

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский технологический университет»

**РТУ МИРЭА**

**ЗАЧТЕНО**

\_\_\_\_\_\_\_\_\_\_/Н.С.Ершов/

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г.

Дисциплина: «Обеспечение устойчивого функционирования автоматизированных систем в условиях внешних воздействий»

Практическая работа 3: «RAID массивы»

Выполнил:

Студент 3 курса

Кутьин З.С

Группа: БББО-05-20

Проверил:

Ершов Н.С.

**Содержание**

**Введение**

**Актуальность работы:** В данной работе рассматриваются и изучаются технологии повышения отказоустойчивости и дублирования данных при помощи RAID массивов.

**Цель работы:** Получить навыки работы с технологиями RAID массивов.

**Задачи:**

* Сделать несколько виртуальных дисков;
* Создать RAID массив определенного уровня из этих дисков;

Пользоваться можно утилитой создания RAID массивов **mdadm** и любыми утилитами управления разделами. Все основные этапы работы обозначить снимками экрана.

Материал:

1. Даже при резервном копировании последствия сбоя диска на сервере могут быть разрушительными. RAID (redundant arrays of inexpensive disks — избыточные массивы независимых дисков) — это система, распределяющая или дублирующая данные с помощью многочисленных дисков. Система RAID не только помогает избежать потери данных но и минимизирует время простоя, связанного с отказом аппаратуры (часто сводя его к нулю), и потенциально повышает производительность.

Систему RAID можно реализовать с помощью специального оборудования, благодаря которому операционная система интерпретирует группу жестких дисков как один составной накопитель. Кроме того, ее можно реализовать так, чтобы операционная система просто считывала или записывала данные на жестких дисках по правилам системы RAID.

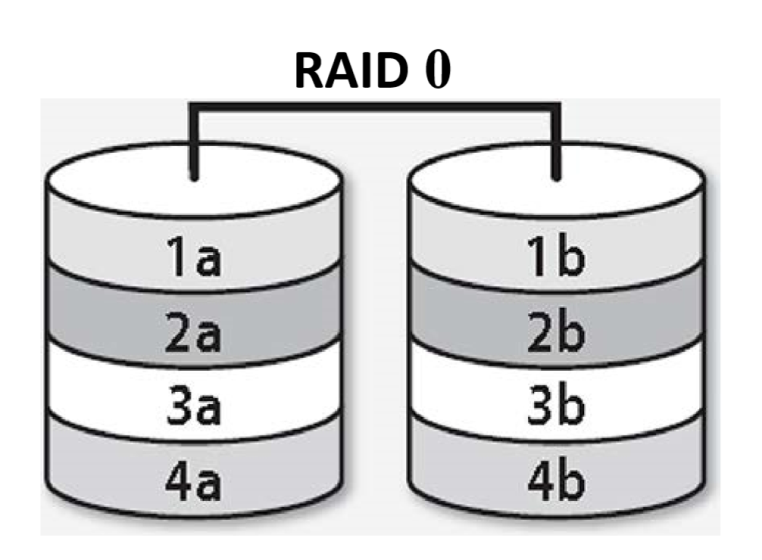
Поскольку диски сами по себе являются узким местом в реализации системы RAID, нет причин предполагать, что аппаратная реализация системы RAID всегда будет работать быстрее, чем оперативная.

В прошлом аппаратная система RAID была доминирующей по двум причинам: из-за недостатка программной поддержки (отсутствие прямой поддержки со стороны операционных систем) и способности аппаратного обеспечения буферизовать записи в форме некой энергонезависимой памяти.

Система RAID может выполнять две основные операции. Во-первых, она может повысить производительность, разбросав данные по многим накопителям и позволив нескольким дискам одновременно передавать или получать поток данных. Во-вторых, она может реплицировать данные по многим накопителям, уменьшая риск, связанный со сбоем одного диска.

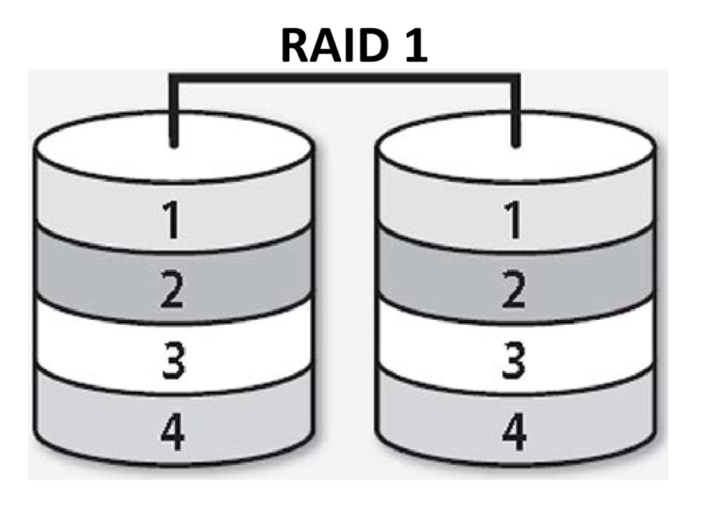
Репликация может осуществляться в двух видах: зеркальное отражение (mirroring), при котором блоки данных побитово репродуцируются на нескольких накопителях, и схемы четности (parityschemes), в которых один или несколько накопителей содержат контрольную сумму для проверки ошибок на остальных накопителях. Зеркальное отражение быстрее, но занимает много места на диске. Схемы четности более экономно используют дисковую память, но работают медленнее.Система RAID традиционно описывается в терминах “уровней”, определяющих конкретные детали параллелизма и избыточности. Возможно, этот термин является неудачным, потому что более высокий уровень не обязательно оказывается лучшим. Уровни — это просто разные конфигурации; вы можете использовать любой из них. На рисунках, приведенных ниже, числа обозначают магнитные ленты, а буквы а, b и с — блоки данных на этих лентах. Блоки, помеченные буквами р и q, являются блоками четности.

Уровень **RAID 0** используется только для повышения производительности. Он объединяет два или более накопителей одинаковых размеров, но, вместо их объединения последовательно один за другим, он распределяет данные между дисками, входящими в пул. Операции последовательного чтения и записи в этом случае оказываются распределенными между несколькими дисками, что снижает время записи и доступа к диску.

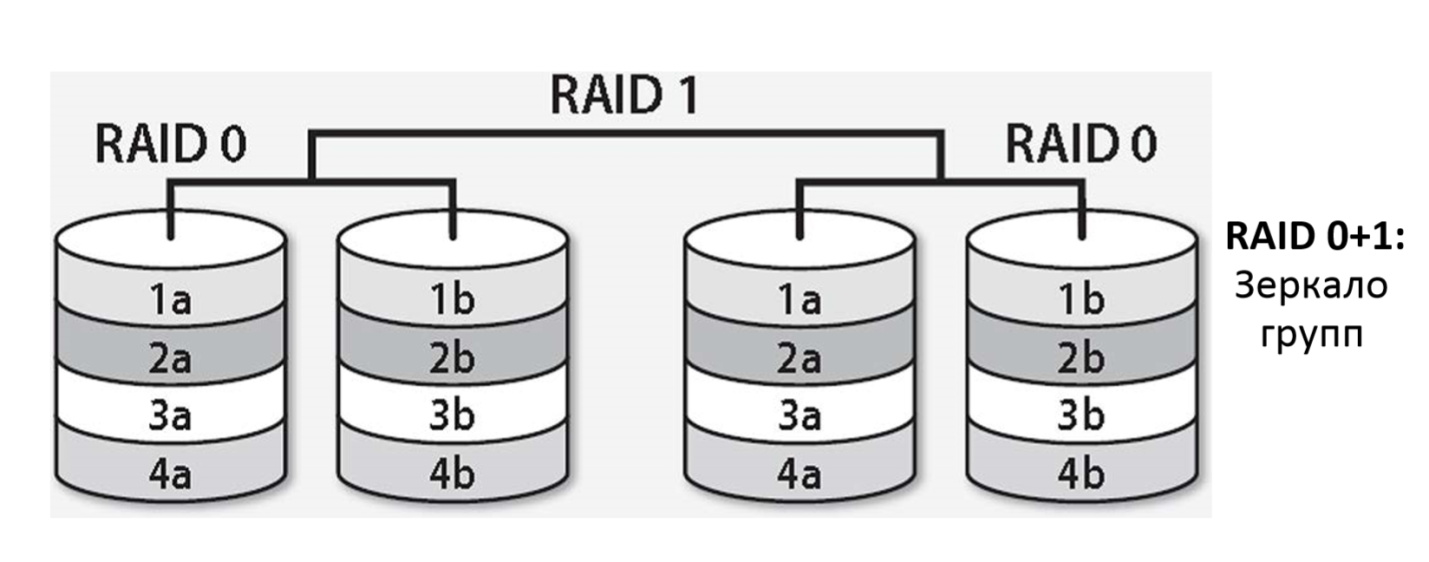
****

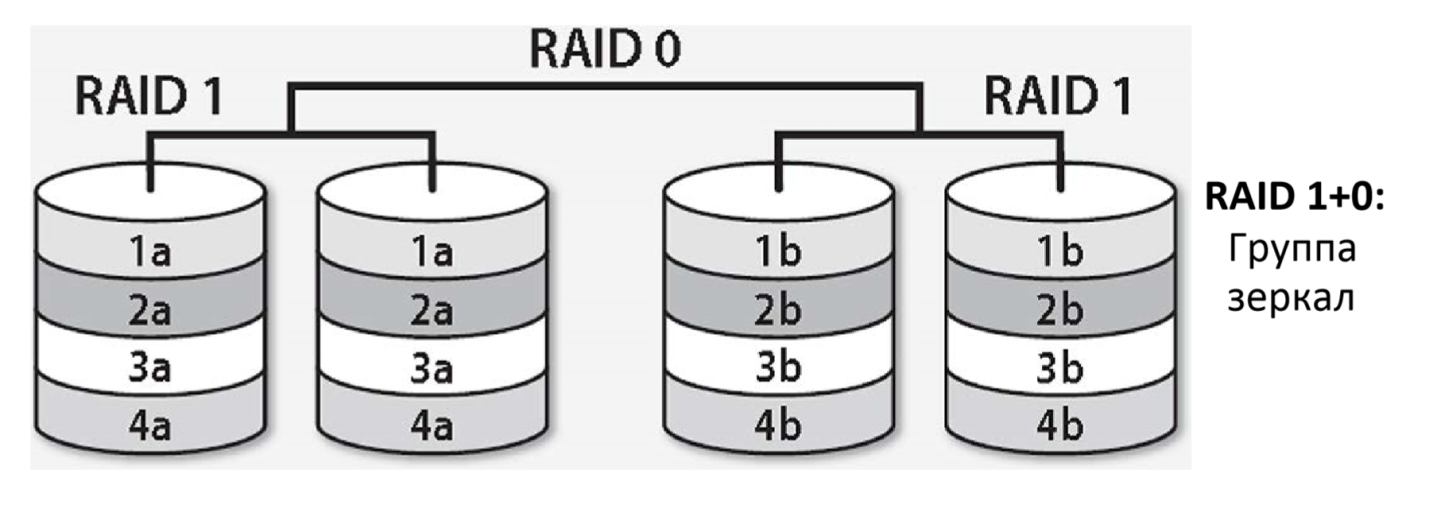
Обратите внимание на то, что надежность уровня 0 ниже, чем надежность отдельных дисков. Вероятность отказа массива из двух дисков в течение года примерно в два раза выше, чем у отдельного диска, и т.д.

Уровень **RAID 1** известен как “зеркальное отражение”. Записи дублируются одновременно на нескольких дисках. Эта схема замедляет процесс записи по сравнению с записью данных на отдельный диск. Однако она обеспечивает скорость считывания, сравнимую с уровнем **RAID 0**, потому что чтение можно распределять между несколькими дублирующими накопителями.

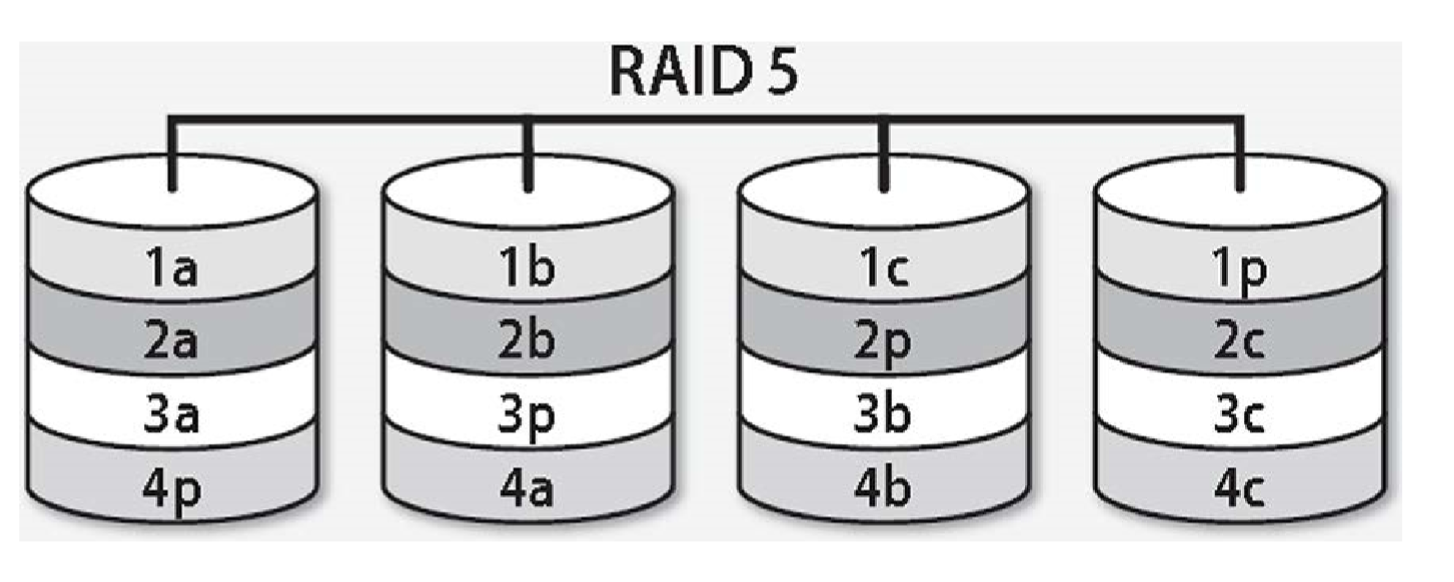
****

Уровни **RAID 1+0** и **0+1** представляют собой группы зеркал или зеркала групп. С логической точки зрения они представляют собой сочетание уровней **RAID 0** и **RAID 1**, но многие контроллеры и программы обеспечивают их непосредственную поддержку. Цель обоих режимов — одновременно достичь производительности уровня **RAID 0** и избыточности уровня **RAID 1**.

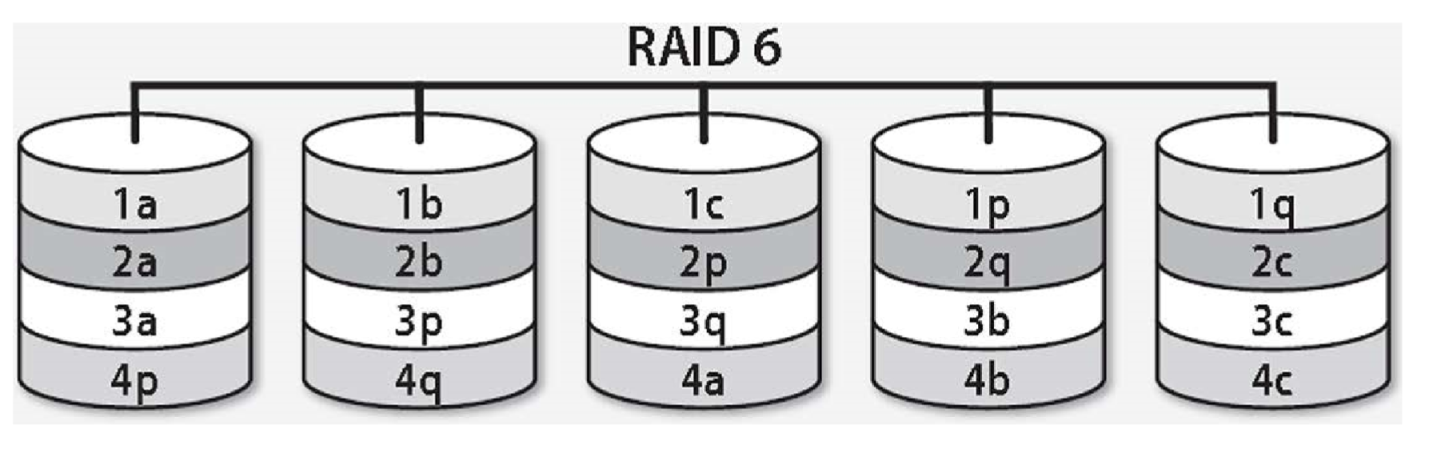
****

****

Уровень **RAID 5** распределяет и данные, и информацию о четности, добавляя избыточность и одновременно повышая производительность чтения. Кроме того, уровень **RAID 5** более эффективно использует дисковое пространство, чем уровень **RAID 1**. Если массив состоит из **N** накопителей (требуется не менее трех), то **N—1** из них могут хранить данные. Следовательно, эффективность использования дисковой памяти на уровне **RAID 5** превышает 67%, в то время как зеркальное отражение не может превысить 50%.

****

Уровень **RAID 6** аналогичен уровню **RAID 5** с двумя дисками четности. Массив **RAID 6** может сохранить работоспособность при полном отказе двух накопителей без потери данных.

****

Уровни **RAID 2, 3** и **4** хотя и определены, но используются редко. Менеджеры логических томов обычно могут и распределять данные (**RAID 0**), и осуществлять зеркальное отражение (**RAID 1**).

Управление программным RAID-массивом в Linux выполняется с помощью программы **mdadm**.

У программы **mdadm** есть несколько режимов работы.

**Assemble (сборка)**

Собрать компоненты ранее созданного массива в массив. Компоненты можно указывать явно, но можно и не указывать — тогда выполняется их поиск по суперблокам.

**Build (построение)**

Собрать массив из компонентов, у которых нет суперблоков. Не выполняются никакие проверки, создание и сборка массива в принципе ничем не отличаются.

**Create (создание)**

Создать новый массив на основе указанных устройств. Использовать суперблоки размещённые на каждом устройстве.

**Monitor (наблюдение)**

Следить за изменением состояния устройств. Для RAID0 этот режим не имеет смысла.

**Grow (расширение или уменьшение)**

Расширение или уменьшение массива, включаются или удаляются новые диски.

**Incremental Assembly (инкрементальная сборка)**

Добавление диска в массив.

**Manage (управление)**

Разнообразные операции по управлению массивом, такие как замена диска и пометка как сбойного.

**Misc (разное)**

Действия, которые не относятся ни к одному из перечисленных выше режимов работы.

**Auto-detect (автоообнаружение)**

Активация автоматически обнаруживаемых массивов в ядре Linux.

Формат вызова

mdadm [mode] [array] [options]

Режимы:

* -A, --assemble — режим сборки
* -B, --build — режим построения
* -C, --create — режим создания
* -F, --follow, --monitor — режим наблюдения
* -G, --grow — режим расширения
* -I, --incremental — режим инкрементальной сборки

Рассмотрим, как выполнить настройку RAID-массива 5 уровня на трёх дисковых разделах. Мы будем использовать разделы:

/dev/hde1

/dev/hdf2

/dev/hdg1

В том случае если разделы иные, необходимо использовать соответствующие имена файлов

**Создание разделов**

Нужно определить на каких физических разделах будет создаваться RAID-массив. Если разделы уже есть, нужно найти свободные (*fdisk -l*). Если разделов ещё нет, но есть неразмеченное место, их можно создать с помощью программ *fdisk* или *cfdisk*.

Просмот существующих разделов:

 %# fdisk -l

Disk /dev/hda: 12.0 GB, 12072517632 bytes

255 heads, 63 sectors/track, 1467 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/hda1 \* 1 13 104391 83 Linux

/dev/hda2 14 144 1052257+ 83 Linux

/dev/hda3 145 209 522112+ 82 Linux swap

/dev/hda4 210 1467 10104885 5 Extended

/dev/hda5 210 655 3582463+ 83 Linux

...

...

/dev/hda15 1455 1467 104391 83 Linux

Просмотр, какие разделы куда смонтированы, и свободное место на них (размеры в килобайтах):

 %# df -k

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/hda2 1035692 163916 819164 17% /

/dev/hda1 101086 8357 87510 9% /boot

/dev/hda15 101086 4127 91740 5% /data1

...

...

...

/dev/hda7 5336664 464228 4601344 10% /var

### 

### Размонтирование

### Если использовать созданные ранее разделы, необходимо размонтировать их. RAID-массив нельзя создавать поверх разделов, на которых находятся смонтированные файловые системы.

%# umount /dev/hde1

 %# umount /dev/hdf2

 %# umount /dev/hdg1

### Изменение типа разделов

Желательно (но не обязательно) изменить тип разделов, которые будут входить в RAID-массив и установить его равным FD (Linux RAID autodetect). Изменить тип раздела можно с помощью fdisk.

Рассмотрим, как это делать на примере раздела /dev/hde1.

 %# fdisk /dev/hde

The number of cylinders for this disk is set to 8355.

There is nothing wrong with that, but this is larger than 1024,

and could in certain setups cause problems with:

1) software that runs at boot time (e.g., old versions of LILO)

2) booting and partitioning software from other OSs

(e.g., DOS FDISK, OS/2 FDISK)

Command (m for help):

Use FDISK Help

Now use the fdisk m command to get some help:

Command (m for help): m

...

...

p print the partition table

q quit without saving changes

s create a new empty Sun disklabel

t change a partition's system id

...

...

Command (m for help):

Set The ID Type To FD

Partition /dev/hde1 is the first partition on disk /dev/hde.

Modify its type using the t command, and specify the partition number

and type code.

You also should use the L command to get a full listing

of ID types in case you forget.

Command (m for help): t

Partition number (1-5): 1

Hex code (type L to list codes): L

...

...

...

16 Hidden FAT16 61 SpeedStor f2 DOS secondary

17 Hidden HPFS/NTF 63 GNU HURD or Sys fd Linux raid auto

18 AST SmartSleep 64 Novell Netware fe LANstep

1b Hidden Win95 FA 65 Novell Netware ff BBT

Hex code (type L to list codes): fd

Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help):

Make Sure The Change Occurred

Use the p command to get the new proposed partition table:

Command (m for help): p

Disk /dev/hde: 4311 MB, 4311982080 bytes

16 heads, 63 sectors/track, 8355 cylinders

Units = cylinders of 1008 \* 512 = 516096 bytes

Device Boot Start End Blocks Id System

/dev/hde1 1 4088 2060320+ fd Linux raid autodetect

/dev/hde2 4089 5713 819000 83 Linux

/dev/hde4 6608 8355 880992 5 Extended

/dev/hde5 6608 7500 450040+ 83 Linux

/dev/hde6 7501 8355 430888+ 83 Linux

Command (m for help):

Save The Changes

Use the w command to permanently save the changes to disk /dev/hde:

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table.

The new table will be used at the next reboot.

Syncing disks.

### 

### Создание RAID-массива

### Создание RAID-массива выполняется с помощью программы mdadm (ключ --create). Мы воспользуемся опцией --level, для того чтобы создать RAID-массив 5 уровня. С помощью ключа --raid-devices укажем устройства, поверх которых будет собираться RAID-массив.

 %# mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/hde1 /dev/hdf2 /dev/hdg1

mdadm: layout defaults to left-symmetric

mdadm: chunk size defaults to 64K

mdadm: /dev/hde1 appears to contain an ext2fs file system

size=48160K mtime=Sat Jan 27 23:11:39 2007

mdadm: /dev/hdf2 appears to contain an ext2fs file system

size=48160K mtime=Sat Jan 27 23:11:39 2007

mdadm: /dev/hdg1 appears to contain an ext2fs file system

size=48160K mtime=Sat Jan 27 23:11:39 2007

mdadm: size set to 48064K

Continue creating array? y

mdadm: array /dev/md0 started.

### 

### Если вы хотите сразу создать массив, где диска не хватает (degraded) просто укажите слово missing вместо имени устройства. Для RAID5 это может быть только один диск; для RAID6 — не более двух; для RAID1 — сколько угодно, но должен быть как минимум один рабочий.

### Проверка правильности сборки

Убедиться, что RAID-массив проинициализирован корректно можно просмотрев файл /proc/mdstat. В этом файле отражается текущее состояние RAID-массива.

 %# cat /proc/mdstat

Personalities : [raid5]

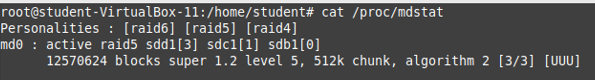
read\_ahead 1024 sectors

md0 : active raid5 hdg1[2] hde1[1] hdf2[0]

4120448 blocks level 5, 32k chunk, algorithm 3 [3/3] [UUU]

unused devices: <none>

Обратим внимание на то, как называется новый RAID-массив. В нашем случае он называется /dev/md0. Обращение к массиву будет происходить по этому имени.



### Создание файловой системы поверх RAID-массива

Новый RAID-раздел нужно отформатировать, т.е. создать на нём файловую систему. Сделать это можно при помощи программы из семейства **mkfs**. Если будет создаваться файловая система ext3, используется программа **mkfs.ext3**. :

 %# mkfs.ext3 /dev/md0

mke2fs 1.39 (29-May-2006)

Filesystem label=

OS type: Linux

Block size=1024 (log=0)

Fragment size=1024 (log=0)

36144 inodes, 144192 blocks

7209 blocks (5.00%) reserved for the super user

First data block=1

Maximum filesystem blocks=67371008

18 block groups

8192 blocks per group, 8192 fragments per group

2008 inodes per group

Superblock backups stored on blocks:

8193, 24577, 40961, 57345, 73729

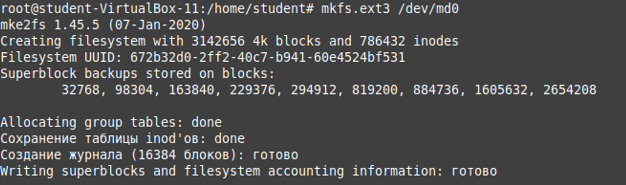
Writing inode tables: done

Creating journal (4096 blocks): done

Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 33 mounts or

180 days, whichever comes first. Use tune2fs -c or -i to override.



Имеет смысл для лучшей производительности файловой системы указывать при создании количество дисков в рейде и количество блоков файловой системы которое может поместиться в один страйп ( chunk ), это особенно важно при создании массивов уровня RAID0,RAID5,RAID6,RAID10. Для RAID1 ( mirror ) это не имеет значения так как запись идет всегда на один device, a в других типах рейдов информация записывается последовательно на разные диски порциями соответствующими размеру stripe. Например если используется RAID5 из 3 дисков, с дефолтным размером страйпа в 64К и используется файловая система ext3 с размером блока в 4К то можно вызывать команду mkfs.ext следующим образом:

%# mkfs.ext3 -b 4096 -E stride=16,stripe-width=32 /dev/md0

stripe-width обычно расчитывается как stride \* N ( N это дата диски в массиве - например в RAID5 - два дата диска и один parity ) Для не менее популярной файловой системы XFS надо указывать не количество блоков файловой системы соответствующих размеру stripe в массиве, а непосредственно размер самого страйпа.

 %# mkfs.xfs -d su=64k,sw=3 /dev/md0

### Создание конфигурационного файла mdadm.conf

Система сама не запоминает какие RAID-массивы ей нужно создать и какие компоненты в них входят. Эта информация находится в файле **mdadm.conf**.

Строки, которые следует добавить в этот файл, можно получить при помощи команды

mdadm --detail --scan --verbose

### Пример её использования:

 %# mdadm --detail --scan --verbose

ARRAY /dev/md0 level=raid5 num-devices=4

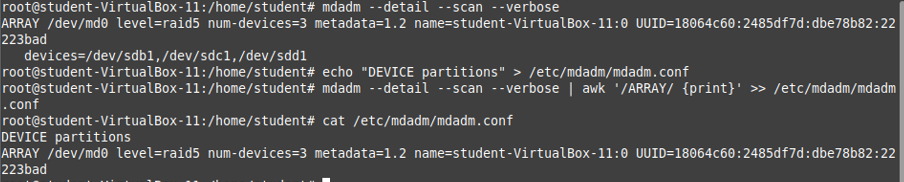
UUID=77b695c4:32e5dd46:63dd7d16:17696e09

devices=/dev/hde1,/dev/hdf2,/dev/hdg1

Если файла **mdadm.conf** ещё нет, можно его создать:

 %# echo "DEVICE partitions" > /etc/mdadm/mdadm.conf

 %# mdadm --detail --scan --verbose | awk '/ARRAY/ {print}' >> /etc/mdadm/mdadm.conf



### Создание точки монтирования для RAID-массива

Поскольку создаётся новая файловая система, вероятно, потребуется и новая точка монтирования. Необходимо дать ей название, например /raid.

 %# mkdir /raid



### Изменение /etc/fstab

Для того чтобы файловая система, созданная на новом RAID-массиве автоматически монтировалась при загрузке, необходимо добавить соответствующую запись в файл /etc/fstab хранящий список автоматически монтируемых при загрузке файловых систем.

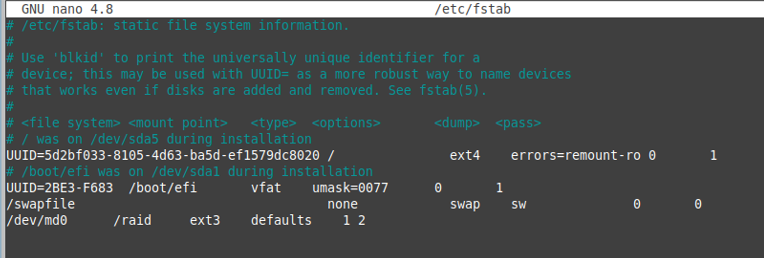
/dev/md0 /raid ext3 defaults 1 2

Если разделы, которые использовались раньше, объединялись в RAID-массив, нужно отключить их монтирование: удалить или закомментировать соответствующие строки в файле /etc/fstab. Закомментировать строку можно символом #

#/dev/hde1 /data1 ext3 defaults 1 2

#/dev/hdf2 /data2 ext3 defaults 1 2

#/dev/hdg1 /data3 ext3 defaults 1 2



### Монтирование файловой системы нового RAID-массива

Для того чтобы получить доступ к файловой системе, расположенной на новом RAID-массиве, её нужно смонтировать. Монтирование выполняется с помощью команды **mount**.

Если новая файловая система добавлена в файл /etc/fstab, можно смонтировать её командой **mount** **-a** (смонтируются все файловые системы, которые должны монтироваться при загрузке, но сейчас не смонтированы).

 %# mount -a

Можно смонтировать только нужный нам раздел (при условии, что он указан в /etc/fstab).

 %# mount /raid



Если раздел в /etc/fstab не указан, то при монтировании мы должны задавать как минимум два параметра — точку монтирования и монтируемое устройство:

 %# mount /dev/md0 /raid

### Проверка состояния RAID-массива

Информация о состоянии RAID-массива находится в файле /proc/mdstat.

 %# raidstart /dev/md0

 %# cat /proc/mdstat

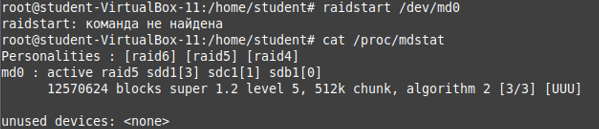
Personalities : [raid5]

read\_ahead 1024 sectors

md0 : active raid5 hdg1[2] hde1[1] hdf2[0]

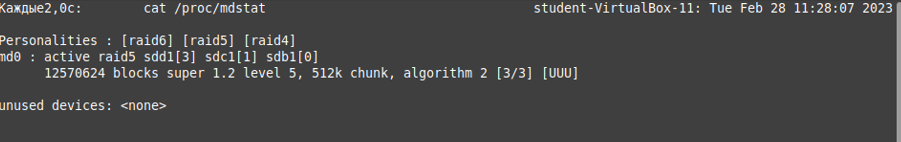
4120448 blocks level 5, 32k chunk, algorithm 3 [3/3] [UUU]

unused devices: <none>



Если в файле информация постоянно изменяется, например, идёт пересборка массива, то постоянно изменяющийся файл удобно просматривать при помощи программы **watch**:

%$ watch cat /proc/mdstat



Проверка целостности программного RAID-массива md0:

echo 'check' >/sys/block/md0/md/sync\_action

Просмотр ошибок в процессе проверки программного RAID-массива по команде check или repair:

cat /sys/block/md0/md/mismatch\_cnt



**Дальнейшая работа с массивом**

### Пометка диска как сбойного

Диск в массиве можно условно сделать сбойным, ключ **--fail**(**-f**):

  %# mdadm /dev/md0 --fail /dev/hde1

 %# mdadm /dev/md0 -f /dev/hde1

### Удаление сбойного диска

Сбойный диск можно удалить с помощью ключа **--remove** (**-r**):

 %# mdadm /dev/md0 --remove /dev/hde1

 %# mdadm /dev/md0 -r /dev/hde1

### Добавление нового диска

Добавить новый диск в массив можно с помощью ключей **--add** (-a) и **--re-add**:

  %# mdadm /dev/md0 --add /dev/hde1

 %# mdadm /dev/md0 -a /dev/hde1

### Сборка существующего массива

Собрать существующий массив можно с помощью **mdadm** **--assemble**. Как дополнительный аргумент указывается, нужно ли выполнять сканирование устройств, и если нет, то какие устройства нужно собирать.

%# mdadm --assemble /dev/md0 /dev/hde1 /dev/hdf2 /dev/hdg1

 %# mdadm --assemble --scan

### Расширение массива

Расширить массив можно с помощью ключа **--grow (-G)**. Сначала добавляется диск, а потом массив расширяется:

 %# mdadm /dev/md0 --add /dev/hdh2

Проверить, что диск (раздел) добавился:

 %# mdadm --detail /dev/md0

 %# cat /proc/mdstat

Если раздел действительно добавился, возможно расширить массив:

 %# mdadm -G /dev/md0 --raid-devices=4

Опция **--raid-devices** указывает новое количество дисков используемое в массиве. Например, было 3 диска, а теперь необходимо расширить до 4-х - указывается 4.

Рекомендуется задать файл бэкапа на случай прерывания перестроения массива, например добавить:

--backup-file=/var/backup

При необходимости, можно регулировать скорость процесса расширения массива, указав нужное значение в файлах

/proc/sys/dev/raid/speed\_limit\_min

/proc/sys/dev/raid/speed\_limit\_max

Необходимо убедиться, что массив расширился:

 %# cat /proc/mdstat

Нужно обновить конфигурационный файл с учётом сделанных изменений:

 %# mdadm --detail --scan >> /etc/mdadm/mdadm.conf

 %# vi /etc/mdadm/mdadm.conf

**Переименование массива**

Для начала необходимо отмонтировать и остановить массив:

 %# umount /dev/md0

 %# mdadm --stop /dev/md0

Затем необходимо пересобрать как md5 каждый из разделов sd[abcdefghijk]1

 %# mdadm --assemble /dev/md5 /dev/sd[abcdefghijk]1 --update=name

или так

 %# mdadm --assemble /dev/md5 /dev/sd[abcdefghijk]1 --update=super-minor

**Удаление массива**

Для начала необходимо отмонтировать и остановить массив:

 %# umount /dev/md0

 %# mdadm -S /dev/md0

Затем необходимо затереть superblock каждого из составляющих массива:

 %# mdadm --zero-superblock /dev/hde1

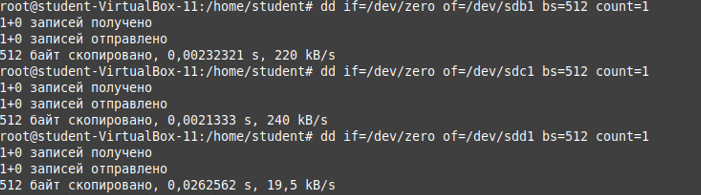
 %# mdadm --zero-superblock /dev/hdf2

Если действие выше не помогло, то выполняется следующая комманда:

 %# dd if=/dev/zero of=/dev/hde1 bs=512 count=1

 %# dd if=/dev/zero of=/dev/hdf2 bs=512 count=1

****

****

**Вывод**

В данной работе была произведена настройка резервирования информации при помощи RAID массивов. Были изучены и освоены методы избыточного дублирования информации в ОС Linux.