

У меня обычно по разрозненным шпаргалкам располагаются различные полезные команды. Решил собрать воедино все, что касается работы с дисками в debian, их настройка, добавление, подключение, расширение и т.д. и поделиться с вами. Все эти команды и приемы будут актуальны практически для любого дистрибутива linux.

Если у вас есть желание научиться работать с роутерами микротик и стать специалистом в этой области, рекомендую по программе, основанной на информации из официального курса **MikroTik Certified Network Associate**. Курс стоящий, все подробности читайте по ссылке. Есть бесплатные курсы.

#### Содержание:

- 1 Цели статьи
- 2 Введение
- 3 Информация о дисках
- 4 Посмотреть свободное место на диске
- 5 Подключить сетевой диск
- 6 Подключить и примонтировать диск
  - 6.1 Разметка диска
  - 6.2 Создание файловой системы ext4, xfs
- 7 Работа в debian с lvm
- 8 Расширение диска
- 9 Проверка диска на ошибки и bad blocks
- 10 Протестировать скорость диска
- 11 Проверить нагрузку на диск
- 12 Создание диска из оперативной памяти (озу)
- 13 Подключение Яндекс Диска
- 14 Заключение

Данная статья является частью единого цикла статьей про сервер Debian.

## Цели статьи

1. Рассмотреть наиболее актуальные вопросы по работе с дисками.
2. Поделиться своими примерами утилит и команд.
3. На конкретных примерах рассмотреть проверку, добавление, расширение дисков.

## Введение

Данная статья будет написана на базе операционной системы Debian. Но в целом, это все будет актуально для любого linux дистрибутива. Я решил пройти по всем основным темам, касающимся работы с дисками. Постарался выстроить повествование последовательно, от простого к сложному, насколько это возможно в данной тематике.

## Информация о дисках

Информацию о дисках в системе можно получить различными способами. Зачастую, интересна информация не только о физических дисках, но и о разделах. Начнем все же с физических дисков. Подробную информацию о железе, в том числе и о дисках, выдает программа **hwinfo**. В базовой системе ее нет, нужно поставить отдельно.

```
# apt install hwinfo
```

Теперь смотрим информацию о дисках:

```
# hwinfo --disk
```

Программа выведет железную информацию о всех дисках в системе. Вот пример вывода одного из физических дисков.



```
root@prox:~# hwinfo --disk
21: IDE 100.0: 10600 Disk
[Created at block.245]
Unique ID: Rw8Y.UCwBgLTI0g8
Parent ID: w7Y8.KlSEapPrOC6
SysFS ID: /class/block/sdd
SysFS BusID: 1:0:0:0
SysFS Device Link: /devices/pci0000:00/0000:00:1f.2/ata1/host1/target1:0:0:1:0:0:0
Hardware Class: disk
Model: "WDC WD30EFRX-68E"
Vendor: "WDC"
Device: "WD30EFRX-68E"
Revision: "0A80"
Serial ID: "WD-WMC4N0941438"
Driver: "ahci", "sd"
Driver Modules: "ahci"
Device File: /dev/sdd
Device Files: /dev/sdd, /dev/disk/by-path/pci-0000:00:1f.2-ata-1, /dev/disk/by-id/wwn-0x50014ee003afdfff5, /dev/disk/by-uuid/3266ed3a-fe16-483a-b737-b59ab7205d96, /dev/disk/by-id/ata-WDC_WD30EFRX-68EUZNo_WD-WMC4N0941438
Device Number: block 8:48-8:63
Geometry (Logical): CHS 364801/255/63
Size: 5860533168 sectors a 512 bytes
Capacity: 2794 GB (3000592982016 bytes)
Config Status: cfg=new, avail=yes, need=no, active=unknown
Attached to: #3 (SATA controller)
```

serveradmin.ru

На выходе максимально подробная информация о диске — вендор, модель, серийный номер, метки диска в системе и много другое. Программа показывает принадлежность диска к рейд массиву, что бывает удобно. Вот вывод информации о диске из рейда adaptec.



```
24: SCSI 02.0: 10600 Disk
[Created at block.245]
Unique ID: LUEV.bFKdksmCzL5
Parent ID: VCu0.4TRixbFmTfl
SysFS ID: /class/block/sdc
SysFS BusID: 0:0:2:0
SysFS Device Link: /devices/pci0000:00/0000:00:01.0/0000:01:00.0/host0/target0:0:2/0:0:2:0
Hardware Class: disk
Model: "Adaptec rlmini"
Vendor: "Adaptec"
Device: "rlmini"
Revision: "V1.0"
Serial ID: "B13B3B2E"
Driver: "aacraid", "sd"
Driver Modules: "aacraid"
Device File: /dev/sdc (/dev/sg2)
Device Files: /dev/sdc, /dev/disk/by-id/scsi-22e3b3bb100d00000, /dev/disk/by-path/pci-0000:01:00.0-scsi-0:0:2:0
Device Number: block 8:32-8:47 (char 21:2)
Geometry (Logical): CHS 60831/255/63
Size: 977250304 sectors a 512 bytes
Capacity: 465 GB (500352155648 bytes)
Config Status: cfg=new, avail=yes, need=no, active=unknown
Attached to: #10 (RAID bus controller)
```

**Название массива в конфигурации adaptec**

serveradmin.ru

Следующая программа, которую я использую, чтобы посмотреть информацию о физическом диске в debian — **smartmontools**. Она более громоздкая, тянет за собой кучу зависимостей, но зато умеет по расписанию следить за смартом дисков, слать уведомления в случае проблем. Фактически это не утилита, а готовый сервис. Ставится так.

```
# apt install smartmontools
```

Возможностей у программы много. Я в рамках данной статьи, покажу только, как посмотреть информацию о диске, в том числе параметры SMART.

```
# smartctl -i /dev/sda
```





```
root@prox:~# smartctl -i /dev/sdd
smartctl 7.0 2018-12-30 r4883 [x86_64-linux-5.0.18-1-pve] (local build)
Copyright (C) 2002-18, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Model Family:      Western Digital Red
Device Model:      WDC WD30EFRX-68EUZNO
Serial Number:     WD-WMC4N0941438
LU WWN Device Id: 5 0014ee 003afdff5
Firmware Version: 80.00A80
User Capacity:     3,000,592,982,016 bytes [3.00 TB]
Sector Sizes:     512 bytes logical, 4096 bytes physical
Rotation Rate:    5400 rpm
Device is:         In smartctl database [for details use: -P show]
ATA Version is:   ACS-2 (minor revision not indicated)
SATA Version is:  SATA 3.0, 6.0 Gb/s (current: 6.0 Gb/s)
Local Time is:    Fri Sep  6 15:36:43 2019 MSK
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
```

SMART диска.

```
# smartctl -A /dev/sdd
```



```
root@prox:~# smartctl -A /dev/sdd
smartctl 7.0 2018-12-30 r4883 [x86_64-linux-5.0.18-1-pve] (local build)
Copyright (C) 2002-18, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===
SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x002f   200   200   051  Pre-fail Always    -         37
  3 Spin_Up_Time            0x0027   182   181   021  Pre-fail Always    -        5883
  4 Start_Stop_Count        0x0032   100   100   000  Old_age  Always    -         148
  5 Reallocated_Sector_Ct   0x0033   200   200   140  Pre-fail Always    -          0
  7 Seek_Error_Rate         0x002e   200   200   000  Old_age  Always    -          0
  9 Power_On_Hours          0x0032   076   076   000  Old_age  Always    -       17983
 10 Spin_Retry_Count        0x0032   100   100   000  Old_age  Always    -          0
 11 Calibration_Retry_Count 0x0032   100   100   000  Old_age  Always    -          0
 12 Power_Cycle_Count       0x0032   100   100   000  Old_age  Always    -         148
192 Power-Off_Retract_Count 0x0032   200   200   000  Old_age  Always    -          28
193 Load_Cycle_Count        0x0032    01    001   000  Old_age  Always    -     1313971
194 Temperature_Celsius     0x0022   120   097   000  Old_age  Always    -          30
196 Reallocated_Event_Count 0x0032   200   200   000  Old_age  Always    -          0
197 Current_Pending_Sector  0x0032   200   200   000  Old_age  Always    -          2
198 Offline_Uncorrectable   0x0030   100   253   000  Old_age  Offline   -          0
199 UDMA_CRC_Error_Count    0x0032   200   200   000  Old_age  Always    -          0
200 Multi_Zone_Error_Rate   0x0008   100   253   000  Old_age  Offline   -          0
```

С диском некоторые проблемы, судя по смарту.

Физические диски посмотрели, теперь посмотрим на список дисков с привязкой к логическим разделам. Я обычно использую 2 утилиты для этого — **lsblk** и **fdisk**.

```
# lsblk -a
```

```
root@debian10:~# lsblk -a
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
fd0                  2:0    1    4K  0 disk
sda                  8:0    0   20G  0 disk
├─sda1                8:1    0  243M  0 part /boot
├─sda2                8:2    0     1K  0 part
├─sda5                8:5    0  19.8G  0 part
│   └─debian10--vg-root 254:0    0  19.8G  0 lvm  /
sr0                 11:0    1 1024M  0 rom
root@debian10:~#
```

Fdisk позволяет сразу посмотреть более подробную информацию о разделах.

```
# fdisk -l | grep /dev/sd
```



```
root@prox:~# fdisk -l | grep /dev/sd
Partition 2 does not start on physical sector boundary.
Disk /dev/sda: 931 GiB, 999643152384 bytes, 1952428032 sectors
/dev/sda1 2048 4095 2048 1M BIOS boot
/dev/sda2 4096 528383 524288 256M EFI System
/dev/sda3 528384 1952427998 1951899615 930.8G Linux LVM
Disk /dev/sdb: 930 GiB, 998579896320 bytes, 1950351360 sectors
/dev/sdb1 2048 1950351326 1950349279 930G Linux LVM
Disk /dev/sdc: 466 GiB, 500352155648 bytes, 977250304 sectors
/dev/sdc1 2048 977250270 977248223 466G Linux LVM
Disk /dev/sdd: 2.7 TiB, 3000592982016 bytes, 5860533168 sectors
Disk /dev/sde: 55.9 GiB, 60021399040 bytes, 117229295 sectors
/dev/sde1 2048 117229261 117227214 55.9G Linux LVM
Partition 2 does not start on physical sector boundary.
```

Мне обычно хватает этих команд, чтобы получить полную информацию о дисках и разделах на них.

## Посмотреть свободное место на диске

Рассмотрим теперь вопрос, как удобнее всего смотреть свободное место на диске. Тут особо вариантов нет — используется известная и популярная утилита **df**.

```
# df -h
```





```
root@debian10:~# df -h
Filesystem                Size      Used Avail Use% Mounted on
udev                     447M          0  447M   0% /dev
tmpfs                     93M       2.8M    90M   3% /run
/dev/mapper/debian10--vg-root 20G       1.1G    18G   6% /
tmpfs                     462M          0  462M   0% /dev/shm
tmpfs                     5.0M          0   5.0M   0% /run/lock
tmpfs                     462M          0  462M   0% /sys/fs/cgroup
/dev/sdal                 236M       48M   176M  22% /boot
tmpfs                     93M          0    93M   0% /run/user/0
root@debian10:~#
```


Команда показывает информацию и заполнении всех примонтированных дисков, в том числе и сетевых, если они присутствуют в системе. Нужно понимать, что эта информация не всегда достоверная. Вот пример такой ситуации — Диск занят на 100% и не понятно чем, df и du показывают разные значения.

Сразу же покажу удобную комбинацию команд, чтобы посмотреть, кто в данной директории занимает больше всего места. Директории выстроятся в список, начиная с самой объемной и далее. В моем примере будут выведены 10 самых больших папок в каталоге.

```
# du . --max-depth=1 -ah | sort -rh | head -10
```



```
root@debian10:/usr# cd /usr
root@debian10:/usr# du . --max-depth=1 -ah | sort -rh | head -10
847M  .
535M  ./lib
236M  ./share
54M   ./bin
23M   ./sbin
64K   ./local
20K   ./include
4.0K  ./src
4.0K  ./libx32
4.0K  ./lib64
root@debian10:/usr#
```



В первой строке будет объем самой директории `/usr`, а далее вложенные в нее. Привожу пример небольшого скрипта, который я люблю использовать, чтобы оценить размер директорий, к примеру, в архиве бэкапов и сохранить информацию в текстовый файл. Актуально, если у вас не настроен мониторинг бэкапов в `zabbix`.

```
echo "=====" >> dir_size.txt
echo "Dirs size `date +%Y-%m-%d_%H-%M`" >> dir_size.txt
echo "=====" >> dir_size.txt
du -s * | sort -nr | cut -f 2- | while read a;do du -hs $a >> dir_size.txt ;done
```

На выходе останется файл `dir_size.txt` следующего содержания.

```
=====  
Dirs size 2019-09-04_18-16  
=====  
3.2T  resad  
2.0T  winshare
```

```
1.7T mail
1.2T doc
957G share
43G web
17G hyperv
6.5G zabbix
5.2G onlyoffice
525M databases
```

В целом, по свободному месту на дисках все. Утилит `df` и `du` достаточно, чтобы закрыть этот вопрос.

## Подключить сетевой диск

Расскажу, как быстро выполнить монтирование наиболее популярных сетевых дисков:

- по smb
- по nfs

В общем случае, подключить сетевой диск по smb можно следующей командой.

```
# mount -t cifs //10.1.4.4/backup /mnt/backup -o user=admin,password=passadmin
```

10.1.4.4/backup	сетевая шара
/mnt/backup	локальная директория, куда монтируем сетевой диск
admin	пользователь
passadmin	пароль

Если команда не отработает и будет ошибка, установите отдельно **cifs-utils**.

```
# apt install cifs-utils
```

С монтированием по smb есть куча нюансов. Сколько различных ошибок я ловил при этом — не счесть. То кодировка не совпадает, то в пароле спец. символы, то шара в домене и надо правильно указать домен. Если в пароле есть спец. символы, пароль можно взять в одинарные кавычки. Домен можно указать через слеш, через плюс, через @. Пробуйте разные варианты, если не получается. Так же имя пользователя с доменом можно тоже брать в кавычки, иногда помогает. Например, вот так — 'domain\admin'. Так же обращайте внимание на версию протокола smb. Ее можно принудительно указывать через опцию `vers`. Так же можно вывести более подробный лог подключения.

```
# mount -t cifs -vvv //10.1.4.4/backup /mnt/backup -o vers=2.1,user=admin,password=passadmin
```

Для подключения сетевого диска по nfs, необходимо установить на машину `nfs-client`, который находится в пакете **nfs-common**.

```
# apt install nfs-common
```

После этого можете подключить сетевую папку по nfs:

```
# mount -t nfs 10.1.4.4:/backup /mnt/backup
```

С nfs, так же как и с cifs, можно вывести расширенный лог и принудительно указать версию.

```
# mount -t nfs -vvv -overs=2 10.1.4.4:/backup /mnt/backup
```

Перед подключением диска можете проверить, а если вам вообще доступ к сетевому диску на сервере:

```
# showmount --exports 10.1.4.4
```

Вы должны увидеть список nfs дисков, к которым у вас есть доступ. Если список пуст, подмонтировать ничего не получится.

## Подключить и примонтировать диск

Расскажу, как подключить и настроить новый физический или виртуальный диск в Debian. У меня в системе есть один диск — `/dev/sda`. Я подключил к ней новый жесткий диск `sdb`. Хочу его отформатировать в файловую систему `xfs` и примонтировать в каталог `/mnt/backup`.

## Разметка диска

Вы можете не создавать разделы на жестком диске, а создать файловую систему прямо по всему диску. Делать так не рекомендуется, хотя лично я никаких проблем не получал при таком использовании диска. С разделами просто удобнее, так как ими можно оперировать — изменять, перемещать, расширять, добавлять и т.д.

Создадим один раздел на диске с помощью утилиты **fdisk**.

```
# fdisk /dev/sdb
```

Если диск чистый, то вам будет предложено создать таблицу разделов на ней. Я обычно gpt создаю. Дальше через графический интерфейс создайте раздел на весь диск и укажите type — Linux filesystem.



```

Disk: /dev/sdb
Size: 10 GiB, 10737418240 bytes, 20971520 sectors
Label: gpt, identifier: 059EF32D-2C1B-2B41-80CD-41307BA40C35

```

Device	Start	End	Sectors	Size	Type
>> /dev/sdb1	2048	20971486	20969439	10G	Linux filesystem

```

Partition UUID: 00E06621-70EC-C74B-B119-6BB5014527B0
Partition type: Linux filesystem (0FC63DAF-8483-4772-8E79-3D69D8477DE4)

```

[ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ] [ Dump ]

Changed type of partition 1.

Как закончите, сохраняйте изменения. Вас попросят подтвердить запись, написав **yes**. Пишите полное слово, не просто у, а именно yes, иначе изменения



не будут записаны. Теперь надо обновить таблицу разделов. Иногда система автоматически не видит новые разделы, а требует перезагрузку, чтобы их увидеть. Перезагружаться не обязательно, достаточно запустить программу **partprobe**.

```
# partprobe -s
```

Если ее нет в системе, то установите пакет **parted**.

```
# apt install parted
```

### Создание файловой системы ext4, xfs

Раздел на новом диске создали. Теперь его надо отформатировать в файловую систему **xfs**. Это не родная система для Debian, поэтому нужно поставить отдельный пакет **xfsprogs** для работы с ней.

```
# apt install xfsprogs
```

Создаем файловую систему xfs на новом диске.

```
# mkfs.xfs /dev/sdb1
```

Если вам нужно создать файловую систему ext4, то ничего ставить дополнительно не нужно. Сразу форматируете раздел в ext4 командой:

```
# mkfs -t ext4 /dev/sdb1
```

Теперь создаем директорию, куда будем монтировать новый диск и подключаем его.

```
# mkdir /mnt/backup  
# mount /dev/sdb1 /mnt/backup
```

Проверяем, что получилось.

```
# df -h
```



```
root@debian10:~# mkdir /mnt/backup
root@debian10:~# mount /dev/sdb1 /mnt/backup
root@debian10:~# df -h
Filesystem                Size      Used Avail Use% Mounted on
udev                     447M          0  447M   0% /dev
tmpfs                     93M       2.8M    90M   3% /run
/dev/mapper/debian10--vg-root 20G       1.1G   18G   6% /
tmpfs                     462M          0  462M   0% /dev/shm
tmpfs                     5.0M          0   5.0M   0% /run/lock
tmpfs                     462M          0  462M   0% /sys/fs/cgroup
/dev/sdal                 236M       48M   176M  22% /boot
tmpfs                     93M          0    93M   0% /run/user/0
/dev/sdb1                 10G       43M    10G   1% /mnt/backup
root@debian10:~#
```

Посмотрим, какую метку получил новый раздел с помощью **blkid**.



```
root@debian10:~# blkid
/dev/sda1: UUID="88c4c0aa-be17-4fd9-b1b7-5c8be142db77" TYPE="ext2" PARTUUID="f997df46-01"
/dev/sda5: UUID="1baegR-eexM-qg6B-BZqx-RMYA-oJXu-dh2Sfz" TYPE="LVM2_member" PARTUUID="f997df46-05"
/dev/mapper/debian10--vg-root: UUID="753d997e-0e32-4a65-b7f0-a755f6b11715" TYPE="ext4"
/dev/sdb1: UUID="415236b0-68bd-4f27-8eaf-5e8ab49d98d7" TYPE="xfs" PARTUUID="00e06621-70ec-c74b-b119-6bb5014527b0"
root@debian10:~#
```

Метку будем использовать для того, чтобы монтировать диск автоматически при загрузке системы. Для этого редактируем файл `/etc/fstab`. Добавляем в самый конец новую строку, чтобы получилось примерно так.

```
/dev/mapper/debian10--vg-root / ext4 errors=remount-ro 0 1
UUID=88c4c0aa-be17-4fd9-b1b7-5c8be142db77 /boot ext2 defaults 0 2
UUID=415236b0-68bd-4f27-8eaf-5e8ab49d98d7 /mnt/backup xfs defaults 0 1
```

Обязательно следите за тем, чтобы в конце файла `fstab` был переход на новую пустую строку. Это важно. Если не сделать переход на пустую строку, система не загрузится.

Очень внимательно редактируйте `fstab`. Ошибка с этим файлом может привести к тому, что система не будет грузиться. Сам сталкивался с этим неоднократно. Я всегда убеждаюсь, что корректно отредактировал `fstab` перезагрузкой системы. У меня были ситуации, когда файл правился с ошибкой, а потом система не перезагружалась месяцами. Через пол года сделал ребут и система не загрузилась. Это был гипервизор с кучей виртуалок. Было не по себе от такого сюрприза. Оказалось, что была ошибка в `fstab`, которую оперативно исправил, благо был доступ к консоли. Внимательно за этим следите.

Поясню еще, почему использовали метку диска, а не название диска в системе — `/dev/sdb1`. Раньше я всегда так и делал. Ну как раньше — лет 7-10 назад. Потом пошли какие-то изменения и стали возникать ситуации, что после добавления новых дисков в систему, менялись системные названия дисков. Когда сталкиваешься с этим впервые — впадаешь в ступор. Вроде только добавил диск в систему, а у тебя все сломалось. То, что было `/dev/sdb` стало `/dev/sdc` со всеми вытекающими последствиями. Выход из этой ситуации — использовать метки разделов, а не названия. Метки не меняются.

## Работа в debian с lvm

LVM тема обширная и раскрыть ее у меня задача не стоит. В сети все это есть, я сам постоянно пользуюсь поиском. Приведу только несколько команд из своей шпаргалки, которыми я регулярно пользуюсь для создания, подключения и изменения lvm дисков. Команды актуальны для любых дистрибутивов, где есть lvm, не только в Debian.

Допустим, вы подключили 2 новых диска или raid массива к серверу и хотите их объединить в единое адресное пространство. Я расскажу, как это сделать. Только сразу обращаю внимание, что подключать одиночные диски так не следует, если там будут храниться важные данные. Выход из строя любого из дисков объединенного раздела приведет к потере всех данных. Это в общем случае. Возможно их можно будет как-то вытащить, но это уже не тривиальная задача.

В системе у меня один диск /dev/sda, я добавил еще 2 — sdb и sdc.

```
# lsblk -a
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0                  2:0    1    4K  0 disk
sda                   8:0    0   20G  0 disk
├─sda1                 8:1    0   243M  0 part /boot
├─sda2                 8:2    0     1K  0 part
└─sda5                 8:5    0  19.8G  0 part
   └─debian10--vg-root 254:0    0  19.8G  0 lvm  /
sdb                   8:16    0   10G  0 disk
sdc                   8:32    0   10G  0 disk
```

Инициализируем диски в качестве физического тома lvm.

```
# pvcreate /dev/sdb /dev/sdc
Physical volume "/dev/sdb" successfully created.
Physical volume "/dev/sdc" successfully created.
```

Теперь создадим группу томов, в которую будут входить оба диска.

```
# vgcreate vgbackup /dev/sdb /dev/sdc
Volume group "vgbackup" successfully created
```

В данном случае `vgbackup` — название созданной группы. Теперь в этой группе томов мы можем создавать разделы. Они в чем-то похожи на разделы обычных дисков. Мы можем как создать один раздел на всю группу томов, так и нарезать эту группу на несколько разделов. Создадим один раздел на всем пространстве группы томов. Фактически, этот раздел будет занимать оба жестких диска, которые мы добавили.

```
# lvcreate -l100%FREE vgbackup -n lv_full
```

`lv_full` название логического раздела. Теперь с ним можно работать, как с обычным разделом. Создавать файловую систему и монтировать к серверу. Сделаем это.

```
# mkfs -t ext4 /dev/vgbackup/lv_full
# mkdir /mnt/backup
# mount /dev/vgbackup/lv_full /mnt/backup
```

Проверяем, что получилось.

```
# df -h | grep /mnt/backup
/dev/mapper/vgbackup-lv_full    20G   45M   19G   1% /mnt/backup
```

Мы подключили `lv` раздел, который расположен на двух жестких дисках. Повторю еще раз — обычные жесткие диски так не собирайте, используйте только `raid` тома для этого.

Теперь для примера давайте удалим этот раздел и создадим 2 новых, один на 14 Гб, другой на 5 Гб и так же их подключим к системе. Для начала удаляем раздел `lv_full`, предварительно отмонтировав его.

```
# umount /mnt/backup
# lvremove /dev/vgbackup/lv_full
```



Проверяем, что раздела нет.

```
# lvs
```



```
root@debian10:/dev/vgbackup# umount /mnt/backup
root@debian10:/dev/vgbackup# df -h
Filesystem                Size      Used Avail Use% Mounted on
udev                      447M          0   447M   0% /dev
tmpfs                      93M       2.8M    90M   3% /run
/dev/mapper/debian10--vg-root 20G       1.1G    18G   6% /
tmpfs                      462M          0   462M   0% /dev/shm
tmpfs                      5.0M          0    5.0M   0% /run/lock
tmpfs                      462M          0   462M   0% /sys/fs/cgroup
/dev/sdal                  236M       48M   176M  22% /boot
tmpfs                      93M          0    93M   0% /run/user/0
root@debian10:/dev/vgbackup# lvremove /dev/vgbackup/lv_full
File descriptor 7 (pipe:[16481]) leaked on lvremove invocation. Parent PID 807: bash
Do you really want to remove active logical volume vgbackup/lv_full? [y/n]: y
Logical volume "lv_full" successfully removed
root@debian10:/dev/vgbackup# lvs
File descriptor 7 (pipe:[16481]) leaked on lvs invocation. Parent PID 807: bash
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
root   debian10-vg -wi-ao---- <19.76g
root@debian10:/dev/vgbackup#
```

Остался только один — системный. Создаем 2 новых раздела:

```
# lvcreate -L14G vgbackup -n lv01
# lvcreate -L4G vgbackup -n lv02
# lvs
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
root   debian10-vg -wi-ao---- <19.76g
lv01   vgbackup -wi-a----- 14.00g
lv02   vgbackup -wi-a-----  4.00g
```

Дальше так же создаем файловые системы и монтируем новые разделы к серверу. Надеюсь, на конкретных примерах я сумел показать удобство и особенность работы с lvm томами и разделами. Дальше мы продолжим эту тему. При автоматическом монтировании через fstab томов lvm можно использовать их имена вида `/dev/mapper/vgbackup-lv_full`, а не метки. Эти имена не меняются.

Вот наглядный пример, где можно использовать lvm тома размазанные на несколько дисков. Есть небольшой файловый сервер с 4-мя sata дисками по 4 tb. Нужно было сделать максимально объемное файловое хранилище. Были собраны 2 mdadm raid1. Немного объема ушло на служебные разделы, а потом все, что осталось объединили в единый lvm том и получили шару объемом 6.4 Tb.



```
[root@xs-design log]# pvs
File descriptor 7 (pipe:[19338208]) leaked on pvs invocation. Parent PID 52453: bash
  PV          VG      Fmt  Attr PSize  PFree
  /dev/md1    vg00  lvm2 a--  <3.62t  0
  /dev/md2    vg00  lvm2 a--  <2.73t  0
[root@xs-design log]# lvs
File descriptor 7 (pipe:[19338208]) leaked on lvs invocation. Parent PID 52453: bash
  LV          VG      Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  epon        vg00  -wi-ao---- <6.35t
[root@xs-design log]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/md127                 20G       5.6G   15G  28% /
devtmpfs                   1.9G         0  1.9G   0% /dev
tmpfs                      1.9G         0  1.9G   0% /dev/shm
tmpfs                      1.9G      173M   1.7G  10% /run
tmpfs                      1.9G         0  1.9G   0% /sys/fs/cgroup
/dev/md126                 1014M     184M   831M  19% /boot
/dev/mapper/vg00-epon      6.4T      3.2T   3.2T  51% /mnt/4tdisk
tmpfs                      375M         0  375M   0% /run/user/0
[root@xs-design log]#
```

Сразу могу сказать, что производительность такой штуки не очень высокая, но зато есть защита от отказа одного из дисков, плюс файловое хранилище нужного объема. Можно было сразу все собрать в `raid10`, но я сейчас уже не помню, почему от этого отказались. Были какие-то объективные причины, а привел эту ситуацию я просто для примера. Таким образом можно объединять различные разнородные массивы для увеличения суммарного объема одного раздела.

## Расширение диска

Теперь представим ситуацию, что у вас используется какой-то `lvm` раздел и вы хотите его увеличить. В общем случае я не рекомендую это делать без особой нужды. Увеличивать можно даже системный диск с `/`, но на практике я получал неожиданные проблемы от такого расширения. Пример такой проблемы — `Booting from Hard Disk error, Entering rescue mode`. В общем случае все должно расширяться без проблем, но когда я разобрался с ошибкой, я

находил в интернете информацию о том, что люди сталкивались с тем же самым именно после расширения системного lvm раздела.

Если раздел не системный, то проблем быть не должно. Последовательность действий следующая при расширении lvm раздела:

1. Добавляем новый диск в систему.
2. Подключаем диск к группе томов, на которой находится раздел, который будем увеличивать.
3. Расширяем lvm раздел за счет свободного места, которое образовалось в группе томов за счет добавления нового диска.

Теперь по пунктам сделаем все это. У нас имеется группа томов из 2-х дисков — sdb и sdc. На этой группе размещен один раздел, который занимает все свободное пространство. Мы его расширим за счет нового диска.

Смотрим, что у нас есть.





```
root@debian10:~# pvs
PV          VG          Fmt Attr PSize  PFree
/dev/sda5   debian10-vg  lvm2 a--  <19.76g  0
/dev/sdb    vgbackup     lvm2 a--  <10.00g  0
/dev/sdc    vgbackup     lvm2 a--  <10.00g  0
root@debian10:~# lvs
LV          VG          Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
root       debian10-vg -wi-ao---- <19.76g
lv_full    vgbackup    -wi-ao---- 19.99g
root@debian10:~# df -h | grep lv_full
/dev/mapper/vgbackup-lv_full 20G  45M  19G  1% /mnt/backup
root@debian10:~#
```

serveradmin.ru

Увеличим раздел `lv_full` до 30 Гб за счет добавления в группу томов нового диска на 10 Гб. Имя этого диска — `sdd`. Добавим его в существующую группу ТОМОВ.

```
# vgextend vgbackup /dev/sdd
Physical volume "/dev/sdd" successfully created.
Volume group "vgbackup" successfully extended
```

Смотрим информацию по томам.

```
# pvs
PV          VG          Fmt Attr PSize  PFree
/dev/sda5   debian10-vg  lvm2 a--  <19.76g  0
/dev/sdb    vgbackup     lvm2 a--  <10.00g  0
/dev/sdc    vgbackup     lvm2 a--  <10.00g  0
/dev/sdd    vgbackup     lvm2 a--  <10.00g  <10.00g
```

Новый диск добавлен в группу и он пока пуст. Расширяем существующий раздел на 100% свободного места группы томов.

```
# lvextend -r -l +100%FREE /dev/vgbackup/lv_full
```



```
root@debian10:~# lvextend -r -l +100%FREE /dev/vgbackup/lv_full
Size of logical volume vgbackup/lv_full changed from 19.99 GiB (5118 extents) to <29.99 GiB (7677 extents).
Logical volume vgbackup/lv_full successfully resized.
resize2fs 1.44.5 (15-Dec-2018)
Filesystem at /dev/mapper/vgbackup-lv_full is mounted on /mnt/backup; on-line resizing required
old_desc_blocks = 3, new_desc_blocks = 4
The filesystem on /dev/mapper/vgbackup-lv_full is now 7861248 (4k) blocks long.

root@debian10:~# df -h | grep lv_full
/dev/mapper/vgbackup-lv_full    30G   44M    28G    1% /mnt/backup
root@debian10:~# pvs
PV          VG          Fmt Attr PSize  PFree
 /dev/sda5  debian10-vg lvm2 a--  <19.76g  0
 /dev/sdb   vgbackup    lvm2 a--  <10.00g  0
 /dev/sdc   vgbackup    lvm2 a--  <10.00g  0
 /dev/sdd   vgbackup    lvm2 a--  <10.00g  0
root@debian10:~# lvs
LV          VG          Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
root       debian10-vg -wi-ao---- <19.76g
lv_full    vgbackup    -wi-ao---- <29.99g
```

Раздел увеличен. Я его не отключал в процессе расширения. Все сделано, как говорится, на горячую. С lvm это очень просто и я наглядно рассказал, как это сделать. А теперь давайте расширим обычный раздел диска, не lvm. Это тоже реально и не сильно сложнее.

Допустим, у нас есть диск /dev/sdb размером 10 Гб, на нем один раздел sdb1, который занимает все свободное пространство диска. Это диск виртуальной машины, который мы можем увеличить через управление дисками гипервизора. Я расширил диск до 20 Гб. Смотрим, что получилось.

```
# fdisk -l | grep /dev/sdb
GPT PMBR size mismatch (20971519 != 41943039) will be corrected by write.
The backup GPT table is not on the end of the device. This problem will be corrected by write.
Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
```

```
/dev/sdb1 2048 20971486 20969439 10G Linux filesystem  
Partition 2 does not start on physical sector boundary.
```

У нас объем диска 20 Гб, на нем только один раздел на 10 Гб. Нам его надо расширить до 20-ти Гб. Диск нужно отмонтировать, прежде чем продолжать.

```
# umount /dev/sdb1
```

Открываем диск в fdisk и выполняем там следующую последовательность действий:

1. Удаляем существующий раздел sdb1 на 10G.
2. Вместо него создаем новый sdb1 на 20G.
3. Записываем изменения.

```
# fdisk /dev/sdb
```



```
root@debian10:~# fdisk /dev/sdb
Welcome to fdisk (util-linux 2.33.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

GPT FMBR size mismatch (20971519 != 41943039) will be corrected by write.
The backup GPT table is not on the end of the device. This problem will be corrected by write.

Command (m for help): p

Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: Virtual Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 709C944D-EA5F-834A-B4B1-77FCD63E5A6F

Device      Start      End  Sectors  Size  Type
/dev/sdb1   2048 20971486 20969439  10G  Linux filesystem

Command (m for help): d
Selected partition 1
Partition 1 has been deleted.

Command (m for help): n
Partition number (1-128, default 1):
First sector (2048-41943006, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-41943006, default 41943006):

Created a new partition 1 of type 'Linux filesystem' and of size 20 GiB.
Partition #1 contains a ext4 signature.

Do you want to remove the signature? [Y]es/[N]o: n
Command (m for help): p

Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: Virtual Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 709C944D-EA5F-834A-B4B1-77FCD63E5A6F

Device      Start      End  Sectors  Size  Type
/dev/sdb1   2048 41943006 41940959  20G  Linux filesystem

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Не удаляем сигнатуру

serveradmin.ru

Проверяем, что получилось:

```
# fdisk -l | grep /dev/sdb
Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
/dev/sdb1  2048 41943006 41940959  20G Linux filesystem
```

Монтируем раздел обратно в систему и проверяем размер.

```
# mount /dev/sdb1 /mnt/backup
# df -h | grep sdb1
/dev/sdb1          9.8G   37M  9.3G   1% /mnt/backup
```

Размер файловой системы на разделе не изменился. Расширяем ее отдельной командой.

```
# resize2fs /dev/sdb1
resize2fs 1.44.5 (15-Dec-2018)
Filesystem at /dev/sdb1 is mounted on /mnt/backup; on-line resizing required
old_desc_blocks = 2, new_desc_blocks = 3
The filesystem on /dev/sdb1 is now 5242619 (4k) blocks long.
```

Проверяем, что получилось.

```
# df -h | grep sdb1
/dev/sdb1          20G   44M   19G   1% /mnt/backup
```

Раздел ext4 расширился до желаемых 20 Гб. Я не пробовал расширять системный раздел ext4 без размонтирования, не было необходимости. Но если у вас системный раздел на xfs, то вот пример того, как его можно расширить без отмонтирования и без остановки сервера — Расширение (увеличение) xfs корневого раздела / без остановки.



## Проверка диска на ошибки и bad blocks

С выходом файловых систем ext4 и xfs я практически забыл, что такое проверка диска на ошибки. Сейчас прикинул и ни разу не вспомнил, чтобы у меня были проблемы с файловой системой. Раньше с ext3 или ufs на freebsd проверка диска на ошибки было обычным делом после аварийного выключения или еще каких бед с сервером. Ext4 и xfs в этом плане очень надежны.

В основном ошибки с диском вызваны проблемами с железом. Как посмотреть параметры smart я уже показал выше. Но если у вас все же появились какие-то проблемы с файловой системой, то решить их можно с помощью **fsck** (File System Check). Обычно она входит в базовый состав системы. Запустить проверку можно либо указав непосредственно раздел или диск, либо точку монтирования. Раздел при этом должен быть отмонтирован.

```
# umount /dev/sdb1
# fsck /dev/sdb1
fsck from util-linux 2.33.1
e2fsck 1.44.5 (15-Dec-2018)
/dev/sdb1: clean, 11/1310720 files, 109927/5242619 blocks
```

Проверка завершена, ошибок у меня не обнаружено. Так же у fsck есть необычная опция, которая не указана в документации или man. Запустив fsck с ключем -c можно проверить диск на наличие бэд блоков.

```
# fsck -c /dev/sdb1
```



```
root@debian10:/var/log# fsck -c /dev/sdb1
fsck from util-linux 2.33.1
e2fsck 1.44.5 (15-Dec-2018)
Checking for bad blocks (read-only test): done
/dev/sdb1: Updating bad block inode.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information

/dev/sdb1: ***** FILE SYSTEM WAS MODIFIED *****
/dev/sdb1: 11/1310720 files (0.0% non-contiguous), 109927/5242619 blocks
root@debian10:/var/log#
```

Насколько я понимаю, проверка выполняет посекторное чтение и просто сообщает о том, что найден бэд блок. Теоретически, можно собрать все эти блоки в отдельный файл и затем передать их утилите e2fsck, которая сможет запомнить эти бэды и исключить из использования.

```
# fsck -c /dev/sdb1 > badblocks.txt
# e2fsck -l badblocks.txt /dev/sdb1
```

На практике я не проверял как это работает и имеет ли вообще смысл в таких действиях. Если с диском замечены хоть малейшие проблемы, я его сразу меняю.

## Протестировать скорость диска

Проверка реальной скорости диска задача не простая. Во-первых, в операционной системе есть кэш. Если используется рейд контроллер, то в нем тоже есть свой кэш. Важно мерить скорость так, чтобы не попадать в эти кэши. К тому же проще всего измерить линейную скорость чтения и записи, но в реальном профиле нагрузки сервера линейной записи и чтения практически не бывает. Так что я предлагаю пару простых способов измерить скорость записи и чтения на глазок, просто чтобы прикинуть или сравнить разные диски.

Для теста записи можно воспользоваться утилитой **dd**, записав пустой файл. Размер файла выбирайте больше объема оперативной памяти. За это отвечает параметр **count**, который задает количество блоков **bs** в 1 мб, которые мы запишем.

```
# sync; dd if=/dev/zero of=tempfile bs=1M count=12000; sync
12000+0 records in
12000+0 records out
12582912000 bytes (13 GB, 12 GiB) copied, 14.4436 s, 871 MB/s
```



```
root@debian10:~# sync; dd if=/dev/zero of=tempfile bs=1M count=12000; sync
12000+0 records in
12000+0 records out
12582912000 bytes (13 GB, 12 GiB) copied, 14.4436 s, 871 MB/s
root@debian10:~#
```

Это обычный ssd диск Samsung 860 EVO. Виртуальная машина работает на нем. А вот результат на рейд контроллере с отложенной записью, где массив raid10 собран из 4-х sata hdd.

```
# sync; dd if=/dev/zero of=tempfile bs=1M count=2000; sync
2000+0 records in
2000+0 records out
2097152000 bytes (2.1 GB) copied, 1.83823 s, 1.1 GB/s
```

Тут явно запись полностью попадает в кэш контроллера, поэтому такая нереальная скорость для обычных hdd дисков. Привожу это для примера, чтобы вы понимали, что то, что вы видите на тестах скорости это не всегда скорость самих дисков. Вот обычный софтовый raid1 на двух hdd дисках.

```
# sync; dd if=/dev/zero of=tempfile bs=1M count=6000; sync
6000+0 records in
6000+0 records out
6291456000 bytes (6.3 GB) copied, 37.7701 s, 167 MB/s
```

Скорость чтения диска можно измерить, к примеру, с помощью программы **hdparm**. Ставится из стандартных репозиториев.

```
# apt install hdparm
# hdparm -t /dev/vda1
/dev/vda1:
Timing buffered disk reads: 742 MB in 3.00 seconds = 247.13 MB/sec
```

Привожу в пример эту программу, так как у нее явно указано, что ключ `t` позволяет производить чтение непосредственно с диска, минуя кэш системы. Хотя способов это сделать есть множество, но этот, как мне кажется, самый простой.

## Проверить нагрузку на диск

Теперь немного поговорим о том, как измерить или посмотреть нагрузку дисков. Иногда видно, что сервер явно очень сильно тормозит. При этом смотришь загрузку CPU и доступную память — всего в избытке, но приложение еле шевелится. Конечно, с нормальным мониторингом вы быстро поймете, что у вас проблемы с диском, но если мониторинга нет, то можно в консоли быстро оценить обстановку с помощью некоторых утилит.

Я предлагаю использовать небольшой пакет утилит **sysstat**. Ставим его.

```
# apt install sysstat
```

Дальше запускаем **iostat** с параметрами.

```
# iostat -xk -t 2
```





```
Device:      rrqm/s  wrqm/s     r/s     w/s    kB/s    kB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
vda          0.00    8.54     0.00   19.10    0.00   295.48   30.95     0.00    3.97   0.00    3.97   0.08   0.15

09/06/2019 10:16:09 PM
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.51    0.00    0.51    7.14    0.00   91.84

Device:      rrqm/s  wrqm/s     r/s     w/s    kB/s    kB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
vda          0.00    8.67     0.00   19.39    0.00   300.00   30.95     0.00    3.87   0.00    3.87   0.11   0.20

09/06/2019 10:16:11 PM
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00    1.01    8.04    0.00   90.45

Device:      rrqm/s  wrqm/s     r/s     w/s    kB/s    kB/s avgrq-sz avgqu-sz   await r_await w_await  svctm  %util
vda          0.00   10.05     0.00   25.63    0.00   383.92   29.96     0.01    3.27   0.00    3.27   0.12   0.30
```

Нас в первую очередь будет интересовать последний столбец *%util*. Чем больше он стремится к 100%, тем выше нагрузка на диск. Если она очень высокая, вы захотите узнать, какой процесс больше всего нагружает диск. Это можно узнать с помощью **pidstat**.

```
# pidstat -d 1
```



```
10:18:28 PM UID PID kB_rd/s kB_wr/s kB_ccwr/s Command
10:18:29 PM 27 13652 0.00 631.37 0.00 mysql
10:18:29 PM UID PID kB_rd/s kB_wr/s kB_ccwr/s Command
10:18:30 PM 27 13652 0.00 305.15 0.00 mysql
10:18:30 PM UID PID kB_rd/s kB_wr/s kB_ccwr/s Command
10:18:31 PM 27 13652 0.00 268.00 0.00 mysql
10:18:31 PM UID PID kB_rd/s kB_wr/s kB_ccwr/s Command
10:18:32 PM 27 13652 0.00 232.65 0.00 mysql
10:18:32 PM UID PID kB_rd/s kB_wr/s kB_ccwr/s Command
10:18:33 PM 27 13652 0.00 230.30 0.00 mysql
```

Видим, что основную нагрузку на диск генерирует процесс `mysqld`. Можем отдельно последить за ним. Для этого надо узнать его `pid` и запустить `pidstat` для конкретного процесса.

```
# pidstat -p `pgrep mysqld` -d 1
```



Time	PM	UID	PID	kB_rd/s	kB_wr/s	kB_ccwr/s	Command
10:21:29	PM	27	13652	0.00	260.00	0.00	mysqld
10:21:30	PM	27	13652	0.00	297.96	0.00	mysqld
10:21:31	PM	27	13652	0.00	456.57	0.00	mysqld
10:21:32	PM	27	13652	0.00	173.74	0.00	mysqld
10:21:33	PM	27	13652	0.00	197.98	0.00	mysqld
10:21:34	PM	27	13652	0.00	428.57	0.00	mysqld
10:21:35	PM	27	13652	0.00	229.70	0.00	mysqld
10:21:36	PM	27	13652	0.00	232.65	0.00	mysqld
10:21:37	PM	27	13652	0.00	230.30	0.00	mysqld
10:21:38	PM	27	13652	0.00	200.00	0.00	mysqld
10:21:39	PM	27	13652	0.00	200.00	0.00	mysqld

Так мы можем мониторить дисковую активность конкретного процесса.

Так же для наглядного мониторинга загрузки диска мне нравится использовать программу **dstat**.

```
# apt install dstat  
# dstat --top-bio
```

process	read	write
systemd	15k	212k
mysqld	0	64k
mysqld	0	8192B
openvpn	0	4096B
filebeat	0	4096B
openvpn	0	4096B
openvpn	0	4096B
mysqld	0	456k
mysqld	0	1096k
mysqld	0	8192B

Чтение (read) is indicated by a red arrow pointing to the 64k value for the first mysqld process.

Запись (write) is indicated by a red arrow pointing to the 456k value for the mysqld process.

Она в реальном времени показывает дисковую нагрузку конкретных приложений. Есть еще похожая команда.

```
# dstat --top-io
```

Если я правильно понял описание, первая показывает скорость доступа к реальным устройствам, а вторая в том числе кэши и всякие виртуальные файловые системы. Возможно я ошибся, вот описание из документации:

—top-bio

show most expensive block I/O process

—top-io

## | *show most expensive I/O process*

Программа `dstat` показывает не только загрузку дисков. Ее можно использовать для комплексного наблюдения за системой. Например вот так.

```
# dstat -tldnpms 10
```

При этом будет выводиться:

- текущее время - t
- средняя загрузка системы - l
- использования дисков - d
- загрузка сетевых устройств - n
- активность процессов - p
- использование памяти - m
- использование подкачки - s
- с интервалов в 10 секунд







```
-----system-----  ---load-avg---  -dsk/total-  -net/total-  ----procs----  -----memory-usage-----  ----swap-----
   time      | 1m  5m 15m | read writ| recv  send| run blk new| used  buff  cach  free| used  free
06-09 22:40:48| 0.01 0.02 0.05| 36k  238k|  0      0|  0.0  0 0.4| 1419M 24.0k 1814M 556M|  0      0
06-09 22:40:58| 0.01 0.02 0.05|  0    26k| 5596B  52k|  0.6  0.1 0.8| 1439M 24.0k 1814M 537M|  0      0
06-09 22:41:08| 0.01 0.02 0.05|  0    86k|  15k  132k|  0      0 0.2| 1422M 24.0k 1814M 553M|  0      0
06-09 22:41:18| 0.01 0.02 0.05|  0    44k| 8144B  97k|  0      0 0| 1416M 24.0k 1814M 560M|  0      0
06-09 22:41:28| 0.01 0.02 0.05|  0 4915B|  12k  22k|  0      0 1.0| 1416M 24.0k 1814M 560M|  0      0
06-09 22:41:38| 0.01 0.02 0.05|  0    81k| 4830B  23k|  0      0 0.2| 1412M 24.0k 1814M 564M|  0      0
06-09 22:41:48|  0 0.01 0.05|  0    71k|  20k  60k|  0.2    0 0| 1427M 24.0k 1814M 548M|  0      0
06-09 22:41:55|  0 0.01 0.05|  0    20k| 7147B 217k|  0      0 0.9| 1411M 24.0k 1814M 565M|  0      0 ^C
```

На этом по анализу нагрузки на диск у меня все. Если у вас есть еще какие-то интересные утилиты для этих задач, прошу поделиться в комментариях. Загрузка дисков одна из самых востребованных метрик в серверах, так как очень сложно правильно ее оценить, в отличие от загрузки сри и memory.

## Создание диска из оперативной памяти (озу)

Расскажу об еще одной интересной возможности при работе с дисками — создание диска из оперативной памяти. Зачем это может пригодиться на практике, я не знаю. Сам использовал только из любопытства для тестов. В проде лично я не вижу применения. Linux обычно и так использует всю доступную оперативную память и кэширует файловые операции. Так что большой необходимости в дисках в оперативной памяти я не вижу. Но может вы найдете реальное применение.

Создаем диск из оперативной памяти размером в 5 Гб и монтируем его в `/mnt/tmpfs`.

```
# mkdir /mnt/tmpfs/
# mount -t tmpfs -o size=5000M tmpfs /mnt/tmpfs/
```

Смотрим, что получилось.

```
# df -h
```



```
root@debian10:~# mkdir /mnt/tmpfs/  
root@debian10:~# mount -t tmpfs -o size=5000M tmpfs /mnt/tmpfs/  
root@debian10:~# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	3.9G	0	3.9G	0%	/dev
tmpfs	799M	8.5M	791M	2%	/run
/dev/mapper/debian10--vg-root	20G	1.1G	18G	6%	/
tmpfs	3.9G	0	3.9G	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	3.9G	0	3.9G	0%	/sys/fs/cgroup
/dev/sdal	236M	48M	176M	22%	/boot
tmpfs	799M	0	799M	0%	/run/user/0
tmpfs	4.9G	0	4.9G	0%	/mnt/tmpfs

Я создал диск в оперативной памяти размером 5 Гб. Проверим его скорость записи.

```
# # sync; dd if=/dev/zero of=/mnt/tmpfs/tempfile bs=1M count=3000; sync  
3000+0 records in  
3000+0 records out  
3145728000 bytes (3.1 GB, 2.9 GiB) copied, 1.11855 s, 2.8 GB/s
```

Неплохо :) Что примечательно, оперативная память не резервируется сразу при создании диска. Она расходуется только если на диск действительно записываются какие-то файлы. Причем расходуется память пропорционально количеству данных, которые на диск записаны. Если у вас есть идеи, как можно использовать такие разделы, делитесь в комментариях. Первое, что приходит в голову — использовать диск из оперативной памяти под кэш статических страниц web сайта. Но как я уже говорил, ядро linux и так должно кэшировать в памяти то, к чему идут интенсивные запросы.

На практике я все это не проверял. Есть какой-то прирост к отклику сайта или нет, если его кэш поместить на такой диск. Было бы любопытно проверить, но как-то все времени не хватает. Может кто-то проверит и поделится результатом. Так же такой диск можно использовать под какую-нибудь тестовую базу данных для ускорения обработки запросов. Можно, конечно, заморочиться и настроить нормально кэши самой базы. Но если файлы базы просто поместить на диск в оперативной памяти, то прирост очевидный будет сразу же, без всяких настроек.

Обращаю внимание на важный нюанс. Следите за заполнением диска из памяти. Если вы израсходуете всю доступную память и начнет работать swap, вы получите сильные тормоза в системе.

## Подключение Яндекс Диска

Статью с настройкой дисков завершу описанием подключения [Яндекс.Диска](#). Я лично давно и интенсивно его использую. У меня есть статья по созданию резервной копии сайта на яндекс.диск. Статья хоть и старая, но актуальная. Я продолжаю использовать предложенные там решения.

Яндекс диск можно подключить как системный диск по webdav. Скажу сразу, что работает это так себе, я давно им не пользуюсь в таком виде. Мне больше нравится работать с ним через консольный клиент linux.

Устанавливаем консольный клиент yandex-disk на Debian.

```
# echo "deb http://repo.yandex.ru/yandex-disk/deb/ stable main" | tee -a /etc/apt/sources.list.d/yandex-disk.list > /dev/null
# apt install gnupg
# wget http://repo.yandex.ru/yandex-disk/YANDEX-DISK-KEY.GPG -O- | apt-key add -
# apt update && apt install yandex-disk
```

Дальше запускаете начальную настройку.

```
# yandex-disk setup
```

После этого яндекс диск подключен к системе и готов к работе. Посмотреть его статус можно командой.

```
# yandex-disk status
```



```
root@debian10:/etc/apt/sources.list.d# yandex-disk status
Synchronization core status: index
Path to Yandex.Disk directory: '/root/Yandex.Disk'
  Total: 514.77 GB
  Used: 455.40 GB
  Available: 59.37 GB
  Max file size: 50 GB
  Trash size: 45.10 GB

Last synchronized items:
  None.

root@debian10:/etc/apt/sources.list.d# yandex-disk stop
Daemon stopped.
root@debian10:/etc/apt/sources.list.d#
```

Остановить или запустить Яндекс.Диск можно командами.

```
# yandex-disk stop
# yandex-disk start
```

Файл конфигурации находится по адресу `/root/.config/yandex-disk/config.cfg`. Туда, к примеру, можно добавить список папок исключений, которые не нужно синхронизировать.

```
exclude-dirs="dir1,exclude/dir2,path/to/another/exclude/dir"
```

Консольный клиент поддерживает символные ссылки. Я много где использовал его. В основном в скриптах по автоматизации бэкапов. К примеру, я останавливал сервис яндекс диска, готовил бэкапы к отправке. Упаковывал их архиватором с разбивкой архивов по размеру. Потом создавал символные ссылки в папке яндекс диска и запускал синхронизацию. Когда она заканчивалась, удалял локальные файлы и останавливал синхронизацию.

Яндекс диск сильно тормозит и падает, если у вас много мелких файлов. Мне доводилось хранить в нем бэкапы с сотнями тысяч файлов. Передать их в облако напрямую было невозможно. Я паковал их в архивы по 2-10 Гб и заливал через консольный клиент. Сразу могу сказать, что это решение в пользу бедных. Этот облачный диск хорош для домашних нужд пользователей и хранения семейных фоток и видео. Когда у вас большие потоки данных, которые нужно постоянно обновлять, работа с яндекс диском становится сложной.

Во-первых, трудно мониторить такие бэкапы. Во-вторых, тяжело убедиться в том, что то, что ты залил в облако, потом нормально скачается и распакуется из бэкапа. Как запасной вариант для архивов, куда они будут складываться раз в неделю или месяц, подойдет. Но как основное резервное хранилище точно нет. Какие только костыли я не придумывал для Яндекс.Диска в процессе промышленной эксплуатации. В итоге все равно почти везде отказался. Да, это очень дешево, но одновременно и очень ненадежно. Он иногда падает. Это хорошо, что упал, можно отследить и поднять. Так же он может зависнуть и просто ничего не синхронизировать, при этом служба будет работать. Все это я наблюдал, когда пытался синхронизировать сотни гигабайт данных. Иногда у меня это получалось :)

## Заключение

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении. Расскажи, как сделать правильно!

На этом по настройке дисков в Debian у меня все. Постарался рассмотреть самые актуальные и полезные моменты. Использовал свои шпаргалки с командами для этого. Теперь можно лазить не в шпаргалки, а сюда :) Надеюсь было полезно кому-то еще. Все замечания, пожелания, исправления и предложения по теме статьи жду в комментариях.

Напоминаю, что данная статья является частью единого цикла статей про сервер Debian.

## Онлайн курс Основы сетевых технологий

Теоретический курс с самыми **базовыми знаниями по сетям**. Курс подходит и начинающим, и людям с опытом. Практикующим системным администраторам курс поможет упорядочить знания и восполнить пробелы. А те, кто только входит в профессию, получат на курсе базовые знания и навыки, без воды и избыточной теории. После обучения вы сможете ответить на вопросы:



- На каком уровне модели OSI могут работать коммутаторы;
- Как лучше организовать работу сети организации с множеством отделов;
- Для чего и как использовать технологию VLAN;
- Для чего сервера стоит выносить в DMZ;
- Как организовать объединение филиалов и удаленный доступ сотрудников по vpn;
- и многое другое.

Уже знаете ответы на вопросы выше? Или сомневаетесь? Попробуйте пройти тест по основам сетевых технологий. Всего 53 вопроса, в один цикл теста входит 10 вопросов в случайном порядке. Поэтому тест можно проходить несколько раз без потери интереса. Бесплатно и без регистрации. Все подробности на странице .

Помогла статья? Есть возможность отблагодарить автора