



ОНЛАЙН-ОБРАЗОВАНИЕ



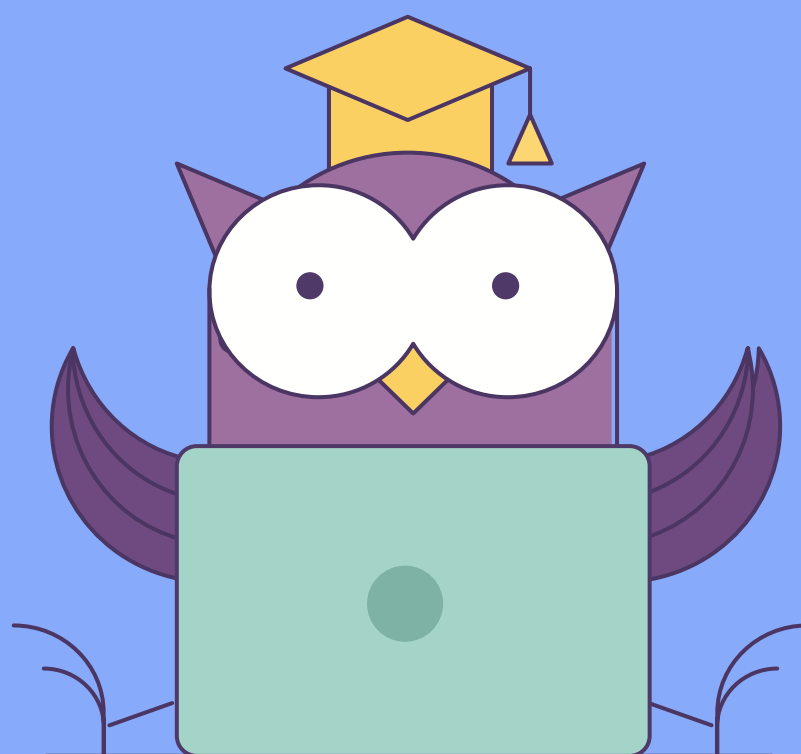
Дисковая подсистема: RAID

Курс «Администратор Linux»

Занятие № 2



Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте ☐ + если все хорошо
Ставьте ☐ - если есть проблемы

Цели вебинара

По итогам занятия вы сможете:

- рассказать какие бывают RAID массивы и чем они отличаются
- получить информацию о дисковой подсистеме на любом сервере с ОС Linux
- Описать процесс сборки и восстановления софт рейда

Дисковая подсистема

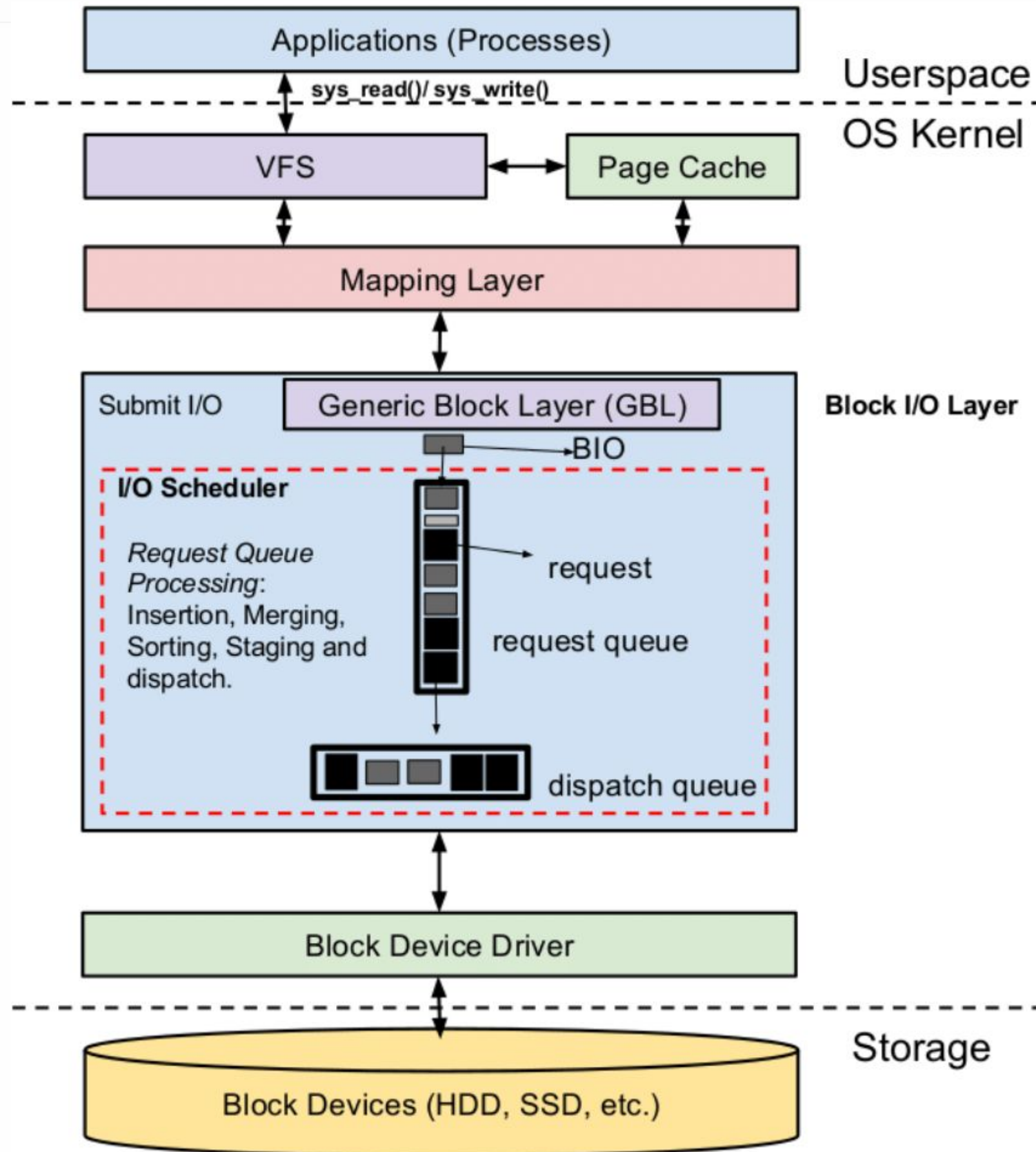
VFS (Virtual File System) - абстрактная файловая система служащая для интерпретации системных вызовов приложений в язык понятный различным файловым системам. Своеобразная унификация.

Самый банальный пример, это когда приложение обращается к локальной и сетевой файловой системе не замечая практической разницы. За него это делает VFS

Блочные устройства в UNIX и Linux — это устройства хранения с произвольным доступом, над которыми размещаются файловые системы. Обеспечивает интерфейс к устройству.

```
[root@mdadm ~]$ ls -la /dev/sd*  
brw-rw----. 1 rootdisk  8,  0 Oct 22 05:32 sda  
brw-rw----. 1 rootdisk  8,  1 Oct 22 05:32 sda1  
brw-rw----. 1 rootdisk  8,  2 Oct 22 05:32 sda2  
brw-rw----. 1 rootdisk  8,  3 Oct 22 05:32 sda3
```

Linux Kernel IO Arch



Блочное устройство обеспечивает обмен блоками данных.

Блок (chunk)— это единица данных фиксированного размера. Размер блока определяется ядром, но чаще всего он совпадает с размером страницы аппаратной архитектуры, и для 32-битной архитектуры x86 составляет 4096 байт.

```
[root@mdadm ~]$ fdisk -l /dev/sda | grep -i sector
```

```
Disk /dev/sda: 42.9 GB, 42949672960 bytes, 83886080 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

Дисковая подсистема

В настоящее время существует два подхода к организации **/dev** –

- **Статическая организация** - специальные файлы для всех возможных устройств вне зависимости от того, загружен драйвер соответствующего устройства или нет. Уходящая в прошлое организация.
- **Динамическая организация** - специальные файлы в **/dev** создаются по мере инициализации устройств и загрузки драйверов, и удаляются при выгрузке соответствующего драйвера или удалении устройства.

Процесс работы со статическим **/dev** особых проблем не вызывает – системный администратор при необходимости просто создает отсутствующие файлы командой **mknod**.

Дисковая подсистема - DevFS

Ядро монтирует к каталогу /dev специальную файловую систему, называемую devfs:

- Целиком находится в оперативной памяти
- Драйвер devfs динамически создает специальный файл
- Динамически же удаляет его

Дисковая подсистема - Udev

В отличие от devfsd, который требовал поддержки со стороны ядра, udev такой поддержки не требует.

- Постоянно закрепленные за устройствами имена, которые не зависят от того, какое положение они занимают в дереве устройств.
- Уведомление внешних по отношению к ядру программ, если устройство было заменено
- Гибкие правила именования устройств.

Дисковая подсистема - devtmpfs

На текущий момент используется devtmpfs + udev (systemd+udevd)

После монтирования корневой файловой системы, этот экземпляр tmpfs перемонтируется ядром в каталог /dev

devtmpfs отвечает за заполнение каталога /dev

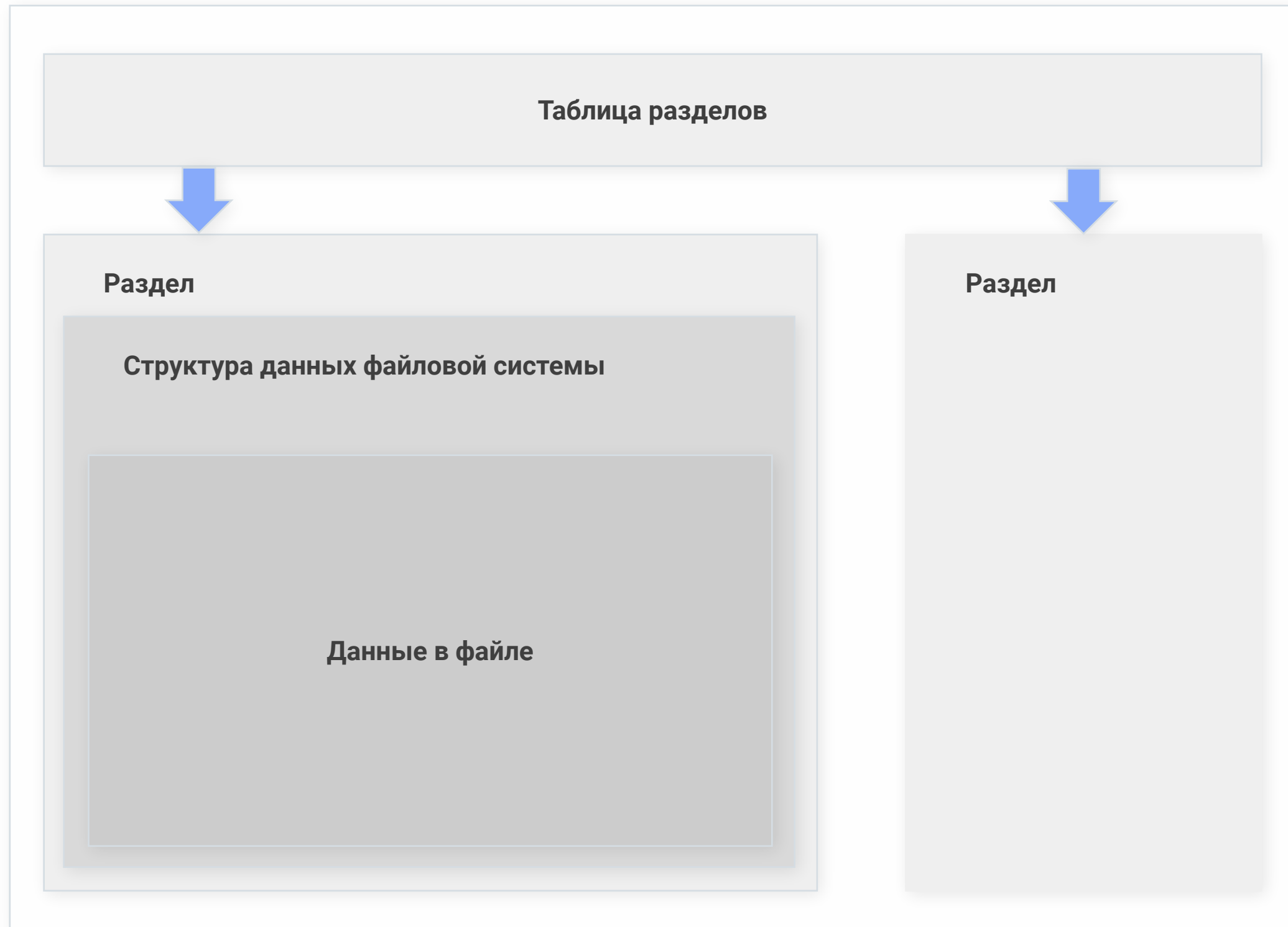
udev за права доступа, необходимые симлинки и запуск скриптов пользователя



Таблицы разделов

Таблицы разделов: MBR

- **Таблица разделов** - логически выделенная часть жесткого диска фиксированного размера видимая ОС как отдельное блочное устройство
- Физически информация о разделах находится на жестком диске в разных местах в зависимости от типа таблицы.
- Мы будем рассматривать MBR (Master Boot Record) и GPT (Guide Partition Table)



Таблицы разделов: MBR

Преимущества

- + Самый популярный и совместимый

Недостатки

- Максимум 4 первичных раздела на диске
- Если первые сектора диска повреждены, диск перестает читаться
- Максимальный раздел – 2.2 Tb

Таблицы разделов: MBR

- 512 байт
- Смещение Длина Описание
000h 446 Код загрузчика
1BEh 64 Таблица разделов
 16 Раздел 1
1CEh 16 Раздел 2
1DEh 16 Раздел 3
1EEh 16 Раздел 4
1FEh 2 Сигнатура (55h AAh)

Таблицы разделов

16 байтный блок раздела

| | | |
|-----|---|---|
| 00h | 1 | Признак активности раздела |
| 01h | 1 | Начало раздела - головка |
| 02h | 1 | Начало раздела - сектор (биты 0-5), дорожка (биты 6,7) |
| 03h | 1 | Начало раздела - дорожка (старшие биты 8,9 хранятся в байте номера сектора) |
| 04h | 1 | Код типа раздела |
| 05h | 1 | Конец раздела - головка |
| 06h | 1 | Конец раздела - сектор (биты 0-5), дорожка (биты 6,7) |
| 07h | 1 | Конец раздела - дорожка (старшие биты 8,9 хранятся в байте номера сектора) |
| 08h | 4 | Смещение первого сектора |
| 0Ch | 4 | Количество секторов раздела |

Таблицы разделов

- Backup MBR

```
[root@mdadm ~]$ dd if=/dev/sdX of=/tmp/sda.mbr bs=512 count=1
```

- Восстановление MBR с сохранением таблицы разделов

```
[root@mdadm ~]$ dd if=/tmp/sda.mbr of=/dev/sdb bs=446 count=1
```

- Утилиты для работы

fdisk, sfdisk - создание/удаление 파티ций

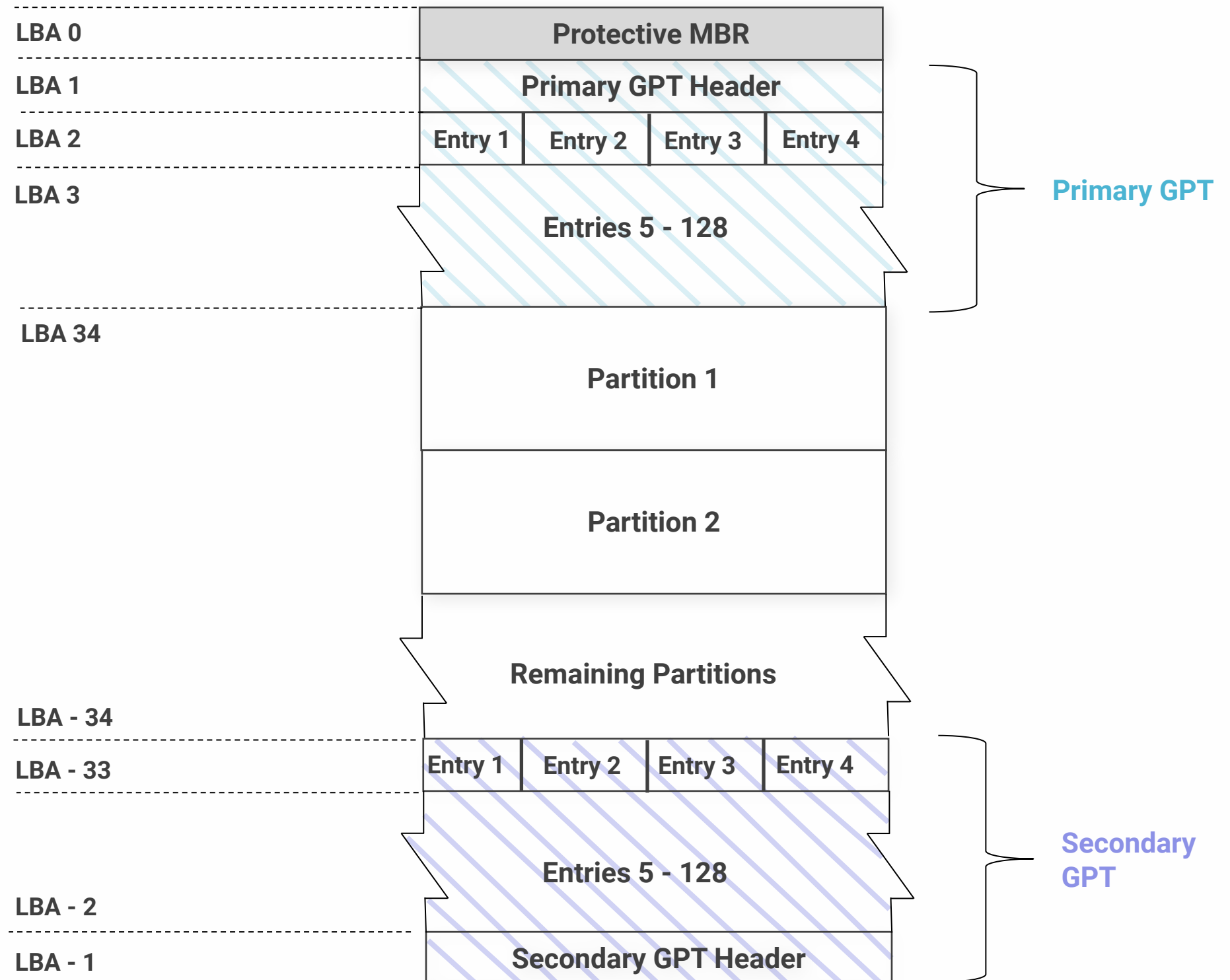
Преимущества

- + Неограниченное кол-во разделов
- + Очень большие ограничения на объем
- + Раздел зарезервирован, хранятся контрольные CRC-суммы, в случае проблем возможно восстановление

Недостатки

- Не поддерживается старыми системами

Таблицы разделов



Таблицы разделов

- Backup GPT

```
[root@mdadm ~]$ sgdisk --backup=sda.gpt.bkp /dev/sda
```

- Восстановление GPT

```
[root@mdadm ~]$ sgdisk --load-backup=sda.gpt.bkp /dev/sda
```

- Утилиты для работы:

gdisk, sgdisk

Полезные команды

Утилиты для работы с партициями

- `gdisk`
- `parted`
- `partx`
- `partprobe`

Утилиты для работы с информацией о железе

- `smartctl`
- `hdparm`
- `dmesg`
- `dmidecode`

RAID's

Задачи стоящие перед дисковой подсистемой

1. Максимальный объем
2. Максимальная скорость
3. Максимальная надежность
4. Минимальная стоимость

Практически все эти задачи решаются разными уровнями RAID ([Redundant Array of Independent/Inexpensive Disks](#)), но, к сожалению, не все сразу.

Задачи стоящие перед дисковой подсистемой

- Например, RAID-0 дает максимум объема, при отсутствующей надежности, а RAID-1 — максимальную надежность, при отсутствии выигрыша в пространстве.
- В случае RAID-0 и RAID-1 цифра в названии равна вероятности восстановления в случае отказа диска в массиве.
- Но есть и другие уровни RAID — 5,6,10, все это в том или ином виде — компромиссы.

- Аппаратный - отдельный контроллер со своим процессором, cache, портами, батареейкой.
- Программный (он же софтверный, он же mdadm) - все через CPU.
- Интегрированный - встроенный в материнскую плату с отдельным чипом для вычисления XOR.

Преимущества

- + Аппаратное решение, не влияющее на производительность основной системы
- + Выделенный CPU
- + Выделенная память для Кэшей
- + Возможность использовать BBU (Battery Backup Unit)
- + Возможность подключения большого количества дисков
- + Прозрачность для загрузчиков (возможность грузиться с любого массива)

Недостатки

- Высокая стоимость
- Высокая сложность
- Разнообразие интерфейсов управления и драйверов
- Низкая «мобильность» /переносимость
- Привязка к железу
- Бóльший простой по времени при аварии
- Очень дорогой ремонт, необходимость закупать впрок контроллеры, которые потом могут прекратить выпускать

Software RAID (mdraid)

Преимущества

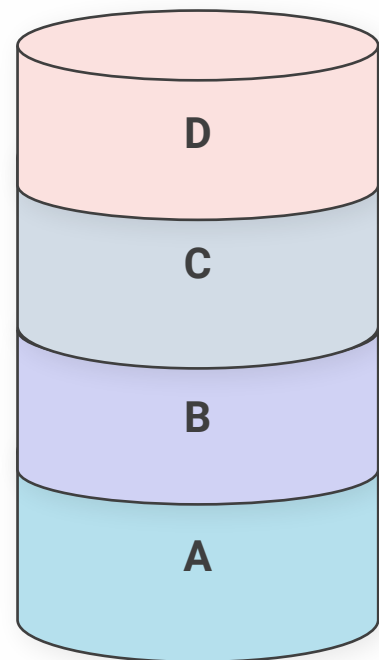
- + Бесплатно
- + Отсутствие привязки к конкретному железу
- + Прозрачность конфигурации
- + Примерно одинаковый интерфейс управления в любом linux.
- + Легкая переносимость между компьютерами.
- + Гибкость конфигурации

Недостатки

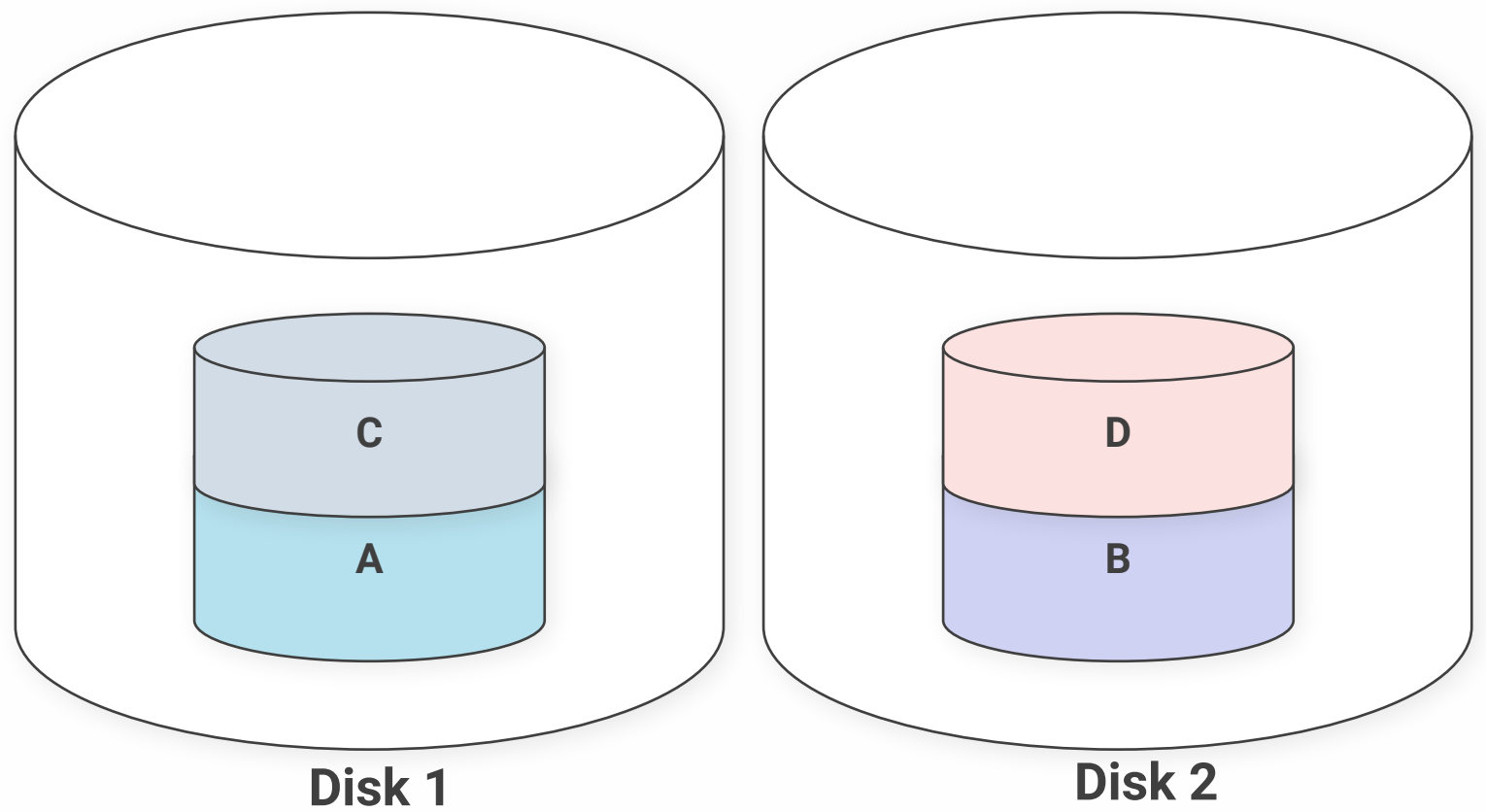
- Отсутствие BBU
- Отсутствие выделенного кэша
- Отсутствие службы поддержки :-)

RAID-0

Data Chunks



RAID-0



$$V_t = V_1 * N$$

RAID-0

Преимущества

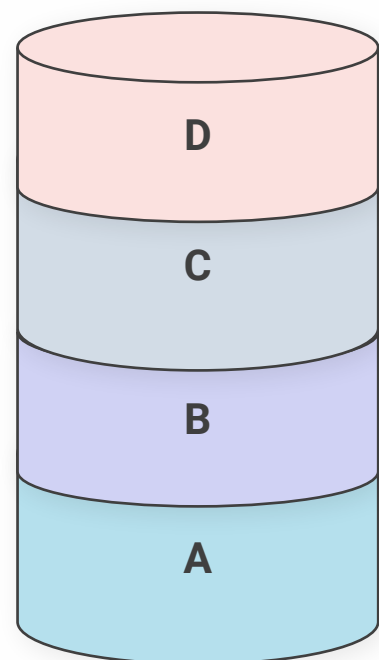
- + Самое быстрое чтение
- + Очень простой
- + Максимальная эффективность использования дискового пространства

Недостатки

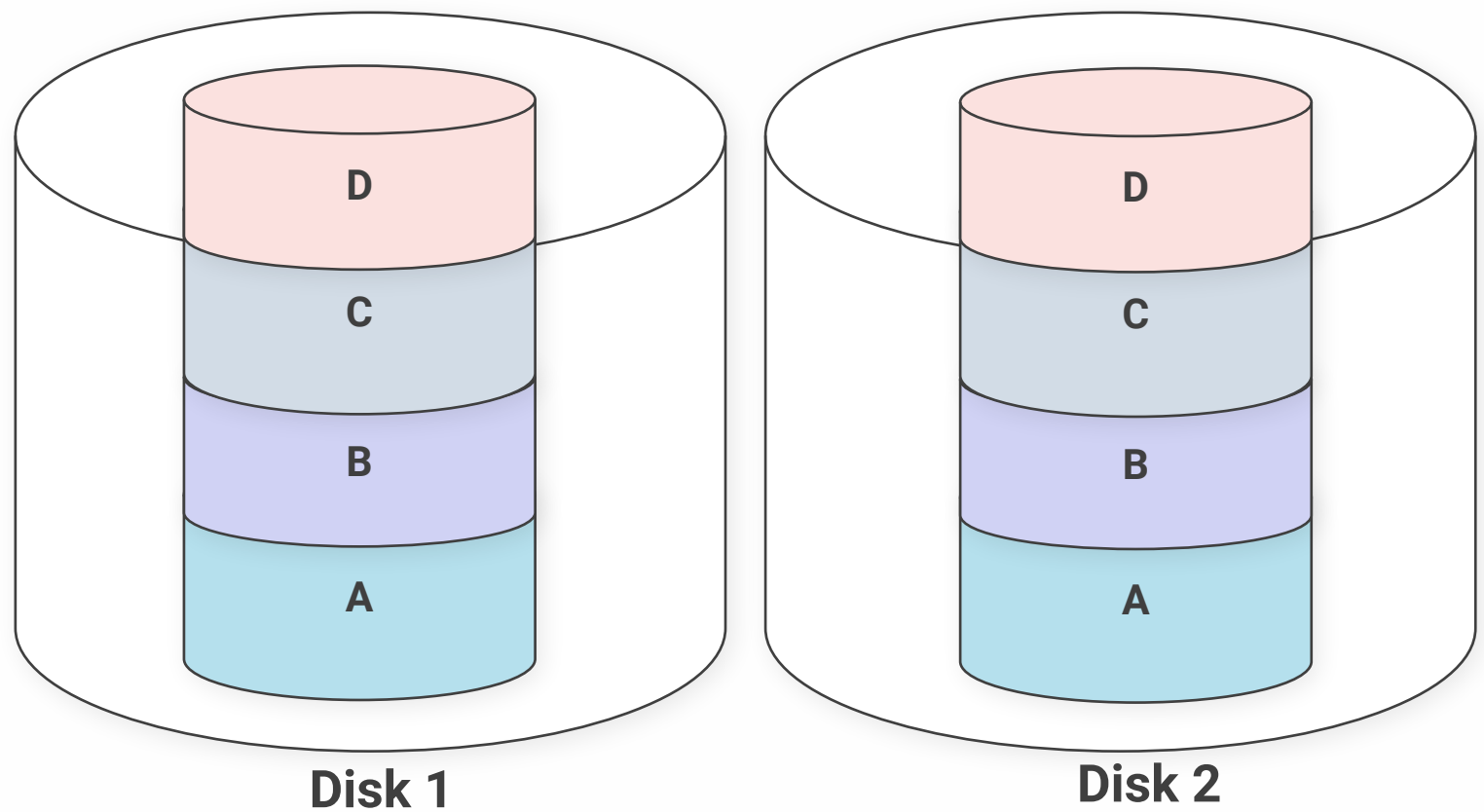
- Не «настоящий» RAID, нет отказоустойчивости: отказ одного диска влечет за собой потерю всех данных массива

RAID-1

Data Chunks



RAID-1



$$V_t = V_1$$

RAID-1

Преимущества

- + Простота реализации
- + Простота восстановления:
перекопировать все данные с
«выжившего» диска
- + Высокая скорость на чтение

Недостатки

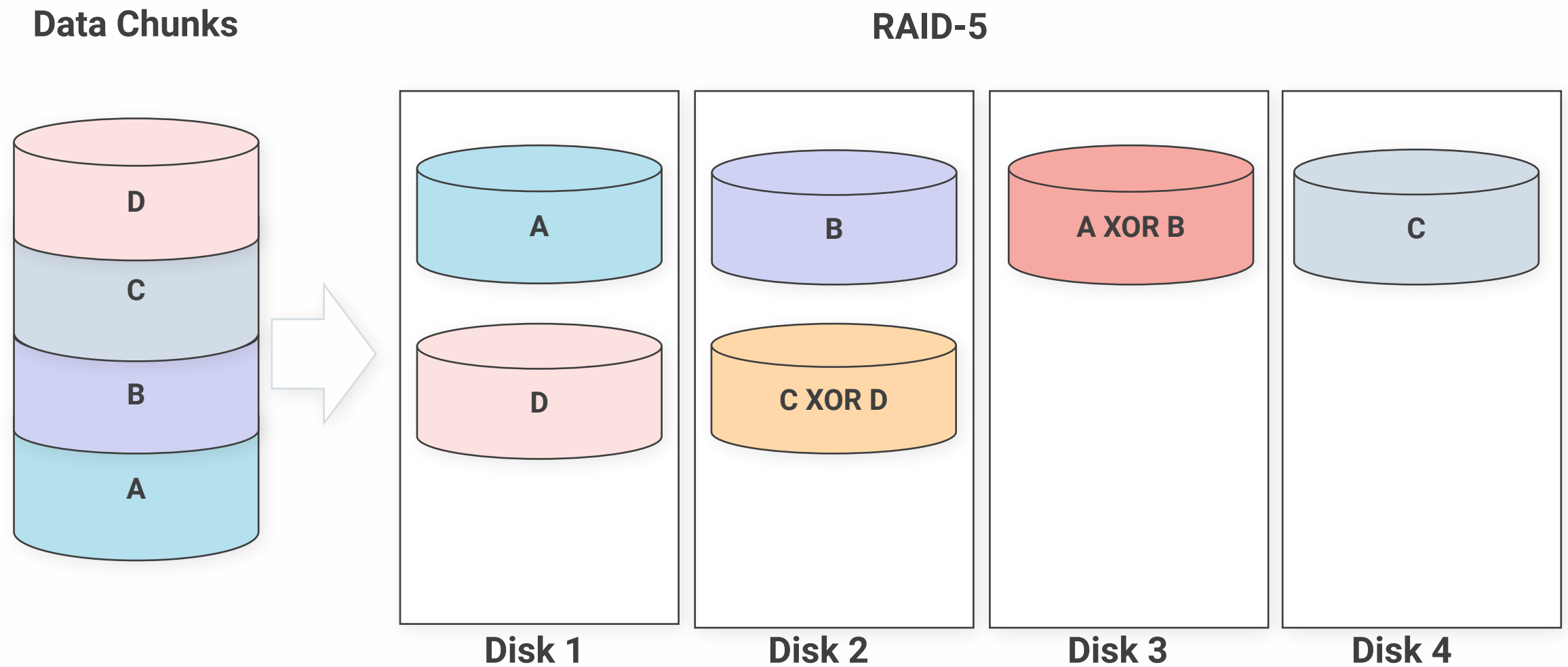
- Высокая стоимость на единицу
объема: 100% избыточность

Логическая операция исключающее ИЛИ выполняется с двумя битами (a и b). Результат выполнения логической операции XOR будет равен 1 (единице), если один из битов a или b равен 1 (единице), во всех остальных случаях, результат равен 0 (нулю).

XOR обладает особенностью, которая даёт возможность заменить любой операнд результатом, и, применив алгоритм xor, получить в результате недостающий операнд.

Например: $a \text{ xor } b = c$ (где a, b, c — три диска рейд-массива), в случае если a откажет, мы можем получить его, поставив на его место c и проведя XOR между c и b: $c \text{ xor } b = a$.

RAID-5



$$V_t = V_1 * (N-1)$$

RAID-5

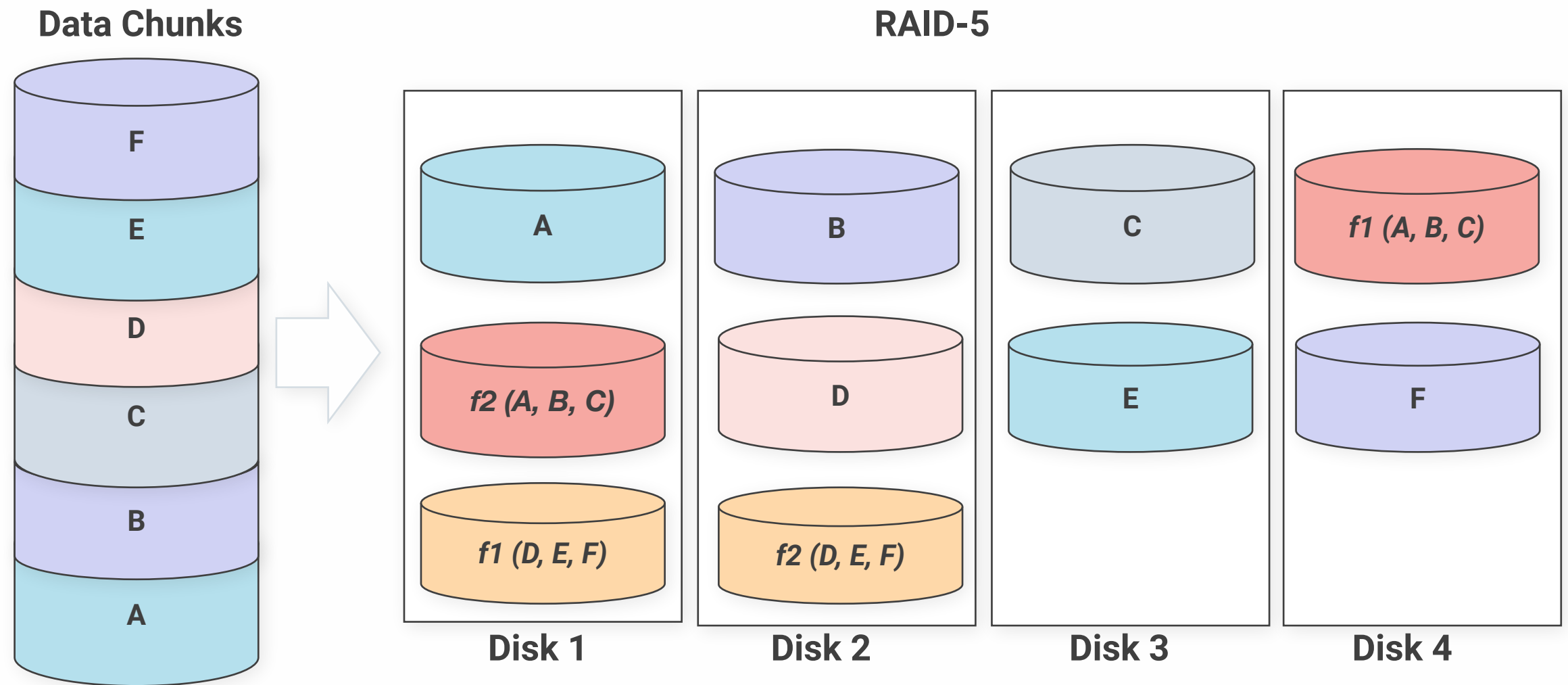
Преимущества

- + Высокая скорость записи данных
- + Достаточно высокая скорость чтения данных
- + Высокая производительность при большой интенсивности запросов чтения/записи данных
- + Малые накладные расходы для реализации избыточности

Недостатки

- Низкая скорость чтения/записи данных малого объема при единичных запросах
- Достаточно сложная реализация
- Сложное восстановление данных

RAID-6



$$V_t = V_1 * (N-2)$$

RAID-6

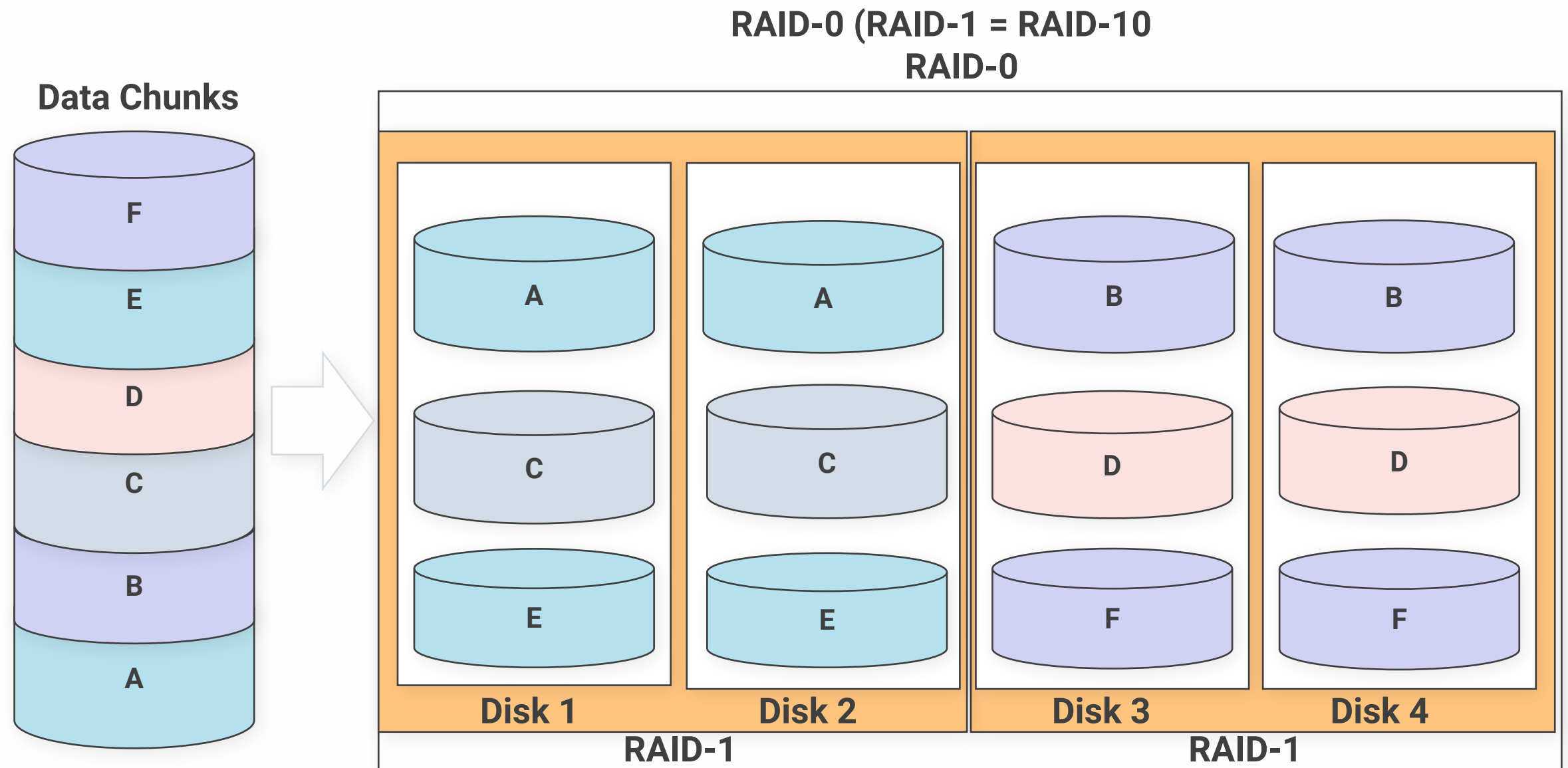
Преимущества

- + Высокая отказоустойчивость
- + Достаточно высокая скорость обработки запросов
- + Относительно малые накладные расходы для реализации избыточности

Недостатки

- Очень сложная реализация
- Сложное восстановление данных
- Очень низкая скорость записи данных

RAID-10



$$V_t = V_1 * N/2$$

RAID-10

Преимущества

- + Самая высокая отказоустойчивость
- + Самая высокая производительность
- + Сочетает в себе преимущества R0 и R1

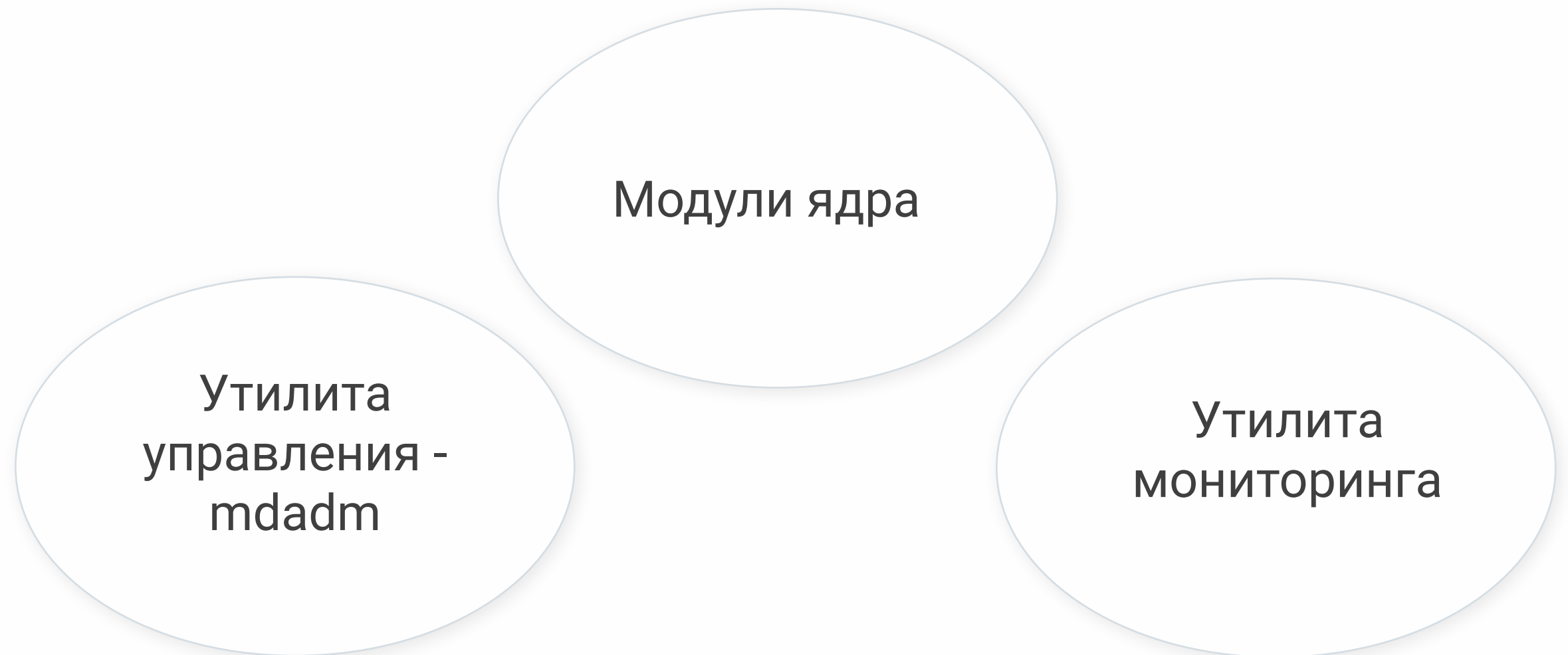
Недостатки

- Двойная стоимость пространства

Экзотические RAID

- 1E (запись по очереди в один из дисков, копия в следующий)
- 5+0 (RAID0 из RAID5)
- 5+1 (RAID1 из RAID5)
- 6+0 (RAID0 из RAID6)
- 6+1 (RAID1 из RAID6)

mdadm



mdraid: элементы

Блочные устройства

- разделы
- диски
- тома lvm

Метаданные

- 0.9, 1.0 - конец устройства
(необходимо для загрузки в некоторых случаях)
- 1.1 - начало
- 1.2 - 4K от начала устройства

mdraid

Массивы mdraid можно создавать из любых блочных устройств:

- Дисков,
- Разделов,
- Томов lvm.

Наиболее **безопасно** создавать RAID поверх разделов, это может помочь сгладить разный размер дисков, позволит на одном наборе дисков создать разные RAID.

Пример организации дисков



Как использовать mdraid

- Подготовка к созданию, “занулить суперблок”

mdadm --zero-superblock \$dev_list

- Создание массива

mdadm --create \$raiddev -l \$level -n \$numdev \$dev_list

- Остановка массива

mdadm -S \$raiddev

- Информация о массиве

mdadm --detail \$raiddev

- Генерация данных для конфигурационного файла

mdadm --examine --scan

mdadm --detail --scan

Как использовать mdraid

- Информация о массиве

```
cat /proc/mdstat
```

- Запуск/остановка проверки

```
echo (check|idle) > /sys/block/md${N}/device/action
```

- Изменение ограничений скорости ребилда

```
# grep . /proc/sys/dev/raid/speed_limit_m*
```

```
/proc/sys/dev/raid/speed_limit_max:200000
```

```
/proc/sys/dev/raid/speed_limit_min:1000
```

```
# echo 10000 > /proc/sys/dev/raid/speed_limit_min
```



BONUS

- Механизм перемещения редко используемых частей оперативной памяти на специально размеченное пространство на блочном устройстве, освобождая при этом ОЗУ
- Другой его функционал в том, что обеспечить отзывчивость системы при переполнении виртуальной памяти и для отсрочки прихода OOM-а

`OOMScoreAdjust=-1000`

`cat /proc/vm/swappiness` - процент свободной оперативной памяти, при котором начинается использование свапа

`echo "vm.swappiness=10" | sudo tee -a /etc/sysctl.conf`

- Какой размер выбрать?
- Создадим файл подкачки:

```
fallocate -l 1G /swapfile
```

```
mkswap /swapfile
```

```
swapon /swapfile
```

```
echo "/swapfile swap swap defaults 0 0" | tee -a /etc/fstab
```

- RAID Levels (10 - для скорости и надежности)
- RAID Controller Cache - например обновление одних и тех же мест в кеше сократит интенсивность записи
- RAID Stripe Size 16 KB, 64 KB, 256 KB or 512 KB - в зависимости от паттерна использования
- `mount -o noatime,nodiratime`
- `sysctl -w vm.dirty_background_ratio=25` - в процентах. Частота сброса данных на диск
- `sysctl -w vm.dirty_ratio=20` - в процентах. Величина наполнения кеша до сброса на диск

- EXT4/XFS
- Минимизировать запись на диск:
 - noatime
 - не держать на нем логи
 - в зависимости от паттерна использования
- `systemctl enable fstrim.timer`
- I/O Scheduler: `noop`

Ваши вопросы?

Домашнее задание: работа с mdadm

Задание:

добавить в Vagrantfile еще дисков

сломать/починить raid

собрать R0/R5/R10 на выбор

прописать собранный рейд в конф, чтобы рейд собирался при загрузке

создать GPT раздел и 5 партиций

в качестве проверки принимаются - измененный Vagrantfile, скрипт для создания рейда

* доп. задание - Vagrantfile, который сразу собирает систему с подключенным рейдом

** перенести работающую систему с одним диском на RAID 1. Даунтайм на загрузку с нового диска предполагается. В качестве проверки принимается вывод команды lsblk до и после и описание хода решения (можно воспользоваться утилитой Script).

**Заполните, пожалуйста,
опрос в ЛК о занятии**

**Спасибо
за внимание!**

До встречи в Slack и на вебинаре

