

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №14

дисциплина: Основы администрирования операционных систем

Студент: Ко Антон Геннадьевич

Студ. билет № 1132221551

Группа: НПИбд-02-23

МОСКВА

2024 г.

Цель работы:

Целью данной работы является получение навыков создания разделов на диске и файловых систем, а также навыков монтирования файловых систем.

Выполнение работы:

Создание виртуальных носителей:

Добавим к нашей виртуальной машине два диска размером 512 МБ. Для добавления диска в VirtualBox для нашей виртуальной машины нажмём на меню «**Настроить**», выберем «**Носители**», затем на контроллере SATA нажмём «**Добавить жёсткий диск**» (Рис. 1.1):

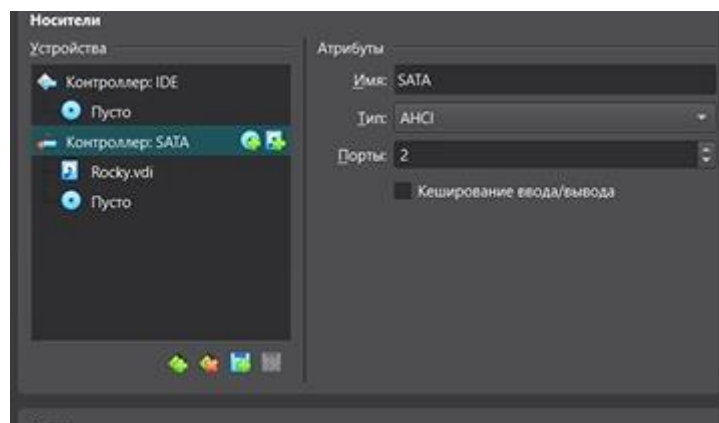


Рис. 1.1. Начало процесса добавления дисков.

В открывшемся окне нажмём «**Создать образ диска**», в следующем окне выберем «**VDI**» и нажмём «**Далее**» (Рис. 1.2), укажем месторасположение диска и его название (**disk1.vdi** или **disk2.vdi**), а также его размер — **512 МБ**, после чего нажимаем «**Создать**» (Рис. 1.3) Для добавления второго диска размером 512 МБ к контроллеру SATA повторим все указанные выше действия.

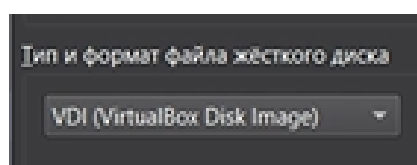


Рис. 1.2. Указание типа виртуального жёсткого диска.

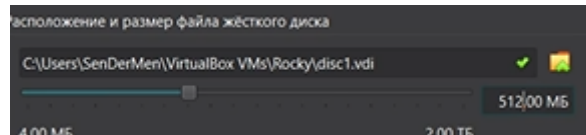


Рис. 1.3. Указание имени и размера файла.

В окне выбора жёсткого диска встанем на обозначение созданного диска и нажмём **«Выбрать»**. (Рис. 1.4):

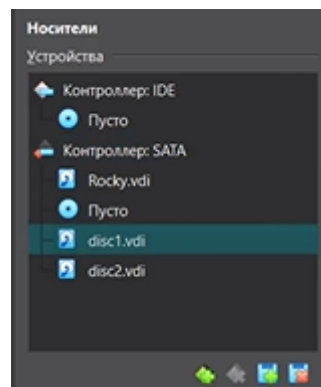


Рис. 1.4. Выбор созданных дисков.

Создание разделов MBR с помощью fdisk:

Запустим нашу виртуальную машину с добавленными дополнительными дисками disk1 и disk2. В командной строке с полномочиями администратора с помощью fdisk посмотрим перечень разделов на всех имеющихся в системе устройствах жёстких дисков:

su –

fdisk --list

В списке отразилась информация о добавленных дисках размером 512 MiB, в частности название разделов: **/dev/sdb** и **/dev/sdc**. (Рис. 2.1):

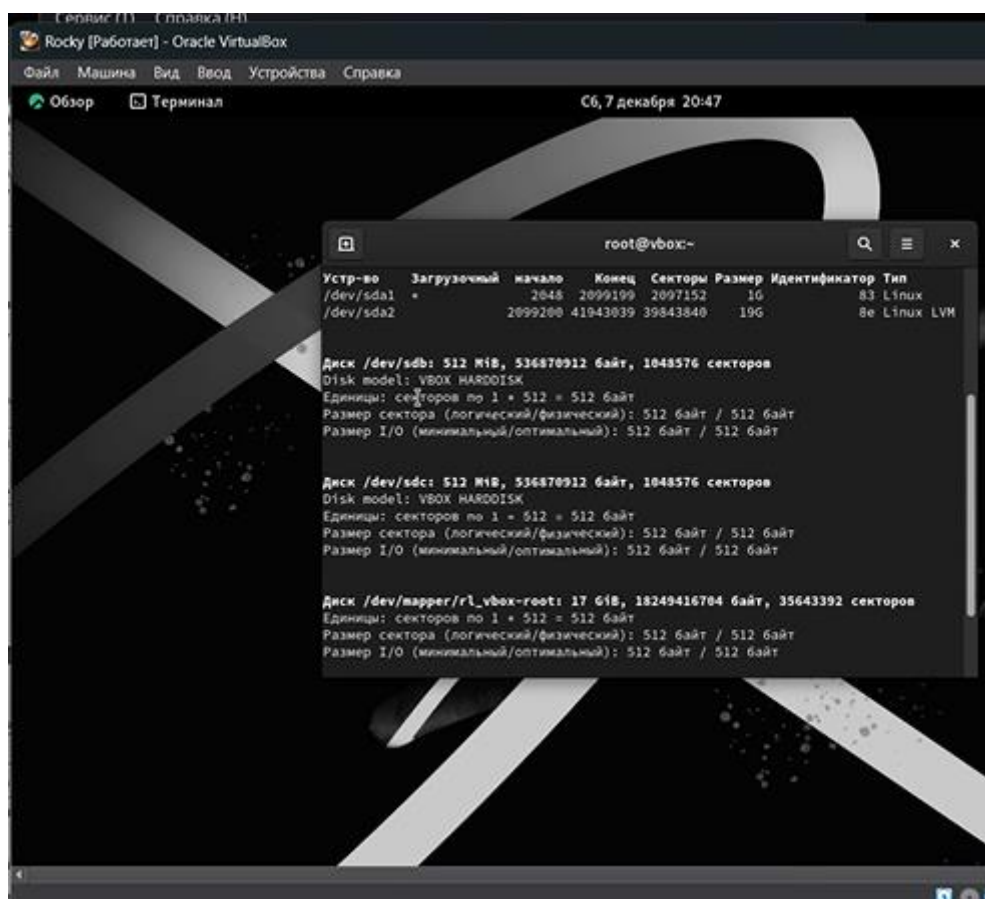


Рис. 2.1. Получение полномочий администратора, просмотр перечня разделов на всех имеющихся в системе устройствах жёстких дисков.

Предположим, что необходимо сделать разметку диска `/dev/sdb` с помощью утилиты `fdisk`: **`fdisk /dev/sdb`**. Изменения останутся в памяти только до тех пор, пока мы не решим их записать (будем внимательны перед использованием команды записи. Утилита `fdisk` записывает изменения на диск только при вводе команды `w`. Если мы допустили ошибку и хотим выйти, то нажмем `q` для выхода из `fdisk` без записи изменений). Введём **`m`**, чтобы получить справку по командам (Рис. 2.2):

```

[root@vbox ~]# fdisk /dev/sdb
Добро пожаловать в fdisk (util-linux 2.37.4).
Изменения останутся только в памяти до тех пор, пока вы не решите записать их.
Будьте внимательны, используя команду write.

Устройство не содержит стандартной таблицы разделов.
Создана новая метка DOS с идентификатором 0x8ad3a9ba.

Команда (m для справки):

```

Рис. 2.2. Начала процесса по разметке диска, получение справки по командам.

Прежде чем делать что-либо, рекомендуется проверить, сколько дискового пространства у нас есть. Нажмём **p**, чтобы просмотреть текущее распределение пространства диска. Обратим внимание на общее количество секторов и последний сектор, который в настоящее время используется. Если последний раздел не заканчивается в последнем секторе, то у нас есть свободное место для создания нового раздела. Введём **n**, чтобы добавить новый раздел. Выберем **p**, чтобы создать основной раздел. Примем номер раздела, который предлагается и укажем первый сектор на диске, с которого начнётся новый раздел. По умолчанию предлагается первый доступный сектор, нажмём **Enter** для подтверждения выбора. Укажем последний сектор, которым будет завершён раздел. По умолчанию предлагается последний сектор, доступный на диске. Если мы согласимся с предложенным по умолчанию вариантом, то после этого упражнения у нас не останется свободного места на диске для создания дополнительных разделов или логических томов. Поэтому мы должны использовать другой последний сектор. Например, введём **+100M**, чтобы создать раздел на 100 MiB. На этом этапе можно определить тип раздела. По умолчанию используется тип раздела Linux. Если мы хотим, чтобы раздел имел какой-либо другой тип, используйте для изменения **t**. Нажмем **Enter**, чтобы принять тип раздела по умолчанию 83. Нажмите **w**, чтобы записать изменения на диск и выйти из fdisk (Рис. 2.3):

```
Команда (я для справки): p
Диск /dev/sdb: 512 MiB, 536870912 байт, 1048576 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 = 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x8ad3a9ba

Команда (я для справки): n
Тип раздела
  p основной (0 primary, 0 extended, 4 free)
  e расширенный (контейнер для логических разделов)
Выберите (по умолчанию - p): p
Номер раздела (1-4, default 1): 1
Первый сектор (2048-1048575, default 2048): 2048
Last sector, +/-sectors or +/-size[K,M,G,T,P] (2048-1048575, default 1048575): +
100M

Создан новый раздел 1 с типом 'Linux' и размером 100 MiB.

Команда (я для справки):
```

Рис. 2.3. Просмотр текущего распределения пространства диска, добавление нового раздела, создание основного раздела, запись изменения на диск и выход из fdisk.

Таблица разделов находится только в памяти ядра. Сравним вывод команды **fdisk -l /dev/sdb** с выводом команды **cat /proc/partitions**. Запишем изменения в таблицу разделов ядра: **partprobe /dev/sdb** (Рис. 2.4):

```
root@vbox:~  
Disk model: VBOX HARDDISK  
Единицы: секторов по 1 * 512 = 512 байт  
Размер сектора (логический/физический): 512 байт / 512 байт  
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт  
Тип метки диска: dos  
Идентификатор диска: 0x8ad3a9ba  
  
Устр-во    Загрузочный  начало  Конец  Секторы  Размер  Идентификатор  Тип  
/dev/sdb1          2048 206847  204800   100M          83 Linux  
[root@vbox ~]# cat /proc/partitions  
major minor #blocks name  
  
11         0     1048575 sr0  
8          0     20971520 sda  
8          1     1048576 sda1  
8          2     19921920 sda2  
11         1     1048575 sr1  
8          16     524288 sdb  
8          17     102400 sdb1  
8          32     524288 sdc  
253        0     17821696 dm-0  
253        1     2097152 dm-1  
[root@vbox ~]# partprobe /dev/sdb  
[root@vbox ~]#
```

Рис. 2.4. Сравнение выводов команд, запись изменения в таблицу разделов ядра.

Создание логических разделов:

В терминале с полномочиями администратора запустим **fdisk /dev/sdb**. Теперь введём **n**, чтобы добавить новый раздел и **e**, чтобы создать расширенный раздел (если расширенный раздел — четвёртый раздел, который мы записываем в MBR, он также будет последним разделом, который можно добавить в MBR. По этой причине он должен заполнить всю оставшуюся часть жёсткого диска нашего компьютера). Нажмём «**Enter**», чтобы принять первый сектор по умолчанию и снова нажмём «**Enter**», когда fdisk запросит последний сектор (Рис. 3.1).

```
[root@vbox ~]# fdisk /dev/sdb

Добро пожаловать в fdisk (util-linux 2.37.4).
Изменения останутся только в памяти до тех пор, пока вы не решите записать их.
Будьте внимательны, используя команду write.

Команда (m для справки): n
Тип раздела
  p    основной (1 primary, 0 extended, 3 free)
  e    расширенный (контейнер для логических разделов)
Выберите (по умолчанию - p): e
Номер раздела (2-4, default 2):
Первый сектор (206848-1048575, default 206848):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (206848-1048575, default 1048575):

Создан новый раздел 2 с типом 'Extended' и размером 411 MiB.
```

Рис. 3.1. Запуск **fdisk /dev/sdb**, добавление нового раздела, создание расширенного раздела.

Теперь, когда расширенный раздел создан, мы можем создать в нём логический раздел. Из интерфейса **fdisk** снова нажмём **n** (утилита сообщает, что нет свободных первичных разделов и по умолчанию предлагает добавить логический раздел с номером 5). Нажмём **Enter**, чтобы принять выбор первого сектора в качестве сектора по умолчанию. На вопрос о последнем секторе введём **+101M**. После создания логического раздела введём **w**, чтобы записать изменения на диск и выйти из **fdisk** (Рис. 3.2):


```

Команда (m для справки): n
Все пространство для логических разделов задействовано.
Добавление логического раздела 5
Первый сектор (208896-1048575, default 208896):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (208896-1048575, default 1048575):
+101M

Создан новый раздел 5 с типом 'Linux' и размером 101 MiB.

Команда (m для справки): w
Таблица разделов была изменена.
Вызывается ioctl() для перечитывания таблицы разделов.
Синхронизируются диски.

[root@vbox ~]#

```

Рис. 3.2. Создание логического раздела, запись изменения на диск и последующий выход.

Чтобы завершить процедуру и обновить таблицу разделов, введём **partprobe /dev/sdb**. Новый раздел теперь готов к использованию. Просмотрим информацию о добавленных разделах (Рис. 3.3):

cat /proc/partitions

fdisk --list /dev/sdb

```

[root@vbox ~]# partprobe /dev/sdb
[root@vbox ~]# cat /proc/partitions
major minor #blocks name
11        0    1048575 sr0
 8         0   20971520 sda
 8         1    1048576 sda1
 8         2   19921920 sda2
11        1    1048575 sr1
 8        16     524288 sdb
 8        17     102400 sdb1
 8        18         1 sdb2
 8        21     103424 sdb5
 8        32     524288 sdc
253        0   17821696 dm-0
253        1   2097152 dm-1
[root@vbox ~]# fdisk --list /dev/sdb
Диск /dev/sdb: 512 MiB, 536870912 байт, 1048576 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт

```

Рис. 3.3. Завершение процедуры и обновление таблицы разделов, просмотр информации о добавленных разделах.

Создание раздела подкачки:

Запустим fdisk: **fdisk /dev/sdb**. Нажмём **n**, чтобы добавить новый раздел (утилита сообщает, что нет свободных первичных разделов и по умолчанию предлагает добавить логический раздел с номером раздела 6). Нажмём «**Enter**», чтобы принять первый сектор по умолчанию. На вопрос о последнем секторе введём **+100M**. Далее изменим тип раздела. Для этого нажмём «**t**», затем укажем номер партии, для которой хотим изменить тип (в данном случае это номер 6). Затем введём код типа раздела (в данном случае 82 — раздел подкачки). После создания логического раздела введём «**w**», чтобы записать изменения на диск и выйти из fdisk (Рис. 4.1):

```
Команда (m для справки): n
Все пространство для логических разделов задействовано.
Добавление логического раздела 6
Первый сектор (417792-1048575, default 417792):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (417792-1048575, default 1048575):
+100M

Создан новый раздел 6 с типом 'Linux' и размером 100 MiB.

Команда (m для справки): t
Номер раздела (1,2,5,6, default 6): 6
Hex code or alias (type L to list all): 82

Тип раздела 'Linux' изменен на 'Linux swap / Solaris'.

Команда (m для справки): w
Таблица разделов была изменена.
Вызывается ioctl() для перечитывания таблицы разделов.
Синхронизируются диски.
```

Рис. 4.1. Запуск fdisk, добавление нового раздела, изменение типа раздела, запись изменений на диск и выход из fdisk.

Чтобы завершить процедуру и обновить таблицу разделов ядра, введём **partprobe /dev/sdb**. Новый раздел теперь готов к использованию. Просмотрим информацию о добавленных разделах (Рис. 4.2):

cat /proc/partitions

```
[root@vbox ~]# partprobe /dev/sdb
[root@vbox ~]# cat /proc/partitions
major minor  #blocks  name

 11         0    1048575  sr0
   8         0   20971520  sda
   8         1    1048576  sda1
   8         2   19921920  sda2
  11         1    1048575  sr1
   8        16     524288  sdb
   8        17     102400  sdb1
   8        18         1  sdb2
   8        21     103424  sdb5
   8        22     102400  sdb6
   8        32     524288  sdc
 253         0   17821696  dm-0
 253         1    2097152  dm-1
```

Рис. 4.2. Завершение процедуры и обновление таблицы разделов ядра, просмотр информации.

Продолжим просмотр информации о добавленных разделах:

fdisk --list /dev/sdb

Отформатируем раздел подкачки, используя команду **mkswap /dev/sdb6**. Для включения вновь выделенного пространства подкачки используем **swapon /dev/sdb6**. Для просмотра размера пространства подкачки, которое в настоящее время выделено, введём **free -m** (Рис. 4.3):

```

[200 1 200102 dm 1
[root@vbox ~]# fdisk --list /dev/sdb
Диск /dev/sdb: 512 MiB, 536870912 байт, 1048576 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x8ad3a9ba

Устр-во    Загрузочный  начало    Конеч  Секторы  Размер  Идентификатор  Тип
/dev/sdb1      2048    206847    204800    100M      83 Linux
/dev/sdb2      206848  1048575    841728    411M      5 Расширенный
/dev/sdb5      208896    415743    206848    101M      83 Linux
/dev/sdb6      417792    622591    204800    100M      82 Linux swap /

[root@vbox ~]# mkswap /dev/sdb6
Setting up swapspace version 1, size = 100 MiB (104853504 bytes)
без метки, UUID=ae7c0052-2b0c-4694-b4c0-c2d15739b860
[root@vbox ~]# swapon /dev/sdb6
[root@vbox ~]# free -m

```

	total	used	free	shared	buff/cache	available
Mem:	1775	1100	76	17	771	675
Swap:	2147	0	2147			

```

[root@vbox ~]#

```

Рис. 4.3. Продолжение просмотра информации о добавленных разделах, форматирование раздела подкачки, включение выделенного пространства подкачки. Просмотр размера пространства подкачки, которое в настоящее время выделено.

Создание разделов GPT с помощью gdisk:

В терминале с полномочиями администратора с помощью gdisk посмотрим таблицы разделов и разделы на втором добавленном нами ранее диске /dev/sdc (Рис. 5.1):

gdisk -l /dev/sdc

```

[root@vbox ~]# gdisk -l /dev/sdc
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.
Disk /dev/sdc: 1048576 sectors, 512.0 MiB
Model: VBOX HARDDISK
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): FE8CBBBC-7546-44B1-AC96-25CCE7B7E7F6
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 1048542
Partitions will be aligned on 2048-sector boundaries
Total free space is 1048509 sectors (512.0 MiB)

Number  Start (sector)    End (sector)  Size      Code  Name
[root@vbox ~]#

```

Рис. 5.1. Просмотр таблиц разделов и разделы на втором добавленном ранее диске.

Создадим раздел с помощью `gdisk`: **`gdisk /dev/sdc`**. Введём «**n**», чтобы добавить новый раздел (принимаем номер раздела по умолчанию, который предлагается). Теперь нас просят задать первый сектор. Нажмем «**Enter**», чтобы принять предлагаемый по умолчанию первый сектор. Чтобы создать раздел диска размером 100 MiB, используем **+100M**. Нажмём «**Enter**», чтобы принять тип раздела 8300 по умолчанию (Рис. 5.2):

```
[root@vbox ~]# gdisk /dev/sdc
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.

Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-1048542, default = 2048) or {+-}size{KMGT}:
Last sector (2048-1048542, default = 1048542) or {+-}size{KMGT}: +100M
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
```

Рис. 5.2. Создание раздела с помощью gdisk, добавление нового раздела.

Теперь раздел создан (но ещё не записан на диск). Нажмём «**p**», чтобы отобразить разбиение диска и нажмём «**w**», чтобы записать изменения на диск (Рис. 5.3):

```
Command (? for help): p
Disk /dev/sdc: 1048576 sectors, 512.0 MiB
Model: VBOX HARDDISK
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 8FE40E20-A009-4669-9A77-ABFEFDF63E51
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 1048542
Partitions will be aligned on 2048-sector boundaries
Total free space is 843709 sectors (412.0 MiB)

Number  Start (sector)    End (sector)  Size      Code  Name
   1           2048         206847   100.0 MiB   8300   Linux filesystem

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/sdc.
The operation has completed successfully.
[root@vbox ~]#
```

Рис. 5.3. Отображение разбиения диска, запись изменений на диск.

Обновим таблицу разделов:

partprobe /dev/sdc

Просмотрим информацию о добавленных разделах (Рис. 5.4):

cat /proc/partitions

gdisk -l /dev/sdc

```
root@vbox:~
The operation has completed successfully.
[root@vbox ~]# partprobe /dev/sdc
[root@vbox ~]# cat /proc/partitions
major minor #blocks name
11          0    1048575 sr0
 8          0   20971520 sda
 8          1    1048576 sda1
 8          2   19921920 sda2
11          1    1048575 rr1
 8         16    524288 sdb
 8         17     102400 sdb1
 8         18           1 sdb2
 8         21    103424 sdb5
 8         22     102400 sdb6
 8         32    524288 sdc
 8         33     102400 sdc1
253          0   17821696 dm-0
253          1    2097152 dm-1
[root@vbox ~]# gdisk -l /dev/sdc
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: protective
```

Рис. 5.4. Обновление таблицы разделов, просмотр информации о добавленных разделах.

Форматирование файловой системы XFS:

В терминале с полномочиями администратора для диска `dev/sdb1` создадим файловую систему XFS: **`mkfs.xfs /dev/sdb1`**. Для установки метки файловой системы в `xfsdisk` используем команду **`xfs_admin -L xfsdisk /dev/sdb1`** (Рис. 6):


```

[root@vbox ~]# mkfs.xfs /dev/sdb1
Filesystem should be larger than 300MB.
Log size should be at least 64MB.
Support for filesystems like this one is deprecated and they will not be support
ed in future releases.
meta-data=/dev/sdb1            isize=512    agcount=4, agsize=6400 blks
      =                       sectsz=512    attr=2, projid32bit=1
      =                       crc=1        finobt=1, sparse=1, rmapbt=0
      =                       reflink=1    bigtime=1 inobtcount=1 nrext64=0
data      =                       bsize=4096   blocks=25600, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0, ftype=1
log        =internal log      bsize=4096   blocks=1368, version=2
      =                       sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none                extsz=4096   blocks=0, rtextents=0
[root@vbox ~]# xfs_admin -L xfsdisk /dev/sdb1
writing all SBs
new label = "xfsdisk"
[root@vbox ~]#

```

Рис. 6. Создание файловой системы XFS, установка метки файловой системы в xfsdisk.

Форматирование файловой системы EXT4:

В терминале с полномочиями администратора для диска `dev/sdb5` создадим файловую систему EXT4: **`mkfs.ext4 /dev/sdb5`**. Для установки метки файловой системы в `ext4disk` используем команду **`tune2fs -L ext4disk /dev/sdb5`**. Для установки параметров монтирования по умолчанию для файловой системы используем команду **`tune2fs -o acl,user_xattr /dev/sdb5`**. В данном случае включены списки контроля доступа и расширенные атрибуты пользователя (Рис. 7):

```
[root@vbox ~]# mkfs.ext4 /dev/sdb5
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 103424 1k blocks and 25896 inodes
Filesystem UUID: b51e19f4-4c47-44e5-bbc5-513c273a4e0e
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

[root@vbox ~]# tune2fs -L ext4disk /dev/sdb5
tune2fs 1.46.5 (30-Dec-2021)
[root@vbox ~]# tune2fs -o acl,user_xattr /dev/sdb5
tune2fs 1.46.5 (30-Dec-2021)
```

Рис. 7. Создание файловой системы EXT4, установка метки файловой системы в ext4disk, установка параметров монтирования по умолчанию для файловой системы.

Ручное монтирование файловых систем:

Для создания точки монтирования для раздела введём **mkdir -p /mnt/tmp**. Чтобы смонтировать файловую систему, используем следующую команду **mount /dev/sdb5 /mnt/tmp**. Для проверки корректности монтирования раздела введём (Рис. 8.1): **mount**

```
[root@vbox ~]# mkdir -p /mnt/tmp
[root@vbox ~]# mount /dev/sdb5 /mnt/tmp
[root@vbox ~]# mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=219606,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=363672k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
```

Рис. 8.1. Создание точки монтирования для раздела, монтирование файловой системы, проверка корректности монтирования раздела.

Чтобы отмонтировать раздел введём: **umount /dev/sdb5**. Проверим, что раздел отмонтирован: **mount** (Рис. 8.2):

```
[root@vbox ~]# umount /dev/sdb5
[root@vbox ~]# mount
```

Рис. 8.2. Монтирование раздела, проверка.

Монтирование разделов с помощью /etc/fstab:

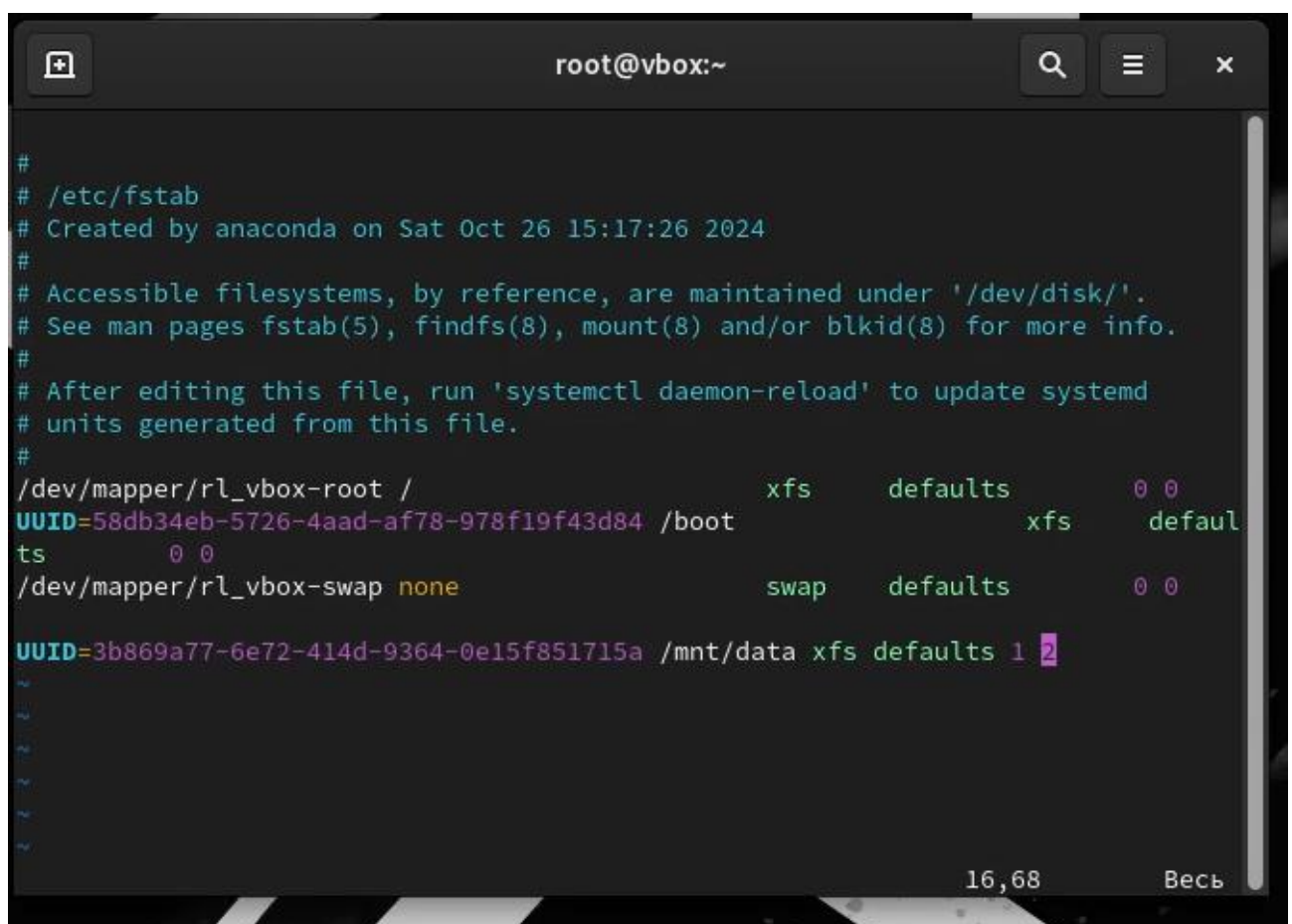
Создадим точку монтирования для раздела XFS /dev/sdb1: **mkdir -p /mnt/data**. Просмотрим информацию об идентификаторах блочных устройств (UUID): **blkid**. Введём **blkid /dev/sdb1** и затем используем мышь, чтобы скопировать значение идентификатора UUID для устройства /dev/sdb1. Откроем файл /etc/fstab на редактирование в текстовом редакторе vim (Рис. 9.1):

```
[root@vbox ~]# mkdir -p /mnt/data
[root@vbox ~]# blkid
/dev/mapper/rl_vbox-swap: UUID="0dae8fbd-1b8a-4258-a36d-1039dfc590f4" TYPE="swap"
/dev/sdb5: LABEL="ext4disk" UUID="b51e19f4-4c47-44e5-bbc5-513c273a4e0e" TYPE="ext4" PARTUUID="8ad3a9ba-05"
/dev/sdb1: LABEL="xfsdisk" UUID="3b869a77-6e72-414d-9364-0e15f851715a" TYPE="xfs" PARTUUID="8ad3a9ba-01"
/dev/sdb6: UUID="ae7c0052-2b0c-4694-b4c0-c2d15739b860" TYPE="swap" PARTUUID="8ad3a9ba-06"
/dev/mapper/rl_vbox-root: UUID="358e5f4d-cbaa-40d8-83e0-29de8a262edd" TYPE="xfs" PARTUUID="8ad3a9ba-03"
/dev/sdc1: PARTLABEL="Linux filesystem" PARTUUID="db1cace7-e41a-493c-bbcb-2f54e7d4f960"
/dev/sda2: UUID="1YYesJ-YxQ3-0KHb-GvWb-S6sp-wRWR-jkiCSc" TYPE="LVM2_member" PARTUUID="4eae4f4e-02"
/dev/sda1: UUID="58db34eb-5726-4aad-af78-978f19f43d84" TYPE="xfs" PARTUUID="4eae4f4e-01"
[root@vbox ~]# blkid /dev/sdb1
/dev/sdb1: LABEL="xfsdisk" UUID="3b869a77-6e72-414d-9364-0e15f851715a" TYPE="xfs" PARTUUID="8ad3a9ba-01"
[root@vbox ~]# vim /etc/fstab
```

Рис. 9.1. Создание точки монтирования для раздела XFS, просмотр информации об идентификаторах блочных устройств, копирование значения идентификатора UUID для устройства, открытие файла /etc/fstab на редактирование в текстовом редакторе vim.

Добавим следующую строку в открытый файл (Рис. 9.2):

UUID=значение_идентификатора /mnt/data xfs defaults 1 2



```
#  
# /etc/fstab  
# Created by anaconda on Sat Oct 26 15:17:26 2024  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.  
#  
# After editing this file, run 'systemctl daemon-reload' to update systemd  
# units generated from this file.  
#  
/dev/mapper/rl_vbox-root / xfs defaults 0 0  
UUID=58db34eb-5726-4aad-af78-978f19f43d84 /boot xfs default  
ts 0 0  
/dev/mapper/rl_vbox-swap none swap defaults 0 0  
UUID=3b869a77-6e72-414d-9364-0e15f851715a /mnt/data xfs defaults 1 2  
~  
~  
~  
~  
16,68 Весь
```

Рис. 9.2. Добавление строки в файл.

Введём команду, которая монтирует всё, что указано в /etc/fstab: **mount -a**.
Проверим, что раздел примонтирован правильно: **df -h** (Рис. 9.3):

```
[root@vbox ~]# vim /etc/fstab
[root@vbox ~]# mount -a
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[root@vbox ~]# df -h
```

Файловая система	Размер	Использовано	Дост	Использовано%	Смонтировано в
devtmpfs	4,0M	0	4,0M	0%	/dev
tmpfs	888M	0	888M	0%	/dev/shm
tmpfs	356M	5,6M	350M	2%	/run
/dev/mapper/rl_vbox-root	17G	5,4G	12G	32%	/
/dev/sda1	960M	409M	552M	43%	/boot
tmpfs	178M	100K	178M	1%	/run/user/1000
/dev/sdb1	95M	6,0M	89M	7%	/mnt/data

Рис. 9.3. Монтирование всего, что указано в /etc/fstab. Проверка.

Самостоятельная работа:

Добавим две партии на диск с разбиением GPT. Создадим оба раздела размером 100 MiB. Один из этих разделов будет настроен как пространство подкачки, другой раздел будет отформатирован файловой системой ext4.

Создадим первый раздел (Рис. 10.1):

```
[root@vbox ~]# gdisk /dev/sdc
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.

Command (? for help): n
Partition number (2-128, default 2): 2
First sector (34-1048542, default = 206848) or {+-}size{KMGT}:
Last sector (206848-1048542, default = 1048542) or {+-}size{KMGT}: +100M
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): l
Type search string, or <Enter> to show all codes:
```

Рис. 10.1. Создание первого раздела.

Отформатируем первый раздел (Рис. 10.2):

```
[root@vbox ~]# mkfs.ext4 /dev/sdc2
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 102400 1k blocks and 25584 inodes
Filesystem UUID: d14be34e-80b7-4917-b282-6f9acb428796
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Рис. 10.2. Форматирование первого раздела.

Создадим второй раздел (Рис. 10.3):

```
[root@vbox ~]# gdisk /dev/sdc
GPT fdisk (gdisk) version 1.0.7

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present
```

Рис. 10.3. Создание второго раздела.

Создадим второй раздел (Рис. 10.4):


```

Command (? for help): n
Partition number (3-128, default 3): 3
First sector (34-1048542, default = 411648) or {+-}size{KMGTP}:
Last sector (411648-1048542, default = 1048542) or {+-}size{KMGTP}: +100M
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): 8300
Changed type of partition to 'Linux filesystem'

Command (? for help): w

Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): y
OK; writing new GUID partition table (GPT) to /dev/sdc.
The operation has completed successfully.

```

Рис. 10.4. Создание второго раздела.

Настроим второй раздел как swar-пространство (Рис. 10.5):

```

[root@vbox ~]# swapon /dev/sdc3
swapon: /dev/sdc3: read swap header failed
[root@vbox ~]# free -m

```

	total	used	free	shared	buff/cache	available
Mem:	1775	1094	109	17	741	680
Swap:	2147	0	2147			

Рис. 10.5. Настройка второго раздела как swar-пространство. (Я понял свою ошибку и забыл написать команду mkswap)

Настроим сервер для автоматического монтирования этих разделов. Установим раздел ext4 на /mnt/data-ext и установим пространство подкачки в качестве области подкачки (Рис. 10.6), (Рис. 10.7), (Рис. 10.6).

```
root@vbox:~  
/dev/sdb6: UUID="ae7c0052-2b0c-4694-b4c0-c2d15739b860" TYPE="swap" PARTUUID="8ad3a9ba-06"  
/dev/mapper/rl_vbox-root: UUID="358e5f4d-cbaa-40d8-83e0-29de8a262edd" TYPE="xfs"  
/dev/sda2: UUID="1YYesJ-YxQ3-0KHb-GvWb-S6sp-wRWR-jkiCSc" TYPE="LVM2_member" PART  
UUID="4eaefbe4-02"  
/dev/sda1: UUID="58db34eb-5726-4aad-af78-978f19f43d84" TYPE="xfs" PARTUUID="4eae  
fbe4-01"  
/dev/sdc2: UUID="d14be34e-80b7-4917-b282-6f9acb428796" TYPE="ext4" PARTLABEL="Li  
nux swap" PARTUUID="cbf17328-3835-42fe-8b1b-16a6616b20be"  
/dev/sdc3: UUID="233bcbbe-ab0d-41a5-ad2f-dcf865babce6" TYPE="ext4" PARTLABEL="Li  
nux filesystem" PARTUUID="d24b9eb3-ccdf-4496-ab94-50ee36df6828"  
/dev/sdc1: PARTLABEL="Linux filesystem" PARTUUID="db1cace7-e41a-493c-bbcb-2f54e7  
d4f960"  
[root@vbox ~]# vim /mnt/data-ext  
[root@vbox ~]# mkdir -p /mnt/data-ext  
[root@vbox ~]# vim /mnt/data-ext  
[root@vbox ~]# mcedit /etc/fstab  
bash: mcedit: команда не найдена...  
Установить пакет «mc», предоставляющий команду «mcedit»? [N/y] n  
  
[root@vbox ~]# bim /etc/fstab  
bash: bim: команда не найдена...  
[root@vbox ~]# vim /etc/fstab
```

Рис. 10.6. Настройка сервера для автоматического монтирования разделов.

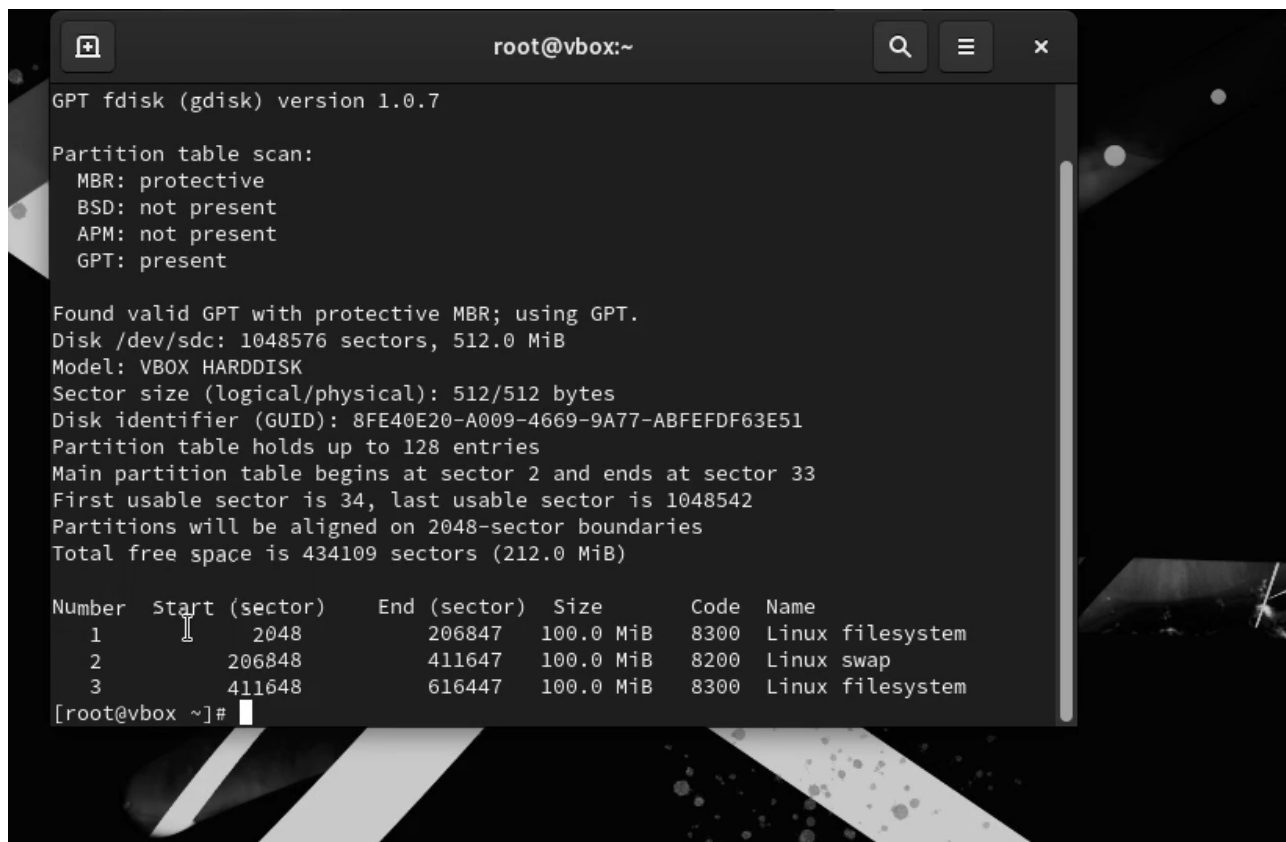

```
root@vbox:~  
#  
# /etc/fstab  
# Created by anaconda on Sat Oct 26 15:17:26 2024  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.  
#  
# After editing this file, run 'systemctl daemon-reload' to update systemd  
# units generated from this file.  
#  
/dev/mapper/rl_vbox-root / xfs defaults 0 0  
UUID=58db34eb-5726-4aad-af78-978f19f43d84 /boot xfs default  
ts 0 0  
/dev/mapper/rl_vbox-swap none swap defaults 0 0  
  
UUID=3b869a77-6e72-414d-9364-0e15f851715a /mnt/data xfs defaults 1 2  
UUID=d14be34e-80b7-4917-b282-6f9acb428796 /mnt/data-ext4 ext4 defaults 1 2  
~  
~  
~  
-- ВСТАВКА -- 18,56 Весь
```

Рис. 10.7. Настройка сервера для автоматического монтирования разделов.

```
[root@vbox ~]# mount -a  
mount: (hint) your fstab has been modified, but systemd still uses  
the old version; use 'systemctl daemon-reload' to reload.  
[root@vbox ~]# df -h  
Файловая система    Размер  Использовано  Дост  Использовано%  Смонтировано в  
devtmpfs             4,0М      0      4,0М           0% /dev  
tmpfs                888М      0      888М           0% /dev/shm  
tmpfs               356М    5,7М    350М           2% /run  
/dev/mapper/rl_vbox-root 17G    5,2G    12G          31% /  
/dev/sda1            960М    409М    552М          43% /boot  
tmpfs               178М    100К    178М           1% /run/user/1000  
/dev/sdb1             95М      6,0М     89М           7% /mnt/data  
/dev/sdc2             89М      14К     82М           1% /mnt/data-ext  
[root@vbox ~]#
```

Рис. 10.8. Настройка сервера для автоматического монтирования разделов.

Перезагрузим систему и убедимся, что всё установлено правильно (Рис. 10.9):



```
root@vbox:~  
GPT fdisk (gdisk) version 1.0.7  
  
Partition table scan:  
  MBR: protective  
  BSD: not present  
  APM: not present  
  GPT: present  
  
Found valid GPT with protective MBR; using GPT.  
Disk /dev/sdc: 1048576 sectors, 512.0 MiB  
Model: VBOX HARDDISK  
Sector size (logical/physical): 512/512 bytes  
Disk identifier (GUID): 8FE40E20-A009-4669-9A77-ABFEFDF63E51  
Partition table holds up to 128 entries  
Main partition table begins at sector 2 and ends at sector 33  
First usable sector is 34, last usable sector is 1048542  
Partitions will be aligned on 2048-sector boundaries  
Total free space is 434109 sectors (212.0 MiB)  
  
Number  start (sector)  End (sector)  Size    Code  Name  
  1             2048        206847    100.0 MiB  8300  Linux filesystem  
  2          206848        411647    100.0 MiB  8200  Linux swap  
  3          411648        616447    100.0 MiB  8300  Linux filesystem  
[root@vbox ~]#
```

Рис. 10.9. Перезагрузка системы и проверка.

Ответы на контрольные вопросы:

1. Какой инструмент используется для создания разделов GUID? **gdisk**
2. Какой инструмент применяется для создания разделов MBR? **fdisk**
3. Какая файловая система используется по умолчанию для систем типа RHEL? **XFS**
4. Какой файл используется для автоматического монтирования разделов во время загрузки? **/etc/fstab**

5. Какой вариант монтирования целесообразно выбрать, если необходимо, чтобы файловая система не была автоматически примонтирована во время загрузки? **mount /dev/sdb5/mnt/tmp**

6. Какая команда позволяет форматировать раздел с типом 82 с соответствующей файловой системой? **t**

7. Вы только что добавили несколько разделов для автоматического монтирования при загрузке. Как можно безопасно проверить, будет ли это работать без реальной перезагрузки? **df -h**

8. Какая файловая система создаётся, если вы используете команду **mkfs** без какой-либо спецификации файловой системы? **swap**

9. Как форматировать раздел EXT4?

mkfs.ext4 /dev/sdb<number>

tune2fs -L ext4disk /dev/sdb<number>

tune2fs -o acl,user_xattr /dev/sdb<number>

10. Как найти UUID для всех устройств на компьютере? **blkid**

Вывод:

В ходе выполнения лабораторной работы были получены навыки создания разделов на диске и файловых систем, а также навыки монтирования файловых систем.