# The Software-RAID HOWTO

Jakob Østergaard (*jakob@ostenfeld.dk*) Переводчик: Максим Дзюманенко (*max@april.kiev.ua* ) Версия 0.90.7 19 Января 2000 г. Дата перевода: 11 Октября 2000 г.

Этот HOWTO описывает, как использовать программный RAID под Linux. Он связан с определенной версией уровня программного RAID, а именно уровнем 0.90 RAID, сделанным Ingo Molnar и другими. Этот уровень RAID будет стандартным в Linux-2.4, и эта версия также используется в ядрах Linux-2.2, поставляемых некоторыми поставщиками. Поддержка RAID 0.90 доступна в виде патчей к Linux-2.0 и Linux-2.2, и, как многие считают, намного более стабильна, чем старый RAID код в тех же ядрах.

## Содержание



## 1 Введение

Для описания старого уровня RAID, который стандартен для 2.0 и 2.2 ядер, смотрите великолепный HOWTO от Linas Vepstas (*linas@linas.org*), доступный из Linux Documentation Project на *linuxdoc.org*. Домашний сайт для этого HOWTO - *http://ostenfeld.dk/~jakob/Software-RAID.HOWTO/*, где изначально появляются обновленные версии. HOWTO написан Jakob Østergaard на основе большой переписки с Ingo Molnar *(mingo@chiara.csoma.elte.hu)*, одним из разработчиков RAID, почтового списка рассылки linux-raid *(linux-raid@vger.rutgers.edu)* и другими людьми.
Домашняя страница перевода - *http://dmv.webjump.com/HOWTOs/*. Обновленные версии, в первую очередь, появляются тут.
Причиной написания этого HOWTO, несмотря на существование Software-RAID-HOWTO, является то, что старый HOWTO описывает программный RAID старого стиля, в стандартных 2.0 и 2.2 ядрах. Этот HOWTO описывает использование RAID нового поколения, разработанного недавно. RAID нового поколения содержит много свойств, не представленных в старом RAID.
Если Вы хотите использовать новый RAID с 2.0 или 2.2 ядрами, Вы должны взять патч к вашему ядру, либо с *ftp://ftp.[your-country-code].kernel.org/pub/linux/daemons/raid/alpha*, либо, с недавних пор, с *http://people.redhat.com/mingo/*. Стандартные ядра 2.2 не содержат прямой поддержки нового RAID, описываемого в этом HOWTO. Для этого необходимы эти патчи. *Старый RAID код в 2.0 и 2.2 ядрах содержит ошибки и не реализует некоторых важных функций, реализованных в новом программном RAID.*
На момент написания, поддержка нового RAID объединена с ядрами серии 2.3, и, таким образом, будет (вполне вероятно) представлена в ядре Linux 2.4, как только оно выйдет. Но пока, стабильные ядра должны быть пропатчены вручную.

Вы можете использовать `-ac` выпуски ядра, сделанные Alan Cox -ом, для поддержки RAID в 2.2. *Часть* из них содержат RAID нового стиля, и это должно избавить Вас от необходимости патчить ядро.
Если Вы хорошо знакомы с RAID, часть информации в этом HOWTO покажется банальной. Просто пропустите ее.

### 1.1 Отречение

Обязательное отречение:
Хотя RAID кажется мне стабильным, и стабильным для многих других людей, у Вас он может не сработать. Если Вы потеряете все ваши данные, вашу работу, или это ударит по Вам - это не моя вина, и не вина разработчиков. Знайте, что вы используете программный RAID и эту информацию на свой риск! Никто не гарантирует, что либо программа, либо эта информация, сколько-нибудь корректна, либо пригодна вобще для использования. Сархивируйте все Ваши данные перед этими экспериментами. Лучше предостеречься, чем сожалеть.
Сказав это, я также должен сказать, что у меня не было проблем со стабильность программного RAID, я действительно без каких-либо проблем использую его на нескольких машинах, и я не видел, чтобы у других людей были проблемы с внезапными падениями или нестабильностью вызванной RAID-ом.

### 1.2 Требования

Этот HOWTO предполагает, что Вы используете последние 2.2.x или 2.0.x ядра с соответствующим raid0145 патчем и raidtools версии 0.90, или Вы используете последнее ядро серии 2.3 (версию $>$ 2.3.46) или, со временем, 2.4. Оба патча и утилиты можно найти на *ftp://ftp.fi.kernel.org/pub/linux/daemons/raid/alpha*, и в некоторых случаях на *http://people.redhat.com/mingo/*. Патч RAID, пакет raidtools и ядро должны, по мере возможности, соответствовать друг другу. Иногда необходимо использовать более старые ядра, если патчи raid не доступны для последнего ядра.

## 2 Почему RAID ?

Есть много преимуществ в использовании RAID. Некоторые из них: возможность комбинировать несколько физических дисков в один большой "виртуальный" диск, увеличение производительности и надежности.

### 2.1 Технические детали

Linux RAID может работать на большинстве устройств. Не имеет значения используете Вы IDE или SCSI диски, или и те и другие. Некоторые люди также более или менее успешно использовали Сетевое блочное Устройство (Network Block Device (NBD)).
Удостоверьтесь, что шины к дискам достаточно быстры. Вы не должны вешать 14 UW-SCSI дисков на одну UW шину, если каждое устройство может дать 10 Мб/с, а шина может только 40 Мб/с. Также, вы должны держать только одно устройство на IDE шине. Работа дисков master/slave ужасна по производительности. IDE очень плох при подключении более одного диска на шину. Конечно, все новые материнские платы содержат две IDE шины, так что Вы можете установить два диска в RAID без покупки дополнительных контроллеров.
Уровень RAID не имеет абсолютно ничего общего с уровнем файловой системы. Вы можете держать любую файловую систему на устройстве RAID, как и на любом другом блочном устройстве.

## 2.2 Термины

Слово "RAID" означает "Программный Linux RAID". Этот HOWTO не рассматривает аспекты аппаратных RAID.
При описании установки, полезно сверить число дисков и их размеры. Каждый раз буква **N** используется для указания количества активных дисков в массиве (не считая резервных дисков). Буква **S**, если не указано обратное, - размер наименьшего устройства в массиве. Буква **P** используется как производительность одного диска в массиве в Мб/с. Мы предполагаем, что диски одинаково быстрые, что может быть не всегда справедливо.
Заметьте, что слова "устройство" и "диск" означают одно и то же. Обычно устройства, используемые для построения RAID, являются разделами диска, не обязательно целыми дисками. Но объединение нескольких разделов на одном диске обычно бессмысленно, таким образом устройства и диски обозначают просто "разделы на различных дисках".

## 2.3 Уровни RAID

Здесь приводится короткое описание того, что поддерживается патчами Linux RAID. Часть из этой информации - чисто базовая информация о RAID, но я добавил несколько замечаний о особенностях реализации уровней в Linux. Если Вы знакомы с RAID, просто пропустите эту секцию. Позже, если возникнут проблемы, можете вернуться к ней :)
Текущие RAID патчи для Linux поддерживают следующие уровни:

- **Линейный режим**
  - Два или более диска объединяются в одно устройство. Диски "добавляются" один к другому, таким образом, запись на устройство RAID будет заполнять сначала диск 0, затем диск 1 и так далее. Диски не обязательно должны быть одного размера. Фактически, размер здесь вообще не имеет значения :)
  - На этом уровне нет избыточности. Если один диск отказывает, Вы, скорее всего, потеряете все Ваши данные. Однако, Вы, возможно, сможете удачно восстановить часть данных, так как в файловой системе будет просто отсутствовать один большой последовательный кусок данных.
  - Производительность чтения и записи не увеличивается для одиночных операциях считывания/записи. Но если несколько пользователей используют устройство, Вам может повезти, и один один пользователь может фактически использовать первый диск, а другой пользователь обращаться к файлам на втором диске. Если это произойдет, вы получите прирост производительности.
- **RAID-0**
  - Также называемый режим "stripe". Подобен линейному режиму, исключая то, что чтение и запись производятся параллельно с двух устройств. Устройства должны иметь приблизительно один размер. Так как весь доступ производится параллельно, устройства заполняются одинаково. Если одно устройство больше, чем другие, это дополнительное пространство все еще используется в RAID устройстве, но при записи в самом конце вашего RAID устройства, Вы получаете доступ только к этому одному диску, что, конечно, снижает производительность.
  - Как и в линейном режиме, на этом уровне нет никакой избыточности. В отличии от линейного режима, Вы не сможете восстановить никаких данных при отказе диска. Если Вы удаляете диск из RAID-0 набора, в RAID устройстве будет не просто отсутствовать последовательный кусок данных, оно будет заполнено маленькими дырочками по всему устройству. e2fsck будет не в состоянии восстановить большую данных на этом устройстве.

  - Производительность чтения и записи увеличится, так как чтение и запись будут выполняться параллельно на дисках. Обычно, это главная причина использования этого уровня RAID. Если шины к дискам достаточно быстрые, Вы сможете получить почти N*P Мб/сек.

- **RAID-1**

  - Это первый режим, который реализует избыточность. RAID-1 может использоваться на двух или более дисках с нулем или более резервными дисками. Этот режим поддерживает точную копию информации одного диска на всех дисках. Конечно, диски должны быть одного размера. Если один из дисков больше другого, Ваш RAID будет размером с наименьший.
  - Если N-1 диск удален (или отказал), все данные все еще целы. Если имеются резервные диски, и если система (SCSI драйвера или IDE чипсет и т.п.) пережили отказ, после обнаружения отказа, начинается немедленная реконструкция зеркала на резервные диски.
  - Производительность записи немного хуже, чем у одного диска, так как на каждый диск массива должны быть посланы идентичные копии записанных данных. Производительность чтения *обычно* достаточно плохая из-за чрезмерного упрощения стратегии балансировки чтения в коде RAID. Однако, реализована более улучшеная стратегия балансировки чтения, которая может быть доступна для патчей RAID для Linux-2.2 (спросите в linux-kernel списке рассылки), и которая будет, по всей вероятности, в стандартной поддержке RAID в 2.4 ядре.

- **RAID-4**

  - Этот уровень RAID не часто используется. Он может быть использован с тремя или более дисками. Вместо полной зеркализации информации, он сохраняет информацию о четности на отдельном диске, и записывает данные на другой диск подобным используемому в RAID-0 образом. Так как один диск зарезервирован для информации четности, размер массива будет (N-1)*S, где S - размер наименьшего устройства в массиве. Как и в RAID-1, диски должны быть либо одного размера, либо S, в формуле (N-1)*S, должно быть размером наименьшего диска в массиве.
  - Если один диск откажет, информация о четности может быть использована для восстановления всех данных. Если два диска откажет - все данные будут потеряны.
  - Причина нечастого использования этого уровня - информация о паритете хранится на одном диске. Эта информация должна быть обновлена *каждый* раз когда ведется запись на один из других дисков. Таким образом, диск с паритетом становится бутылочным горлышком, если он не намного быстрее остальных дисков. Однако, если так случилось, что у Вас много медленных дисков и один очнь быстрый - этот уровень RAID может быть очень полезен.

- **RAID-5**

  - Это, пожалуй, самый полезный режим RAID для тех, кто хочет соединить несколько физических дисков, и к тому же сохранить избыточность. RAID-5 может быть использован на трех или более дисках, с нулем или более резервных дисков. Размер результирующего RAID-5 устройства будет (N-1)*S, как и в RAID-4. Главное отличие между RAID -5 и -4 в том, что распределением информации о паритете по всем устройствам, избегается проблема бутылочного горлышка в RAID-4.
  - Если один из этих дисков отказывает, все данные все еще не повреждены, благодаря информации о паритете. Если имеются резервные диски, при отказе диска немедленно начинается реконструкция. Если отказывает два диска одновременно - все данные потеряны. RAID-5 может пережить отказ одного диска, но не двух или более.
  - Обычно увеличивается производительность как чтения, так и записи, но тяжело предсказать насколько.

### 2.3.1 Резервные диски

Резервные диски - диски, которые не являются частью RAID тома, пока один из активных дисков откажет. Когда обнаруживается отказ диска, он маркируется как "плохой" и, если имеются резервные диски, немедлено начинается реконструкция.
Таким образом, резервные диски добавляют дополнительную безопасность, особенно к RAID-5 системам, где, возможно, тяжело достичь этого (физически). Это позволяет работать системе некоторое время, с отказавшим диском, так как вся избыточность полагается на наличие резервных дисков.
Вы не можете быть уверены, что Ваша система переживет отказ диска. Драйвер RAID уровня должен обрабатывать дисковые отказы очень хорошо, но SCSI драйвера могут не правильно обрабатывать ошибки, или IDE чипсет может заблокироваться, или может случиться много всякого другого.

## 2.4 Виртуальная память на RAID

Нет причин использовать RAID для увеличения производительности виртуальной памяти. Ядро само может распределять подкачку на несколько дисков, если Вы укажете одинаковый приоритет им в fstab файле.
Правильный fstab выглядит так:

```
/dev/sda2       swap            swap    defaults,pri=1   0 0
/dev/sdb2       swap            swap    defaults,pri=1   0 0
/dev/sdc2       swap            swap    defaults,pri=1   0 0
/dev/sdd2       swap            swap    defaults,pri=1   0 0
/dev/sde2       swap            swap    defaults,pri=1   0 0
/dev/sdf2       swap            swap    defaults,pri=1   0 0
/dev/sdg2       swap            swap    defaults,pri=1   0 0
```

Такая конфигурация позволяет делать подкачку параллельно на несколько SCSI дисков. RAID не нужен, так как это было свойством ядра уже давно.
Другая причина использовать RAID для подкачки - высокая готовность. Если Вы установили загрузку системы и т.д. с RAID-1 устройства, система должна пережить отказ диска. Но если система выполняет подкачку с уже отказавшего устройства, будьте уверены - она рухнет. Подкачка на RAID-1 устройстве решит эту проблему.
Было много дискуссий о стабильности подкачки на RAID устройстве. Дебаты продолжаются, так как это сильно зависит от других аспектов ядра. Что касается этого документа, кажется подкачка на RAID должна быть вполне стабильна, *исключая* время реконструкции массива (т.е. поле того, как вставлен новый диск в деградировавший массив). Когда выйдет 2.4 это решение будет одним из наиболее более быстрых, но тогда, Вы должны жестко протестировать систему, пока сами не будете удовлетворены стабильностью или решите, что Вы не будете использовать подкачку на RAID.
В можете установить подкачку в файл на файловой системе RAID устройства, или Вы можете установить RAID устройство как swap раздел, на Ваше усмотрение. Как обычно, RAID устройство - просто блочное устройство.

# 3 Аппаратные решения

Эта секция имеет отношение к некоторым аппаратным особенностям запуска программного RAID.

## 3.1 Конфигурирование IDE

В самом деле возможно запустить RAID на IDE дисках. Также может быть достигнута превосходная производительность. Фактически, сегодняшние цены на IDE устройства и контроллеры делают IDE заслуживающим раздумий, при создании новой RAID системы.

- **Физическая стабильность:** IDE устройства традиционно более низкого качества, чем SCSI устройства. Даже сейчас, гарантия на IDE устройства - типично один год, когда на SCSI часто три или пять. Однако неправильно будет сказать, что IDE диски по определению плохо сделаны, Вы должны осознавать, что IDE диски *некоторых* производителей *могут* отказывать более часто, чем подобные SCSI диски. Однако, другие производители используют одинаковые механические части для обоих, SCSI и IDE устройств. Это все сводится к одному: Все диски отказывают, рано или поздно, и вы должны быть готовы к этому.

- **Целостность данных:** Раньше, в IDE не было способа гарантировать, что данные посланные по IDE шине будут данными, фактически записанными на диск. Это потому, что не хватало проверки четности, проверки контрольных сумм, и т.п. Со стандартом Ultra-DMA, IDE теперь могут делать проверку контрольных сумм данных, которые они получили, и таким образом стало очень неправдоподобно исказить данные.

- **Производительность:** Я не собираюсь детально обсуждать здесь производительность IDE. Коротко, история такова:

  - IDE устройства быстрые (12 Мб/с и выше)
  - IDE создают большую нагрузку на CPU, чем SCSI (но кого это интересует?)
  - Используйте **один** IDE диск на IDE шине, slave диски портят производительность

- **Отказоустойчивость:** ДрайверIDE обычно устойчив и переносит отказ IDE диска. RAID уровень пометит диск как отказавший, и если Вы работаете с RAID уровня 1 или выше, машина должна хорошо работать, до тех пор пока Вы ее не остановите на ремонт.

**Очень** важно, чтобы Вы использовали **один** IDE диск на IDE шине. Два диска не только разрушают производительность, но отказ диска - часто гарантирует отказ шины, и, таким образом, отказывают все устройства на этой шине. При отказоустойчивом RAID (RAID уровней 1,4,5), отказ одного может быть обработан, но отказ двух дисков (два диска на одной шине отказывают из-за отказа одного диска) приведут массив в неиспользуемое состояние. Также, при отказе master диска на шине, slave или IDE контроллер может быть сбит с толку. Одна шина - один диск, это правило.
Существуют дешевые PCI IDE контроллеры. Вы часто можете получить две или четыре шины за $80. Учитывая значительно более низкую цену IDE по сравнению со SCSI дисками, я бы сказал, что IDE дисковые массивы могут быть в самом деле хорошим решением, если оно реализуется относительно небольшим (возможно около 8) количеством дисков, которые можно подключить к типичной системе (если, конечно, у Вас достаточно PCI слотов для этих IDE контроллеров).
У IDE есть небольшая проблема с кабелями при применении в больших массивах. Даже если У вас достаточно PCI слотов, маловероятно, что Вы сможете поставить более 8 дисков в систему и запустить ее без искажений данных при передаче (из-за слишком длинных IDE кабелей).

## 3.2 Горячая замена

Некоторое время это была горячая тема списка рассылки linux-kernel. Хотя горячая замена дисков поддерживается в некоторой степени, она все еще нечто не очень простое.

### 3.2.1 Диски IDE с горячей заменой

**Не делайте !** IDE совсем не работает с горячей заменой. Конечно, у вас может работать, если ваш IDE драйвер скомпилирован как модуль (возможно только в 2.2 серии ядер), и вы пере-загружаете его после замены диска. Но Вы можете запросто сжечь IDE контроллер, и Вы будете блокированы на значительно большее время, чем, если бы Вы заменили диск на выключенной машине.
Главная проблема, исключая выбросы электричества, которые могут разрушить вашу аппаратуру, - шина IDE должны быть пересканирована после замены дисков. Текущий IDE драйвер так не умеет. Если новый диск на 100% идентичен старому (геометрией и т.п.), он *может* работать без пере-сканирования шины, но на самом деле, Вы ходите по краю обрыва.

### 3.2.2 Диски SCSI с горячей заменой

Обычная SCSI аппаратура не позволяет горячей замены. Однако, это **может** работать. Если Ваш SCSI драйвер поддерживает пере-сканирование шины, и удаление и добавление устройств, Вы можете делать горячую замену. Однако, на обычной SCSI шине Вы не должны отключать устройства, пока система включена. Но опять же, это может сработать (и вы можете закончить сгоревшей аппаратурой).
Уровень SCSI **должен** пережить смерть диска, но пока не все драйвера SCSI могут так делать. Если Ваш SCSI драйвер падает при отказе диска, ваша система упадет с ним, и горячая замена реально не возможна.

### 3.2.3 Горячая замена с SCA

Со SCA, возможна горячая замена устройств. Однако, у меня нет аппаратуры на которой можно это попробовать, и я не слышал, чтобы кто-то пробовал, так что реально я не могу дать никаких рекомендаций, как это сделать.
Если Вы хотите поиграть с этим, Вы, в любом случае, должны знать о внутренностях SCSI и RAID. Таким образом я не собираюсь писать здесь что-либо, что я не проверил, вместо этого я могу дать несколько нитей к размышлению:

- grep на предмет **remove-single-device** в **linux/drivers/scsi/scsi.c**
- Взгляните на **raidhotremove** и **raidhotadd**

Не все SCSI драйвера поддерживают добавление и удаление устройств. В серии 2.2 ядер, по крайней мере Adaptec 2940 и Symbios NCR53c8xx драйвера, кажется, поддерживаю это, другие - не известно. Я буду благодарен, если кто-то даст сюда дополнительные факты ...

# 4 Установка RAID

## 4.1 Общие установки

Вот что Вам нужно для любых уровней RAID:

- Ядро. Предпочтительно стабильной серии 2.2.X, или последнее 2.0.X. (Если 2.4 уже вышло, в то время когда Вы это читаете, используйте его)
- Патчи RAID. Это обычно патч для последних ядер. (Если Вы найдете 2.4 ядро, патчи уже в нем, и вы можете забыт о них)
- RAID утилиты.
- Терпение, пицца, и Ваш любимый кофейный напиток.

Все программы могут быть найдены на `ftp://ftp.fi.kernel.org/pub/linux` RAID утилиты и патчи в `daemons/raid/alpha` подкаталоге. Ядра - в `kernel` подкаталоге.
Пропатчите ядро, сконфигурируйте его для включения поддержки желаемого уровня RAID. Скомпилируйте его и установите.
Затем распакуйте, сконфигурируйте, скомпилируйте и установите утилиты RAID.
Так, все хорошо. Если Вы сейчас перезагрузитесь, Вы должны получить файл называемый `/proc/mdstat`. Запомните его, этот файл - Ваш друг. Сделав `cat /proc/mdstat` посмотрите его содержимое. Это должно Вам сказать, что у Вас зарегистрированы правильные свойства RAID (RAID режим), и устройства RAID уже активны.
Создайте разделы, которые хотите включить в RAID набор.
Сейчас, рассмотрим специфику режимов.

## 4.2 Линейный режим

Итак, у вас есть два или более раздела, не обязательно одного размера (но конечно могут быть), которые Вы хотите объединить вместе.
Установите /etc/raidtab файл для описания вашей конфигурации. Я устанавливаю raidtab для двух дисков в линейный режим, и файл выглядит так:

```
raiddev /dev/md0
        raid-level      linear
        nr-raid-disks   2
        chunk-size      32
        persistent-superblock 1
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Резервные диски тут не поддерживаются. Если диск умрет, массив умрет вместе с ним. Не существует информации для помещения на резервный диск.
Вы, возможно удивлены тем, что мы указали здесь chunk-size (размер куска), при том, что линейный режим просто объединяет два диска в один большой без параллелизма. Да, Вы полностью правы, это - лишнее. Просто поставьте какой-то размер куска и не беспокойтесь об этом.
Итак, создадим массив. Запускаем команду

```
mkraid /dev/md0
```

Это должно инициализировать ваш массив, записать отдельные суперблоки, и запустить массив.
Загляните в /proc/mdstat. Вы должны заметить, что массив уже запущен.
Теперь, ВЫ можете создать файловую систему, просто как и на любом другом устройстве, смонтируйте ее, включите ее в Ваш fstab и тому подобное.

## 4.3 RAID-0

У вас должно быть два или более устройств, приблизительно одного размера, и Вы хотите объединить их емкость и также производительность путем параллельного доступа.
Установите файл /etc/raidtab для описания Вашей конфигурации. Пример raidtab выглядит таким образом:

```
raiddev /dev/md0
        raid-level      0
        nr-raid-disks   2
        persistent-superblock 1
        chunk-size     4
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

В линейном режиме, резервные дики не поддерживаются. RAID-0 не имеет избыточности, так что если диск умрет, массив умрет вместе с ним.
Еще раз, просто запустите

```
mkraid /dev/md0
```

для инициализации массива. Это должно инициализировать суперблок и запустить устройство. Загляните в /proc/mdstat чтобы посмотреть, что произошло. Вы должны увидеть, что Ваше устройство запущено.
/dev/md0 теперь готов к форматированию, монтированию, использованию и издевательствам.

## 4.4 RAID-1

У вас есть два устройства приблизительно одного размера, и Вы хотите из зеркализировать. В конце концов, Вы можете использовать больше устройств, которые Вы можете держать как резервные диски, и которые автоматически станут частью зеркала, если одно из из активных устройств сломается.
Установите `/etc/raidtab` файл подобно этому:

```
raiddev /dev/md0
        raid-level      1
        nr-raid-disks   2
        nr-spare-disks  0
        chunk-size     4
        persistent-superblock 1
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Если у Вас есть резервные диски, Вы можете добавить их в конец спецификации устройства

```
        device          /dev/sdd5
        spare-disk      0
```

Не забудьте установить соответственно nr-spare-disks запись (количество резервных дисков).
Итак, мы все установили для запуска инициализации RAID. Зеркало должно быть сконструировано, содержимое ( сейчас это, однако, не важно, так как устройство все еще не форматировано) двух дисков должно быть синхронизировано.
Подаем команду

```
mkraid /dev/md0
```

для начала инициализации зеркала.
Проверьте `/proc/mdstat` файл. Он должен сказать вам, что устройство /dev/md0 было запущено, зеркало начало реконструироваться, а также оценочное время завершения реконструкции.
Реконструкция делается в периоды отсутствия ввода-вывода. Так что, ваша система должна быть еще достаточно отзывчива, хотя Ваш индикатор дисковой активности должен хорошо светиться.
Процесс реконструкции прозрачен, так что Вы можете, фактически, использовать зеркало несмотря на реконструкцию.
Попробуйте форматировать устройство, при запущенной реконструкции. Это должно работать. Также Вы можете смонтировать его и использовать в процессе реконструкции. Конечно, если неисправный диск разрушается при реконструкции, Вам не повезло.

## 4.5 RAID-4

**Заметьте!** Я сам не тестировал эту конфигурацию. Конфигурация ниже - моя догадка, а не что-то, фактически запущенное мною.
У вас есть три или более приблизительно одного размера диска, один значительно быстрее других, и Вы хотите скомбинировать их все в одно большое устройство, которое все еще содержит немного избыточной информации. В конце концов у вас есть несколько устройств, которые Вы хотите использовать как резервные диски.
Установите файл /etc/raidtab подобно этому:

```
raiddev /dev/md0
        raid-level      4
        nr-raid-disks   4
```

```
        nr-spare-disks  0
        persistent-superblock 1
        chunk-size      32
        device          /dev/sdb1
        raid-disk       0
        device          /dev/sdc1
        raid-disk       1
        device          /dev/sdd1
        raid-disk       2
        device          /dev/sde1
        raid-disk       3
```

Если у Вас резервные диски, они должны быть вставлены аналогичным образом, следуя спецификациям raid-disk;

```
        device         /dev/sdf1
        spare-disk     0
```

Как обычно, ваш массив может быть инициализирован командой

```
   mkraid /dev/md0
```

Перед форматированием, Вы должны просмотреть секцию специальных опций mke2fs.

## 4.6 RAID-5

У Вас есть три или более дисков приблизительно одного размера, Вы хотите скомбинировать их в большое устройство, но еще содержащее некоторую степень избыточности. В конце концов у Вас есть несколько дисков для использования как резервных, которые не будут частью массива до отказа другого устройства.
Если Вы используете N дисков, где S - размер наименьшего, размер всего массива будет (N-1)*S. Это "не включает" пространство используемое для информации о четности (избыточности). Итак, если любой диск отказывает, все данные остаются целыми. Но, если два диска отказывают, все данные потеряны.
Установите файл /etc/raidtab так:

```
raiddev /dev/md0
        raid-level      5
        nr-raid-disks   7
        nr-spare-disks  0
        persistent-superblock 1
        parity-algorithm        left-symmetric
        chunk-size      32
        device          /dev/sda3
        raid-disk       0
        device          /dev/sdb1
        raid-disk       1
        device          /dev/sdc1
        raid-disk       2
        device          /dev/sdd1
        raid-disk       3
        device          /dev/sde1
        raid-disk       4
        device          /dev/sdf1
        raid-disk       5
        device          /dev/sdg1
        raid-disk       6
```

Если у Вас есть резервные диски, они должны быть вставлены подобным образом, следуя спецификациям raid-disk;

```
device          /dev/sdh1
spare-disk      0
```

И так далее.
Размер куска в 32 KB хорошее начальное значение для многих общих применений файловой системы. Массив, на котором используется вышеуказанный raidtab, - устройство размером 7 раз по 6 GB = 36 GB (запомните (n-1)*s = (7-1)*6 = 36) Оно содержит файловую систему ext2 с размером блока 4 Кб. Если Ваша файловая система намного больше или Вы храните очнь большие фалы, Вы должны установить больший размер куска и размер блока файловой системы.
Итак, хватит разговоров. Вы установили raidtab, так что посмотрим, работает ли он. Подаем команду

```
mkraid /dev/md0
```

и смотрим, что получилось. Надеюсь Ваши диски заработали как сумасшедшие, так как начался процесс реконструкции Вашего массива. Загляните в `/proc/mdstat` чтобы посмотреть что происходит.
Если устройство успешно создано, начался процесс реконструкции. Ваш массив не устойчив, пока фаза реконструкции не завершена. Однако, массив полностью функционален (кроме, конечно, обработки дисковых отказов), и Вы можете его форматировать и использовать, пока он реконструируется.
Перед форматированием массива, посмотрите секцию специальных опций mke2fs.
Итак, сейчас вы запустили свое RAID устройство, Вы можете всегда остановить его или снова запустить используя

```
raidstop /dev/md0
```

или

```
raidstart /dev/md0
```

команды.
Вместо помещения этого в init-файлы и многократных перезагрузок чтобы заставить это работать, читайте далее, и запустите авто-детектирование.

## 4.7 Отдельный суперблок

Вернемся в "Старые Добрые Времена" (TM), raidtools читали Ваш read `/etc/raidtab` файл и затем инициализировали массив. Однако, это требовало наличия файловой системы та том, на чем был смонтирован `/etc/raidtab`. Это не подходило для загрузки с RAID.
Также, старый подход приводил к сложностям при монтировании файловой системы на RAID устройствах. Они не должны были, как обычно, вставляться в `/etc/fstab` файл, но должны были монтироваться из скриптов инициализации.
Отдельный суперблок решил эти проблемы. Когда массив инициализируется с опцией `persistent-superblock` в файле `/etc/raidtab`, в начале всех дисков массива записывается специальный суперблок. Это позволяет ядру читать конфигурацию устройств RAID прямо с затрагиваемых дисков, вместо чтения конфигурационного файла, который может не всегда доступен.
Однако Вы должны поддерживать целостность файла `/etc/raidtab`, так как Вам он может понадобиться позже при реконструкции массива.
Если вы хотите автоматического детектирования RAID устройств при загрузке - отдельный суперблок обязателен. Это описано в секции **Автоматическое детектирование**.

## 4.8 Размер кусков

Размер куска заслуживает обьяснения. Вы можете никогда не писать полностью параллельно на дисковый набор. Если у Вас два диска и Вы хотите записать байт, Вы должны, фактически, записать четыре бита на каждый диск, каждый второй бит должен пойти на диск 0 а другие на диск 1. Аппаратно это не поддерживается. Вместо этого, мы выбираем некоторый размер куска, который мы определяем как наименьшую "атомарную" порцию данных, которые могут быть записаны на диски. Запись 16 Кб с размером куска в 4 Кб, приведет к записи первого и третьего 4 Кбайтных кусочков на первый диск, а второго и четвертого на второй, в случае RAID-0 из двух дисков. Таким образом, для длинных записей, вы можете увидеть меньшие накладные расходы при довольно больших размерах кусков, в то время как массивы, которые в основном содержит небольшие файлы, имеют преимущество при небольших размерах куска.
Размеры куска должны быть указаны для всех уровней RAID, включая линейный режим. Однако, размер куска безразличен для линейного режима.
Для оптимальной производительности, Вы должны поэкспериментировать с этим значением, также как и с размером блока файловой системы, которую вы создаете в массиве.
Аргумент опции chunk-size в `/etc/raidtab` указывает размер кусочка в килобайтах. Так "4" означает "4 Кб".

### 4.8.1 RAID-0

Данные записываются "почти" в параллельном режиме на диски массива. Фактически, `chunk-size` байт записываются на каждый диск последовательно.
Если Вы указываете размер куска в 4 Кб, и пишете 16 Кб на массив из трех дисков, RAID система будет писать 4 Кб на диски 0, 1 и 2, параллельно, а оставшиеся 4 Кб на диск 0.
Размер куска в 32 KB - разумное начальное значение для большинства массивов. Но оптимальное занчение сильно зависит от количества в дисков, содержимого файловой системы на массиве, и многих других факторов. Поэкспериментируйте с этим, для получения производительности.

### 4.8.2 RAID-1

Для записи, размер куска не влияет, так как все данные, в любом случае, должны быть записаны на все диски. Однако для чтения, размер куска указывает сколько данных читаются последовательно с участвующих дисков. Так как все диски массива содержат одинаковую информацию, чтение может быть сделано параллельно, подобным RAID-0 образом.

### 4.8.3 RAID-4

Когда сделана запись на массив RAID-4, также должна быть обновлена информация о паритете на паритетном диске. Размер куска - размер паритетных блоков. Если байт записывается на массив RAID-4, потом `chunk-size` байт считываются с N-1 дисков, вычисляется информация о паритете, и `chunk-size` байт записываются на паритетный диск.
Размер куска также влияет на производительность чтения также как и в RAID-0, так как считывания с RAID-4 делаются аналогично.

### 4.8.4 RAID-5

На RAID-5 размер куска имеет такое же занчение как и в RAID-4.
Разумный размер куска для RAID-5 массива - 128 KB, но как обычно, Вы можете поэкспериментировать с ним.
Посмотрите далее секцию специальных опций mke2fs. Это влияет на производительность RAID-5.

## 4.9 Опции mke2fs

Существует специальная опция форматирования RAID-4 ил -5 устройств с mke2fs. Опция `-R stride=nn` позволяет mke2fs лучше размещать различные ext2 специфичные структуры данных разумным способом на устройство RAID.
Если размер куска 32 Кб, это значит, что 32 Кб последовательных данных будут лежать на одном диске. Если Вы хотите создать ext2 файловую систему с размером блока в 4 Кб, Вы сделаете так, что будет восемь блоков файловой системы в одном куске. Мы можем указать эту информацию для утилиты mke2fs, при создании файловой системы:

```
mke2fs -b 4096 -R stride=8 /dev/md0
```

Производительность RAID-{4,5} строго зависит от этой опции. Я не уверен как опция stride будет воздействовать на другие уровни RAID. Если у кого-то есть эта информация, пожалуйста пошлите ее мне.
Размер блока ext2fs *строго* определяет производительность файловой системы. Вы всегда должны использовать размер блока 4Кб на любой файловой системе более чем нескольких сот мегабайт, если Вы не помещаете очень большое число маленьких файлов на нее.

## 4.10 Авто-детектирование

Авто-детектирование позволяет ядру автоматически распознавать устройства RAID при загрузке, сразу после завершения обычного детектирования разделов.
Для этого требуется несколько вещей:

1. Вам нужна поддержка авто-детектирования в ядре. Проверьте это.
2. Вы должны создать RAID устройства используя отдельный суперблок
3. Тип раздела устройств используемых в RAID должен быть установлен в **0xFD** (запустите fdisk и установите тип в "fd")

ЗАМЕТКА: Удостоверьтесь, что Ваш RAID НЕ ЗАПУЩЕН перед сменой типа раздела. Используйте `raidstop /dev/md0` для остановки устройства.
Если Вы сделаете 1, 2 и 3 как указано выше, авто-детектирование должно быть установлено. Попробуйте перезагрузиться. После загрузки системы, сделайте cat `/proc/mdstat` и это должно показать, что Ваш RAID запущен.
При загрузке, Вы должны увидеть сообщения подобные этим:

```
Oct 22 00:51:59 malthe kernel: SCSI device sdg: hdwr sector= 512
 bytes. Sectors= 12657717 [6180 MB] [6.2 GB]
Oct 22 00:51:59 malthe kernel: Partition check:
Oct 22 00:51:59 malthe kernel:  sda: sda1 sda2 sda3 sda4
Oct 22 00:51:59 malthe kernel:  sdb: sdb1 sdb2
Oct 22 00:51:59 malthe kernel:  sdc: sdc1 sdc2
Oct 22 00:51:59 malthe kernel:  sdd: sdd1 sdd2
Oct 22 00:51:59 malthe kernel:  sde: sde1 sde2
Oct 22 00:51:59 malthe kernel:  sdf: sdf1 sdf2
Oct 22 00:51:59 malthe kernel:  sdg: sdg1 sdg2
Oct 22 00:51:59 malthe kernel: autodetecting RAID arrays
Oct 22 00:51:59 malthe kernel: (read) sdb1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdb1,1>
Oct 22 00:51:59 malthe kernel: (read) sdc1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdc1,2>
Oct 22 00:51:59 malthe kernel: (read) sdd1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdd1,3>
```

```
Oct 22 00:51:59 malthe kernel: (read) sde1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sde1,4>
Oct 22 00:51:59 malthe kernel: (read) sdf1's sb offset: 6205376
Oct 22 00:51:59 malthe kernel: bind<sdf1,5>
Oct 22 00:51:59 malthe kernel: (read) sdg1's sb offset: 6205376
Oct 22 00:51:59 malthe kernel: bind<sdg1,6>
Oct 22 00:51:59 malthe kernel: autorunning md0
Oct 22 00:51:59 malthe kernel: running: <sdg1><sdf1><sde1><sdd1><sdc1><sdb1>
Oct 22 00:51:59 malthe kernel: now!
Oct 22 00:51:59 malthe kernel: md: md0: raid array is not clean --
 starting background reconstruction
```

Это отрывок при авто-детектировании массива RAID-5, который не был чисто остановлен (например при крахе машины). Была автоматически инициирована реконструкция. Монтирование этого устройства вполне безопасно, так как реконструкция прозрачна и все данные целы ( только паритетная информация противоречива - но она не нужна, пока диск не откажет).
Автоматически стартующие устройства также автоматически останавливаются при выключении. Не беспокойтесь о init скриптах. Просто используйте устройства /dev/md как любые другие /dev/sd или /dev/hd устройства.
Да, это в самом деле очень просто.
Вы можете взглянуть в Ваш init-scripts для любых raidstart/raidstop комманд. Они часто есть в стандартных RedHat init скриптах. Они используются для RAID старого стиля, и не используются в RAID нового стиля с авто-детектированием. Посто удалите строки, и все будет очень просто.

## 4.11 Загрузка с RAID

Существует несколько путей для установки системы, которая монтирует свою корневую файловую систему на устройство RAID. На текущий момент, только графический инсталлятор RedHat Linux 6.1 позволяет прямую установку на устройство RAID. Так что вполне вероятно Вам прийдется немного повозиться, если это Вам нужно, но это вполне возможно.
Последний официальный дистрибутив lilo (версия 21) не работает с устройствами RAID, и, таким образом, ядро не может быть загружено с устройства RAID. Если Вы используете эту версию, Ваша файловая система `/boot` должна быть расположена на не-RAID устройстве. Чтобы быть уверенным, что Ваша система загрузится в любом случае, создайте подобные `/boot` разделы на всех дисках вашего RAID, таким образом BIOS всегда загрузит данные с первого попавшегося диска. Это требует, чтобы Вы не загружались с отказавшего диска.
Для redhat 6.1 стал доступен патч к lilo 21, который способен найти `/boot` на RAID-1. Заметьте, что это не работает для любого другого уровня, RAID-1 (mirroring) - единственный поддерживаемый уровень RAID. Этот патч (`lilo.raid1`) может быть найден в `dist/redhat-6.1/SRPMS/SRPMS/lilo-0.21-10.src.rpm` на любом зеркале redhat. Пропатченая версия LILO позволит `boot=/dev/md0` в `lilo.conf` и сделает каждый диск зеркала загрузочным.
Другой путь быть уверенным, что Ваша система сможет всегда загрузится - создать загрузочную дискету после завершения всех установок. Если диск, на котором, расположена файловая система `/boot` умирает, Вы сможете всегда загрузиться с дискеты.

## 4.12 Корневая файловая система на RAID

В случае загрузки системы с RAID, корневая файловая система (/) должна монтироваться на устройство RAID. Ниже предлагается два метода для достижения этого. Так как ни один из текущих дистрибутивов (по крайней мере которые я знаю) не поддерживает инсталляцию на RAID устройство, методы предполагают, что Вы устанавливаете на обычный раздел, и затем, когда установка завершена, перемещаете содержимое Вашей не-RAID корневой файловой системы на новое RAID устройство.

### 4.12.1 Метод 1

Этот метод предполагает, что У вас есть резервный диск, который не входит в конфигурируемый RAID, и на который Вы можете инсталлировать систему.

- Сначала установите обычную систему на ваш дополнительный диск.
- Запустите планируемое ядро, возьмите raid-патчи и утилиты и сделайте загрузку Вашей системы с новым RAID-способным ядром. Убедитесь, что поддержка RAID **в** ядре, и не загружается как модуль.
- Итак, сейчас Вы должны сконфигурировать и создать планируемый к использованию RAID для корневой файловой системы. Эта стандартная процедура описана в этом документе.
- Просто убедитесь, что все в порядке, попробуйте перезагрузить систему, чтобы посмотреть загрузится ли новый RAID. Должен загрузиться.
- Поместите файловую систему на новый массив (используя `mke2fs`), и смонтируйте его в /mnt/newroot
- Сейчас, скопируйте содержимое Вашей текущей корневой файловой системы (с резервного диска) на новую корневую файловую систему (массив). Есть много способов это сделать, один из них

  ```
  cd /
  find . -xdev | cpio -pm /mnt/newroot
  ```

- Вы должны модифицировать файл `/mnt/newroot/etc/fstab` для использования правильного устройства (корневого устройства `/dev/md?`) для корневой файловой системы.
- Сейчас, размонтируйте текущуют `/boot` файловую систему, и смонтируйте вместо нее загрузочное устройство указанное в `/mnt/newroot/boot`. Это требуется для LILO для успешного запуска на следующем шаге.
- Обновите `/mnt/newroot/etc/lilo.conf` для указания на правильные устройства. Загрузочное устройство должно все еще быть обычным диском (не-RAID устройством), но root устройство должно указывать на Ваш новый RAID. Когда сделано, запустите

  ```
  lilo -r /mnt/newroot
  ```

  Этот запуск LILO дожен завершиться без ошибок.
- Перезагрузите систему, и смотрите, чтобы все происходило как ожидается :)

Если Вы делаете это с IDE дисками, удостоверьтесь что установили в BIOS, что все диски "auto-detect" типа, таким образом BIOS позволит Вашей машине загружаться даже если диск отсутствует.

### 4.12.2 Метод 2

Этот метод требует, чтобы Вы использовали raidtools/патч, которые включают директиву failed-disk. Это должны быть утилиты/патч для всех ядер от 2.2.10 и выше.
Вы можете использовать этот метод **только** на RAID уровня 1 и выше. Идея состоит в использовании установки системы на диск специально отмеченный как отказавший в RAID, тогда скопировав систему на RAID, запущенный в деградированном режиме, и затем сделав доступным для RAID уже не нужный "инсталляционный диск", уничтожаете старую установку, но запускаете RAID в не деградированном режиме.

- Сначала, установите обычную систему на один диск (который позже будет частью Вашего RAID). Важно, чтобы этот диск ( или раздел) не был наименьшим. Если это так, будет позже не возможно добавить его в массив RAID!
- Теперь, возьмите ядро, патчи, утилиты и т.п. Вы уже заучили это. Сделайте Вашу систему загрузочной с Вашим новым ядром, который содержит необходимую поддержку RAID скомпилированой в ядре.
- Сейчас, установите RAID с вашим текущим корневым устройством как `отказавшим диском` в фале `raidtab`. Не помещайте `отказавший диск` как первый диск в `raidtab`, это создаст проблемы с запуском. Создайте RAID, и поместите файловую систему на него.
- Попробуйте перегрузиться и посмотреть, запустится ли RAID должным образом
- Скопируйте системные файлы и ре-конфигурируйте систему для использования RAID в качестве корневого устройства, как описано в предыдущей секции.
- Когда Ваша система успешно загрузиться с RAID, Вы можете модифицировать файл `raidtab` для включения предыдущего `отказавшего диска` как обычного `raid-диска`. Сейчас, `raidhotadd` диск ваш RAID.
- Сейчас Вы должны получить систему, которая может загружаться с не деградированного RAID.

## 4.13 Выполнение загрузки ситемы с RAID

Чтобы ядро смогло смонтировать корневую файловую систему, вся поддержка для устройства, на котором расположена корневая файловая система, должна быть в ядре. Таким образом, в случае монтирования файловой системы на RAID устройстве, ядро *должно* содержать поддержку RAID.
Обычный способ удостовериться, что ядро может видеть RAID устройство - просто скомпилировать ядро с вкомпилированой поддержкой RAID. Убедитесь, что компилируете поддержку RAID *в* ядро, а *не* как модули. Ядро не может загружать модули (с корневой файловой системы) перед монтированием файловой системы.
Однако, с RedHat-6.0 поставляется с ядром, которое содержит поддержку RAID нового стиля в виде модулей, я здесь опишу как можно использовать стандартное ядро RedHat-6.0 и все еще загружать систему с RAID.

### 4.13.1 Загрузка с RAID в виде модуля

Для достижения этого, Вы должны указать LILO использовать RAM-диск. Используйте команду `mkinitrd` для создания ramdisk, содержащего все модули ядра необходимые для загрузки корневого раздела. Это можно сделать так:

```
mkinitrd --with=<module> <ramdisk name> <kernel>
```

Например:

```
mkinitrd --with=raid5 raid-ramdisk 2.2.5-22
```

Это должно гарантировать, что ядро найдет указанный RAID модуль при монтировании устройства во время загрузки.

### 4.14 Ловушки

Никогда, НИКОГДА, **никогда** не пере-размечайте разделы дисков, которые являются частью рабочего RAID. Если Вы должны изменить таблицу разделов на диске, который - часть RAID, сначала остановите массив, затем пере-размечайте.
Поместить много дисков на одну шину очень просто. Обычная шина Fast-Wide SCSI может выдерживать до 10 Мб/с, что меньше, чем могут дать многие современные диски. Размещение шести таких дисков на одной шине, конечно, не даст ожидаемого прироста производительности.
Многие SCSI контроллеры дадут Вам наивысшую производительность, если SCSI шины почти маскимально забиты дисками на них. Вы не увидите увеличения производительности от использования двух 2940s с двумя старыми SCSI дисками, вместо простого подключения двух дисков к одному контролеру.
Если Вы забудете опцию persistent-superblock, Ваш массив может сразу не запуститься после остановки. Просто пере-создайте массив с установленной опцией в raidtab.
Если RAID-5 отказывается реконструироваться после удаления и повторного добавления диска, это может быть из-за порядка устройств в raidtab. Попытайтесь переместить первую "device ..." и "raid-disk ..." пару в низ описания массива в фале raidtab.
Большинство "сообщений об ошибках" , которые мы видим в linux-kernel, от людей, у которых как-то не работает правильный патч RAID с правильной версией raidtools. Если Вы используете 0.90 RAID, убедитесь, что используете raidtools для этой версии.

## 5 Тестирование

Если Вы планируете использовать RAID для получения отказоустойчивости, Вы также должны проверить Вашу конфигурацию, чтобы увидеть, что она действительно рабочая. Итак, как кожно имитировать отказ диска ?
Коротко - Вы не можете, исключая, возможно, "горячее" выдергивание шнура из жесткого диска, отказ которого Вы хотите имитировать. Вы никогда знаете, что может случиться при отказе диска. Возможна электрическая блокировка шины к которой он подсоединен, что приведет к недоступности всех устройств на шине. Хотя я никогда о таком не слышал. Диск может также просто выдавать ошибки чтения/записи на уровне SCSI/IDE, что в свою очередь даст уровню RAID корректно обработать эту ситуацию. Это, к счастью, происходит чаще всего.

### 5.1 Имитация отказа диска

Если Вы хотите имитировать отказ диска - отсоедините устройство. Вы должны делать это при **выключенном питании**. Если Вы заинтересованы в тестировании выживут ли Ваши данные без одного диска, по сравнению с обычным количеством, нет иного выхода, как отключение. Завершите систему, отсоедините диск и загрузитесь снова.
Посмотрите в syslog, и загляните в `/proc/mdstat`, чтобы посмотреть как действует RAID. Сработало?
Запомните, что Вы **должны** запускать массив RAID-{1,4,5} для возможности пережить отказ диска. Линейный или RAID-0 откажут полностью при отсутствии диска.
Когда Вы подключили диск снова (при выключенном питании, конечно), Вы можете опять добавить "новое" устройство в RAID, командой `raidhotadd`.

### 5.2 Имитация повреждения данных

RAID (будь то программный или аппаратный), предполагает, что если запись на диск не вернула ошибку, то запись была успешной. Следовательно, если Ваш диск повреждает данные без возврата ошибки, Ваши данные *bygym* повреждены. Это конечно очень не желательно, но возможно, и это приведет к повреждению файловой системы.

RAID не может и не должен защищать от повреждения данных на носителе. Следовательно, нет никакого смысла намеренно повреждать данные (используя `dd` например) на диске, чтобы посмотреть как RAID система это обработает. Наиболее вероятно (если Вы не повредите суперблок RAID), что RAID уровень никогда не догадается о повреждении, но файловая система на Вашем RAID устройстве будет повреждена.
Это путь вещей предполагаемых для работы. RAID не гарантирует целостности данных, он просто позволяет Вам сохранить данные при отказе диска (это, конечно, справедливо для RAID уровня 1 или выше).

# 6 Реконструкция

Если Вы читали другие части этого HOWTO, Вы должны уже хорошо представлять как вызывается реконструкция деградировавшего RAID. Я обобщаю:

- Выключаем систему
- Заменяем отказавший диск
- Включаем систему снова.
- Используем `raidhotadd /dev/mdX /dev/sdX` для добавления диска в массив
- Пьем кофе, пока работает автоматическая реконструкция

И это так.
Итак, обычно это так, пока Вам не повезет и Ваш RAID станет нерабочим из-за отказа более одного диска. Это может фактически случиться, если у Вас несколько дисков на одной шине, и один диск захватит шину при отказе. Другие диски, в порядке, но будут недоступны для RAID уровня, так как шина блокирована, и они будут помечены как отказавшие. На RAID-5, где у Вас может быть резервный диск, потеря двух или более дисков может быть фатальной.
Следующая секция - объяснение, которое прислал мне Martin Bene, и описал возможность восстановления при жутком сценарии описанном выше. Это использует директиву `failed-disk` в Вашем `/etc/raidtab`, таким образом это будет работать с ядрами 2.2.10 и выше.

## 6.1 Восстановление при множественных отказах диска

Сценарий таков:

- Контроллер умирает и отключает два диска одновременно,
- Все диски на одной scsi шине могут быть недоступны, если отказывает диск,
- Отсоединяется кабель...

Коротко: довольно часто у Вас *временный отказ* нескольких дисков одновременно; в последствии суперблоки RAID не синхронизированы и Вы уже не можете инициализировать Ваш RAID массив. Остается одно: перезаписать суперблоки RAID подав `mkraid --force`
Чтобы это сделать, Вам нужно иметь свежий `/etc/raidtab` - если он **НЕ ТОЧНО** соответствует устройствам и исходному порядку дисков, это не сработает.
Посмотрите в sylog на результат попытки запуска массива, Вы увидите отсчет событий для каждого суперблока; обычно лучше оставить диск с наименьшим отсчетом события, т.е. с самым старым.
Если Вы делаете `mkraid` без `failed-disk`, нить восстановления немедленно выругается и начнет перестроение блоков паритета - не то, что Вам сейчас нужно.
С `failed-disk`, В можете точно указать какие диски Вы хотите активировать и, возможно, попробовать различные комбинации для лучшего результата. Подсказка, при этих экспериментах монтируете систему в режиме только для чтения... Это было успешно использовано, по крайней мере, двумя парнями, с которыми я контактировал.

| Размер куска | Размер блока | Чтение Кб/с | Запись Кб/с |
|---|---|---|---|
| 4k | 1k | 19712 | 18035 |
| 4k | 4k | 34048 | 27061 |
| 8k | 1k | 19301 | 18091 |
| 8k | 4k | 33920 | 27118 |
| 16k | 1k | 19330 | 18179 |
| 16k | 2k | 28161 | 23682 |
| 16k | 4k | 33990 | 27229 |
| 32k | 1k | 19251 | 18194 |
| 32k | 4k | 34071 | 26976 |

# 7 Производительность

Эта секция содержит несколько бенчмарков из реальных систем, использующих программный RAID.
Бенчмарки производились программой `bonnie`, и всегда с файлами в два или более раза большими объема физической памяти (RAM) в машине.
В приведенных бенчмарках измерялась *только* пропускная способность записи и чтения одного большого файла. Это полезная информация, если интересует максимальная пропускная способность ввода/вывода для длинных блоков данных. Однако, эти цифры мало говорят нам о производительности, при использовании массива для спула новостей, web-сервера, и т.д. Всегда помните, эти цифры - результат запуска "синтетического" теста. Несколько реальных программ делают то же, что и `bonnie`, и хотя хорошо смотреть на эти цифры, они не являются основными индикаторами реальной производительности.
Сейчас, у меня есть результаты с моей собственной машины. Конфигурация такова:

- Dual Pentium Pro 150 MHz
- 256 MB RAM (60 MHz EDO)
- Три IBM UltraStar 9ES 4.5 GB, SCSI U2W
- Adaptec 2940U2W
- Один IBM UltraStar 9ES 4.5 GB, SCSI UW
- Adaptec 2940 UW
- Ядро 2.2.7 с RAID патчами

Три U2W диска повешены на U2W контроллер, и UW диск на UW контроллер.
Представляется невозможным передавать более 30 Мб/с по шинам SCSI на этой системе, используя RAID или нет. Как я думаю, это из-за слишком старой системы, скорости памяти и ограничений того, что можно послать через SCSI контроллеры.

## 7.1 RAID-0

**Чтение** это - **Последовательный блочный ввод**, и **Запись** это - **Последовательный блочный вывод**. Размер файла во всех тестах - 1Гб. Тести были проведены в однопользовательском режиме. Драйвер SCSI был сконфигурирован для не использования очереди помеченных комманд.
Отсюда видно, что размер куска в RAID не имеет значения. Однако, размер блока ext2fs должен быть как можно более большим, как 4KB (т.е. размер страницы) на IA-32.

| Размер куска | Размер блока | Чтение Кб/с | Запись Кб/с |
|---|---|---|---|
| 32k | 4k | 33617 | 27215 |

| Размер куска | Размер блока | Чтение Кб/с | Запись Кб/с |
|---|---|---|---|
| 8k | 1k | 11090 | 6874 |
| 8k | 4k | 13474 | 12229 |
| 32k | 1k | 11442 | 8291 |
| 32k | 2k | 16089 | 10926 |
| 32k | 4k | 18724 | 12627 |

### 7.2 RAID-0 с TCQ

Тут, драйвер SCSI был сконфигурирован для использования очереди помеченных комманд (TCQ), с глубиной очереди - 8. Все остальное как и в предыдущем случае.
Сдесь больше не производилось тестов. Как видно TCQ немного увеличивает производительность записи, но на самом деле здесь совсем не большая разница.

### 7.3 RAID-5

Массив был сконфигурирован в режим RAID-5, и были сделаны подобные тесты.
Сейчас, и размер куска и размер блока действительно дают различие.

### 7.4 RAID-10

RAID-10 это- "зеркало stripes", или, массив RAID-1 двух массивов RAID-0. Размер куска - размер кусков в обоих, и в RAID-1 и в двух RAID-0 массивах. Я не проводил тестов с различающимися этими размерами кусков, хотя это должна быть вполне правильная установка.
Больше тестов не производилось. Размер файла был 900Мб, так как четыре раздела заняли по 500 Мб, так что не осталось места для 1Гб файла в этой конфигурации (RAID-1 на двух 1000Мб массивах).

## 8 Благодарности

Следующие люди участвовали в создании этой документации:

- Ingo Molnar
- Jim Warren
- Louis Mandelstam
- Allan Noah
- Yasunori Taniike

| Размер куска | Размер блока | Чтение Кб/с | Запись Кб/с |
|---|---|---|---|
| 32k | 1k | 13753 | 11580 |
| 32k | 4k | 23432 | 22249 |

- Martin Bene
- Bennett Todd
- The Linux-RAID mailing list people
- The ones I forgot, sorry :)

Пожалуйста, присылайте автору коррективы, предложения и т.п. Это единственный путь совершенствования HOWTO.