

raid(4)

НАЗВАНИЕ

raid - драйвер диска RAIDframe

СИНТАКСИС

```
device raidframe
```

ОПИСАНИЕ

Драйвер **raid** обеспечивает поддержку RAID-массивов уровней 0, 1, 4 и 5 (и других!) в FreeBSD. Это руководство предполагает, что читатель хотя бы немного знаком с RAID-массивами и соответствующей терминологией. Также предполагается, что читатель знает, как конфигурировать диски и псевдоустройства в ядрах, как генерировать ядра и как разбивать диски на разделы.

Устройство RAIDframe поддерживает несколько уровней RAID, в том числе:

RAID 0

обеспечивает простое разбиение данных на полосы, распределенные по компонентам массива.

RAID 1

обеспечивает зеркалирование.

RAID 4

обеспечивает разбиение данных на полосы по компонентам, с сохранением информации о четности на выделенном диске (в данном случае, на последнем компоненте массива).

RAID 5

обеспечивает разбиение данных на полосы по компонентам, с распределением информации о четности по всем компонентам.

Драйвер RAIDframe поддерживает множество и других уровней RAID, включая *проверку четности* (Even-Odd parity), RAID уровня 5 с *rotated sparing*, *цепочной декластеризацией* (chained declustering) и *декластеризацией с чередованием* (interleaved declustering). Подробнее об этих различных конфигурациях RAID читатель может узнать в документации RAIDframe, указанной в разделе [ИСТОРИЯ](#).

В зависимости от сконфигурированного *уровня паритета* (parity level), драйвер устройства может обрабатывать сбои дисков-компонентов. Количество допустимых сбоев зависит от выбранного уровня паритета. Если драйвер может справиться со сбоем диска, и этот сбой происходит, то система работает в "деградированном режиме". В этом режиме все недостающие данные необходимо реконструировать на основе данных и информации о четности, представленных в других компонентах массива. Это приводит к замедлению доступа к данным, но означает, что сбой не вызывает полную остановку системы.

Драйвер RAID поддерживает и требует использования "меток компонентов" (component labels). *Метка компонента* содержит важную информацию о компоненте, включая указанный пользователем серийный номер, строку и столбец для этого компонента в наборе RAID, и являются ли данные (и информация о четности) на компоненте "чистыми". Если драйвер определяет, что метки существенно не согласованы друг с другом (например, не совпадают два или более серийных номера) или что метка компонента не соответствует его месту в наборе (например, метка компонента утверждает, что он является третьим в 6-дисковом наборе, но в RAID-массиве он является третьим в 5-дисковом наборе), то устройство не будет

сконфигурировано. Если драйвер определяет, что некорректной, видимо, является метка ровно одного компонента, а RAID-набор сконфигурирован как способный продолжать работу при сбое одного компонента, то RAID-набор будет сконфигурирован, не компонент с некорректной меткой будет помечен как сбойный (failed), и RAID-набор начнет работать в деградированном режиме. Если все компоненты согласованы между собой, RAID-набор будет нормально сконфигурирован.

Метки компонентов также используются для поддержки автоматического выявления и конфигурирования RAID-наборов. RAID-набор можно пометить как автоконфигурируемый, и тогда он будет конфигурироваться автоматически в процессе загрузки ядра. Файловые системы на RAID-наборе, который конфигурируется автоматически, также пригодны для использования в качестве корневой файловой системы. В настоящее время имеется только ограниченная поддержка (на архитектурах **alpha** и **pmax**) загрузки ядра непосредственно с набора RAID 1, а загрузка с любых других RAID-наборов вообще не поддерживается. Для использования RAID-набора в качестве корневой файловой системы ядро обычно получается с небольшого не-RAID раздела, после чего для корневой файловой системы можно использовать любой автоматически конфигурируемый RAID-набор. Дополнительную информацию об автоматическом конфигурировании RAID-наборов см. на странице справочного руководства **raidctl(8)**.

Драйвер поддерживает "горячее резервирование": подключенные диски, не используемые активно в существующей файловой системе. При сбое диска, драйвер может пересоздать сбойный диск на диске горячей замены или обратно на замененном диске. Если допускается горячая замена компонентов, можно удалить сбойный диск, установить новый вместо него, и скопировать данные обратно. Операция *обратного копирования* (sorbuck), как следует из ее названия, будет копировать реконструированные данные с диска горячего резервирования на ранее сбойный (а теперь - замененный) диск. Диски для горячего резервирования тоже можно добавлять на ходу с помощью **raidctl(8)**.

Если компонент не может быть выявлен при конфигурировании устройства RAID, этот компонент просто помечается как сбойный (failed).

Пользовательская утилита для выполнения всех действий по конфигурированию raid и других действий с набором - **raidctl(8)**. Самое главное, что утилиту **raidctl(8)** с опцией **-i** необходимо использовать для инициализации всех RAID-наборов. В частности, эта инициализация включает пересоздание данных о четности. Это пересоздание данных о четности также необходимо, когда новое устройство RAID запускается впервые или после не чистой остановки RAID-устройства. Указывая опцию **-P** утилиты **raidctl(8)** и выполняя это перевычисление всех данных о четности перед выполнением команд **fsck(8)** или **newfs(8)**, можно гарантировать целостность файловой системы и данных о четности. Стоит повторить, что перевычисленные данных о четности обязательно перед созданием или использованием любой файловой системы на устройстве RAID. Если данные четности некорректны, нельзя будет правильно восстановить недостающие данные.

Уровни RAID можно комбинировать иерархически. Например, устройство RAID 0 может состоять из нескольких устройств RAID 5 (которые, в свою очередь, могут состоять из физических дисков или любых других устройств RAID).

Для нормального функционирования RAID-устройства важно, чтобы диски были жестко привязаны к соответствующим адресам (т.е. не оставались в "свободном плавании", в результате которого диск с идентификатором SCSI ID 4 может оказаться устройством **/dev/da0c**). Это верно для всех типов дисков, включая IDE, SCSI и т.п. Для дисков IDE используйте опцию **ATAPI_STATIC_ID** в файле конфигурации ядра. Для SCSI подключать устройства следует в порядке, соответствующем их ID. Примеры применения этого подхода см. на странице справочного руководства **cam(4)**. Причина фиксации адресов устройств следующая: Рассмотрим систему с тремя SCSI-дисками, имеющими SCSI ID 4, 5 и 6, и соответствующими компонентам **/dev/da0e**, **/dev/da1e** и **/dev/da2e** набора типа RAID 5. Если диск с SCSI ID 5 сбой и система перегружается, старое устройство **/dev/da2e** станет доступно как **/dev/da1e**. Драйвер RAID может выявить, что позиции компонентов изменились, и не позволит нормально сконфигурировать набор. Если адреса устройств жестко "защиты", однако, драйвер RAID выявит, что недоступен средний компонент, и сможет запустить набор RAID 5 в деградированном режиме. Учтите, что коду автоматического выявления и

конфигурирования не важно, где находятся компоненты. Код автоконфигурирования правильно сконфигурирует устройство даже после перемещения любого количества компонентов.

Первый шаг при использовании драйвера **raid** - проверить, что он правильно сконфигурирован в ядре. Это делается с помощью строки следующего вида:

```
pseudo-device    raidframe    # Дискное устройство RAIDframe
```

в файле конфигурации ядра. Количество устройств указывать не нужно, поскольку драйвер автоматически создаст и сконфигурирует новые экземпляры устройств при необходимости. Для включения автоматического выявления и конфигурирования компонентов RAID-наборов, просто добавьте строку:

```
options          RAID_AUTOCONFIG
```

в файл конфигурации ядра.

Все разделы-компоненты должны быть типа **FS_BSDFFS** (например, 4.2BSD) или **FS_RAID**. Использование последнего типа настоятельно рекомендуется, и обязательно, если надо обеспечить автоматическое конфигурирование RAID-набора. Поскольку RAIDframe оставляет место для меток диска, компонентами RAID могут быть просто неформатированные диски (raw disks) или разделы, занимающие целый диск.

Более детальное описание фактического использования устройства **raid** представлено на странице справочного руководства **raidctl(8)**. Системному администратору настоятельно рекомендуется разобраться, какие шаги необходимо предпринять для пересоздания, копирования данных на замененный диск и перевычисления информации о четности, до того, как произойдет сбой компонента. Неправильные действия при сбое компонента могут привести к потере данных.

ПРЕДУПРЕЖДЕНИЯ

Некоторые уровни RAID (1, 4, 5, 6 и другие) могут защитить от потери данных при сбое одного компонента. Однако потеря двух компонентов системы RAID 4 или 5, либо потеря одного компонента системы RAID 0 приводит к потере всей файловой системы на этом RAID-устройстве. RAID - **не замена** продуманной стратегии резервного копирования.

Перевычисление четности **должно** выполняться в любой ситуации, когда есть шанс, что эти данные могут быть повреждены. К этим ситуациям относятся сбои системы или добавление ранее не использовавшегося RAID-устройства. Некорректная информация о четности приведет к катастрофе в случае любого сбоя компонента - лучше использовать RAID 0, и получить дополнительное место и выигрыш в скорости, чем хранить информацию о четности, но не обеспечивать ее корректность. Использование RAID 0 хотя бы не создает видимость повышенной защиты данных.

ФАЙЛЫ

/dev/raid*

специальные файлы устройств raid.

ССЫЛКИ

config(8), fsck(8), mount(8), newfs(8), raidctl(8)

ИСТОРИЯ

Драйвер **raid** в ОС FreeBSD - это портированный RAIDframe, механизм для быстрого прототипирования RAID-структур, разработанный сотрудниками Parallel Data Laboratory университета Карнеги-Мэллона (Carnegie Mellon University - CMU). RAIDframe, в том виде, как он исходно распространялся CMU, предоставляет RAID-симулятор для ряда различных архитектур, а также драйвер устройства пользовательского уровня и драйвер устройства ядра для Digital Unix. Драйвер **raid** - версия уровня ядра механизма RAIDframe v1.1, основанная на портированной Грэггом Остером (Greg Oster) в ОС NetBSD версии RAIDframe.

Более полное описание внутреннего устройства и возможностей RAIDframe можно найти в статье "**RAIDframe: A Rapid Prototyping Tool for RAID Systems**", by William V. Courtright II, Garth Gibson, Mark Holland, LeAnn Neal Reilly, and Jim Zelenka, опубликованной Parallel Data Laboratory университета Карнеги-Мэллона. Драйвер **raid** впервые появился в ОС FreeBSD 4.4.

АВТОРСКИЕ ПРАВА

Авторские права на RAIDframe (RAIDframe Copyright) следующие:

Copyright (c) 1994-1996 Carnegie-Mellon University.
All rights reserved.

Permission to use, copy, modify and distribute this software and its documentation is hereby granted, provided that both the copyright notice and this permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS "AS IS" CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Университет Carnegie Mellon требует от пользователей этого ПО пересылать по адресу:

Software Distribution Coordinator или Software.Distribution@CS.CMU.EDU
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3890

любые улучшения или расширения, которые они сделали, и предоставлять университету права на дальнейшее распространение этих изменений.

FreeBSD 4.9, 20 октября 2002 года

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