vinum** следует учитывать также следующее:

- Создание нескольких дисков **vinum** на одном физическом диске не дает никаких преимуществ. На каждом диске 131,5 Кбайта используется для метки и хранения информации о конфигурации, а изменение конфигурации вызывает снижение производительности. Используйте лучше поддиски соответствующего размера.
- Можно увеличить размер конкатенированного сплетения **vinum**, но вот размер сплетений с чередованием и сплетений типа RAID-5 в настоящее время увеличить нельзя. В настоящее время размер существующей файловой системы UFS увеличить тоже нельзя, но планируется сделать сплетения и файловые системы расширяемыми.

## УПРАВЛЕНИЕ СОСТОЯНИЕМ

Для объектов **vinum** существует понятие состояния. Подробнее см. на странице справочного руководства **vinum(4)**. Объекты полностью доступны только в запущенном (**up**) состоянии. Чтобы изменить состояние объекта на **up**, используйте команду **start**. Для перевода объекта в состояние **down** используйте команду **stop**. Обычно переход в другие состояния осуществляется автоматически с учетом взаимосвязей объектов. Например, при добавлении сплетения к тому поддиски сплетения оказываются в состоянии **empty**, показывая, что хотя фактически диск доступен, данные на поддиске некорректны. В результате установки такого состояния, соответствующее сплетение будет установлено в состояние сбоя (**faulty**).

### Состояние "reviving"

Во многих случаях при запуске поддиска система должна скопировать данные на поддиск. В зависимости от размера поддиска, это может потребовать много времени. На это время поддиск устанавливается в состояние **reviving** ("оживляется"). После успешного завершения операции копирования он автоматически запускается (устанавливается в состояние **up**). Можно останавливать и перезапускать процесс, выполняющий "оживление". Система отслеживает, насколько диск уже "оживлен" и, при повторном выполнении команды **start**, копирование продолжается с этой позиции.

Для поддержки согласованности тома пока одно или несколько его сплетений "оживляется", драйвер **vinum** записывает на поддиски, уже восстановленные до места записи. Он может также читать из сплетения, если считываемая область уже была восстановлена.

## ОСОБЕННОСТИ

Следующие особенности работы не связаны с ошибками, и у них есть весьма существенные причины, но, оказалось, что они вызывают непонимание. Каждая из этих особенностей уже была описана в соответствующем разделе выше.

1. Диски **vinum** являются разделами диска UNIX, и должны быть разделами типа **vinum**. Это отличается от драйвера **ccd**, которому необходимы разделы типа 4.2BSD. Такое поведение драйвера **ccd** провоцирует опасные ошибки (в оригинале - **is an invitation to shoot yourself in the foot**. Тому, кто придумает адекватный перевод - приз. *Прим. В.К.*): при использовании **ccd** можно легко перезаписать файловую систему. Утилита **vinum** этого не допустит.

По аналогичным причинам команда **vinum start** не позволит создать диск на разделе "**c**". Раздел "**c**" используется системой для представления всего диска, и должен быть типа **unused**. Тут есть явный конфликт, который утилита **vinum** решает, просто не используя раздел "**c**".

2. При создании тома с несколькими сплетениями утилита **vinum** не инициализирует сплетения автоматически. Это означает, что содержимое их неизвестно, но определенно не согласовано. В результате, по умолчанию **vinum** устанавливает для всех вновь созданных сплетений, кроме первого, состояние **faulty**. Чтобы синхронизировать их с первым сплетением, необходимо их запустить, что приводит к копированию данных со сплетения, находящегося в состоянии **up**. В зависимости от размера задействованных поддисков, это может потребовать много времени.

На практике пользователей не часто интересует, что было в сплетении при создании, и другие диспетчеры томов жульничают, просто устанавливая всем сплетениям состояние **up**. Утилита **vinum** предлагает два способа обеспечить перевод вновь созданных сплетений в состояние **up**:

- Создать сплетения, а затем синхронизировать их с помощью команды **vinum start**.
  - Создать том (не сплетение) с ключевым словом **setupstate**, которое требует от утилиты **vinum** игнорировать любые несогласованности и сразу переводить сплетения в состояние **up**.
3. Некоторые из команд, поддерживаемых в настоящее время утилитой **vinum**, фактически не нужны. По непонятным мне причинам оказывается, что пользователи часто пытаются использовать команды **label** и **resetconfig**, хотя, в частности, **resetconfig** выдает всевозможные страшные предупреждения. Не используйте эти команды без веских на то причин.
  4. Некоторые переходы из одного состояния в другое интуитивно непонятны. Фактически, не ясно, ошибка это или особенность. Если оказывается, что вы не можете запустить объект, находящийся в некотором странном состоянии, например, **reborn subdisk**, сначала попробуйте перевести его в состояние **stopped** с помощью команд **stop** или **stop -f**. Если это работает, вы затем сможете его запустить. Если окажется, что это единственный способ выйти из ситуации, и более простые методы не срабатывают, сообщите, пожалуйста, об этой ситуации.
  5. Если вы собираете модуль ядра с опцией **-DVINUMDEBUG**, надо утилиту **vinum** тоже собирать с опцией **-DVINUMDEBUG**, поскольку размер некоторых объектов данных, используемых обоими компонентами, зависит от этой опции. Если этого не сделать, команды будут завершаться с выдачей соответствующего сообщения об ошибке.
  6. Команда **vinum read** имеет особенно противный синтаксис. Когда-то она была единственным способом запуска **vinum**, но сейчас предпочтительнее использовать для этого команду **vinum start**. Команда **vinum read** должна использоваться только в целях сопровождения. Обратите внимание, что ее синтаксис изменился, а аргументами должны быть секции диска (slices), например, **/dev/da0**, а не разделы (partitions) вроде **/dev/da0e**.

## ФАЙЛЫ

**/dev/vinum**

каталог с файлами устройств для объектов **vinum**

**/dev/vinum/control**

управляющее устройство для **vinum**

**/dev/vinum/plex**

каталог, содержащий файлы устройств для сплетений **vinum**

**/dev/vinum/sd**

каталог, содержащий файлы устройств для поддисков **vinum**

## СРЕДА

**VINUM\_HISTORY**

Имя журнального файла, по умолчанию, - **/var/log/vinum\_history**.

**VINUM\_DATEFORMAT**

Формат дат в журнальном файле, по умолчанию, - **"%e %b %Y %H:%M:%S"**.

**EDITOR**

Имя редактора, который будет использоваться для редактирования файлов конфигурации, по умолчанию - **vi**.

## ССЫЛКИ

[\*\*strftime\(3\)\*\*](#), [\*\*vinum\(4\)\*\*](#), [\*\*disklabel\(8\)\*\*](#), [\*\*newfs\(8\)\*\*](#)

<http://www.vinumvm.org/vinum/>,

<http://www.vinumvm.org/vinum/how-to-debug.html>.

## АВТОРЫ

Грег Лихи (Greg Lehey) <[grog@lemis.com](mailto:grog@lemis.com)>

## ИСТОРИЯ

Утилита **vinum** впервые появилась в ОС FreeBSD 3.0. Компонент RAID-5 системы **vinum** был разработан для корпорации **Cybernet Inc.** ([www.cybernet.com](http://www.cybernet.com)), для ее продукта NetMAX.

## ОШИБКИ

Драйвер **vinum(4)** не использует подсистему **geom(4)**, поэтому тома **vinum(4)** нельзя использовать для средств, использующих GEOM, например, **gbde(8)**.

FreeBSD 4.9 5 мая 2003 года

Copyright (no c) - Fuck copyright! 2003-2004 В. Кравчук. [OpenXS Initiative](#), перевод на русский язык