

Лабораторная работа №14

Жукова Арина Александровна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Создание виртуальных носителей	6
2.2	Создание разделов MBR с помощью fdisk	7
2.3	Создание логических разделов	9
2.4	Создание раздела подкачки	10
2.5	Создание разделов GPT с помощью gdisk	12
2.6	Форматирование файловой системы XFS	14
2.7	Форматирование файловой системы EXT4	15
2.8	Ручное монтирование файловых систем	15
2.9	Монтирование разделов с помощью /etc/fstab	17
2.10	Самостоятельная работа	18
2.11	Ответы на контрольные вопросы	19
3	Выводы	22
	Список литературы	23

Список иллюстраций

2.1	Добавление жестких дисков	6
2.2	Перечень жестких дисков	7
2.3	Справка по командам	8
2.4	Создаём новый раздел	8
2.5	Записываем изменения на диск	9
2.6	Сохранение в таблицу	9
2.7	Добавление расширенного раздела	10
2.8	Обновление таблиц, просмотр информации	10
2.9	Добавление нового раздела	11
2.10	Обновляем таблицу ядра	11
2.11	Форматирование раздела подкачки	12
2.12	Просмотр разделов	12
2.13	Добавление нового раздела	13
2.14	Обновление таблицы, просмотр информации	14
2.15	Создание файловой системы, установка метки файловой системы	15
2.16	Создание файловой системы	15
2.17	Монтируем файловую систему	16
2.18	Монтирование раздела	16
2.19	Создание точки монтирования	17
2.20	Редактирование файла	17
2.21	Проверка примонтирование	18
2.22	Создание партиций	18
2.23	Форматирование файловой системой	19
2.24	Настройка как пространство прокачки	19
2.25	Настройка сервера	19

Список таблиц

1 Цель работы

Получить навыки создания разделов на диске и файловых систем. Получить навыки монтирования файловых систем.

2 Выполнение лабораторной работы

2.1 Создание виртуальных носителей

Добавляем к виртуальной машине два диска размером 512 МБ (рис. 2.1).

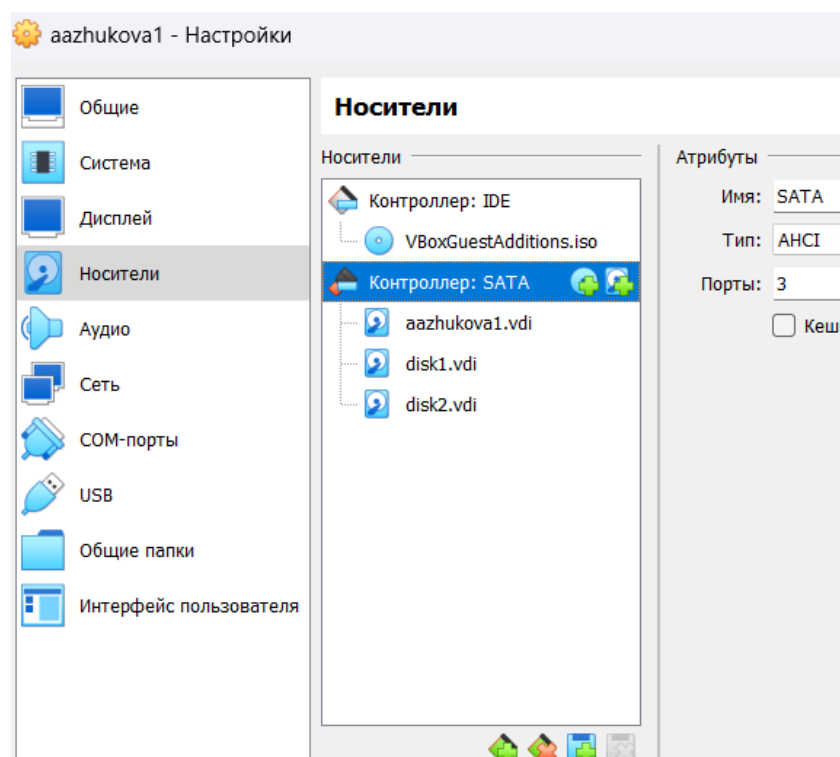


Рис. 2.1: Добавление жестких дисков

2.2 Создание разделов MBR с помощью fdisk

1. В командной строке с полномочиями администратора с помощью fdisk просматриваю перечень разделов на всех имеющихся в системе устройствах жёстких дисков (рис. 2.2).

```
[aazhukoval@aazhukoval ~]$ su -
Пароль:
[root@aazhukoval ~]# fdisk --list
Диск /dev/sda: 40 GiB, 42949672960 байт, 83886080 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: 0x7c41a9df

Устр-во   Загрузочный  начало      Конец      Секторы  Размер  Идентификатор  Тип
/dev/sda1 *                2048      2099199    2097152      1G        83 Linux
/dev/sda2                2099200    83886079    81786880     39G        8e Linux LVM

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

Диск /dev/sdc: 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

Диск /dev/mapper/rl-root: 35,05 GiB, 37631295488 байт, 73498624 секторов
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт

Диск /dev/mapper/rl-swap: 3,95 GiB, 4240441344 байт, 8282112 секторов
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
[root@aazhukoval ~]#
```

Рис. 2.2: Перечень жестких дисков

2. Необходимо сделать разметку диска /dev/sdb с помощью утилиты fdisk. Ввожу m, чтобы получить справку по командам (рис. 2.3).

```
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 0x62ab59c3.

Команда (m для справки): m

Справка:

DOS (MBR)
a  переключение флага загрузки
b  редактирование вложенной метки диска BSD
c  переключение флага dos-совместимости

Общие
d  удалить раздел
F  показать свободное неразмеченное пространство
l  список известных типов разделов
n  добавление нового раздела
p  вывести таблицу разделов
t  изменение типа раздела
v  проверка таблицы разделов
i  вывести информацию о разделе
```

Рис. 2.3: Справка по командам

3. Нажимаем **p**, чтобы просмотреть текущее распределение пространства диска. Вводим **n**, чтобы добавить новый раздел. Выбираем **p**, чтобы создать основной раздел. Указываем первый сектор на диске, с которого начнётся новый раздел. По умолчанию предлагается первый доступный сектор, нажимаем **Enter** для подтверждения выбора. Вводим **+100M**, чтобы создать раздел на 100 MiB (рис. 2.4).

```
Команда (m 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: 0x62ab59c3

Команда (m 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):
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.
```

Рис. 2.4: Создаём новый раздел

4. Нажимаем **w**, чтобы записать изменения на диск и выйти из fdisk (рис. 2.5).


```

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

```

Рис. 2.5: Записываем изменения на диск

5. Таблица разделов находится только в памяти ядра. Сравниваем вывод команды `fdisk -l /dev/sdb` с выводом команды `cat /proc/partitions`. Записываем изменения в таблицу разделов ядра (рис. 2.6).

```

[root@aazhukoval ~]# fdisk -l /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: 0x62ab59c3

Устр-во   Загрузочный  начало  Конец  Секторы  Размер  Идентификатор  Тип
/dev/sdb1          2048 206847  204800    100M      83 Linux
[root@aazhukoval ~]# cat /proc/partitions
major minor #blocks name
 8         0  41943040 sda
 8         1   1048576 sda1
 8         2  40893440 sda2
 8        16   524288 sdb
 8        17   102400 sdb1
 8        32   524288 sdc
11         0    52250 sr0
253        0  36749312 dm-0
253        1  4141056 dm-1
[root@aazhukoval ~]# partprobe /dev/sdb

```

Рис. 2.6: Сохранение в таблицу

2.3 Создание логических разделов

1. Добавляем новый раздел. Вводим `e`, чтобы создать расширенный раздел. Нажимаем `Enter`, чтобы принять первый сектор по умолчанию и снова нажимаем `Enter`, когда `fdisk` запросит последний сектор. Создаём в нём логический раздел. Нажимаем `n`. Свободных первичных разделов нет, добавляем логический раздел с номером 5. Нажимаем `Enter`, чтобы принять выбор первого сектора в качестве сектора по умолчанию. На вопрос о последнем

секторе вводим +101M. После создания логического раздела записываем изменения на диск (рис. 2.7).

```
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 (т 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.

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

Рис. 2.7: Добавление расширенного раздела

2. Обновляем таблицу разделов, просматриваем информацию о добавленных разделах (рис. 2.8).

```
[root@aazhukoval ~]# partprobe /dev/sdb
[root@aazhukoval ~]# cat /proc/partitions
major minor #blocks name
8 0 41943040 sda
8 1 1048576 sda1
8 2 40893440 sda2
8 16 524288 sdb
8 17 102400 sdb1
8 18 1 sdb2
8 21 103424 sdb5
8 32 524288 sdc
11 0 52250 sr0
253 0 36749312 dm-0
253 1 4141056 dm-1
```

Рис. 2.8: Обновление таблиц, просмотр информации

2.4 Создание раздела подкачки

1. Добавляем новый раздел. По умолчанию добавляем логический раздел под номером 6. Нажимаем Enter, чтобы принять первый сектор по умолчанию. На вопрос о последнем секторе вводим +100M. Далее меняем тип раздела.

Для этого нажимаем **t**, вводим номер раздела и тип раздела (в данном случае **82** — раздел подкачки). Записываем изменения (рис. 2.9).

```

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
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. 2.9: Добавление нового раздела

2. Чтобы завершить процедуру и обновляем таблицу разделов ядра, просматриваем информацию о добавленных разделах (рис. 2.10).

```

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

 8         0  41943040 sda
 8         1   1048576 sda1
 8         2  40893440 sda2
 8        16   524288 sdb
 8        17   102400 sdb1
 8        18         1 sdb2
 8        21   103424 sdb5
 8        22   102400 sdb6
 8        32   524288 sdc
11         0    52250 sr0
253        0  36749312 dm-0
253        1   4141056 dm-1

[root@aazhukoval ~]# 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: 0x62ab59c3

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 / Solaris

```

Рис. 2.10: Обновляем таблицу ядра

3. Форматируем раздел подкачки. Для включения вновь выделенного пространства подкачки используем `swapon /dev/sdb6`. Просматриваем размер пространства подкачки, которое выделено в настоящее время (рис. 2.11).

```
[root@aazhukoval ~]# mkswap /dev/sdb6
Setting up swapspace version 1, size = 100 MiB (104853504 bytes)
6e3 метки, UUID=e54faf6b-4011-4bfe-88f3-339b7b3f19d1
[root@aazhukoval ~]# swapon /dev/sdb6
[root@aazhukoval ~]# free -m
```

	total	used	free	shared	buff/cache	available
Mem:	3658	1191	1897	15	804	2467
Swap:	4143	0	4143			

```
[root@aazhukoval ~]#
```

Рис. 2.11: Форматирование раздела подкачки

2.5 Создание разделов GPT с помощью gdisk

1. В терминале с полномочиями администратора с помощью gdisk просматриваем таблицы разделов и разделы на втором добавленном диске /dev/sdc (рис. 2.12).

Просмотр разделов

Рис. 2.12: Просмотр разделов

2. Создаём раздел с помощью gdisk. Вводим n , чтобы добавить новый раздел. Применяем номер раздела по умолчанию, который предлагается. Нажимаем Enter, чтобы принять предлагаемый по умолчанию первый сектор. При запросе последнего сектора вводим +100M. Теперь предлагается установить тип раздела. Нажимаем Enter, чтобы принять тип раздела 8300 по умолчанию. Нажимаем р, чтобы отобразить разбиение диска. Записываем изменение диска (рис. 2.13).

```

[root@aazhukoval ~]# 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):
First sector (34-1048542, default = 2048) or {+}-size{KMGTP}:
Last sector (2048-1048542, default = 1048542) or {+}-size{KMGTP}: +100M
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

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): 9286F61F-D496-43BA-B044-4C42EFE7DC9B
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.

```

Рис. 2.13: Добавление нового раздела

3. Обновляем таблицу раздела ядра и просматриваем информацию о добавленных разделах (рис. 2.14).

```

[root@aazhukoval ~]# partprobe /dev/sdb
[root@aazhukoval ~]# cat /proc/partitions
major minor #blocks name
8        0      41943040 sda
8        1      1048576 sda1
8        2      40893440 sda2
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
11        0        52250 sr0
253       0      36749312 dm-0
253       1      4141056 dm-1
[root@aazhukoval ~]# gdisk -l /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.
Disk /dev/sdc: 1048576 sectors, 512.0 MiB
Model: VBOX HARDDISK
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 9286F61F-D496-43BA-B044-4C42EFE7DC9B
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
[root@aazhukoval ~]#

```

Рис. 2.14: Обновление таблицы, просмотр информации

2.6 Форматирование файловой системы XFS

1. В терминале с полномочиями администратора для диска `dev/sdb1` создаём файловую систему XFS, устанавливаем метки файловой системы в `xfsdisk` (рис. 2.15).

```
[root@aazhukoval ~]# 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 supported 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 nnext64=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@aazhukoval ~]# xfs_admin -L xfsdisk /dev/sdb1
writing all SBs
new label = "xfsdisk"
```

Рис. 2.15: Создание файловой системы, установка метки файловой системы

2.7 Форматирование файловой системы EXT4

1. В терминале с полномочиями администратора для диска `dev/sdb5` создаём файловую систему EXT4. Устанавливаем метки файловой системы в `ext4disk`, параметры монтирования по умолчанию для файловой системы (рис. 2.16).

```
[root@aazhukoval ~]# mkfs.ext4 /dev/sdb5
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 103424 1k blocks and 25896 inodes
Filesystem UUID: fd1958a6-9920-47ac-a7e3-1c040e0754a3
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@aazhukoval ~]# tune2fs -L ext4disk /dev/sdb5
tune2fs 1.46.5 (30-Dec-2021)
[root@aazhukoval ~]#
[root@aazhukoval ~]#
[root@aazhukoval ~]# tune2fs -o acl,user_xattr /dev/sdb5
tune2fs 1.46.5 (30-Dec-2021)
```

Рис. 2.16: Создание файловой системы

2.8 Ручное монтирование файловых систем

1. Создаём точки монтирования для раздела. Монтируем файловую систему. Для проверки корректности монтирования раздела ввожу `mount` (рис. 2.17).

```
[root@aazhukoval ~]# mkdir -p /mnt/tmp
[root@aazhukoval ~]# mount /dev/sdb5 /mnt/tmp
[root@aazhukoval ~]# 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=460412,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=749300k,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)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/mapper/rl-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=19607)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
none on /run/credentials/systemd-tmpfiles-setup-dev.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
none on /run/credentials/systemd-sysctl.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
/dev/sda1 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
none on /run/credentials/systemd-tmpfiles-setup.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
public on /media/sf_public type vboxsf (rw,nodev,relatime,ioccharset=utf8,uid=0,gid=978,dmode=0770,fmode=0770,tag=VBBox,Automounter)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=374648k,nr_inodes=93662,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/dev/sr0 on /run/media/aazhukoval/VBox_GAs_7.0.20 type iso9660 (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,uhelper=udisks2)
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/dev/sdb5 on /mnt/tmp type ext4 (rw,relatime,seclabel)
```

Рис. 2.17: Монтируем файловую систему

2. Чтобы отмонтировать раздел, используем `umount` с именем точки монтирования. Проверяем монтировку (рис. 2.18).

```
[root@aazhukoval ~]# umount /dev/sdb5
[root@aazhukoval ~]# 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=460412,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=749300k,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)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/mapper/rl-root on / type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=19607)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
none on /run/credentials/systemd-tmpfiles-setup-dev.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
none on /run/credentials/systemd-sysctl.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
/dev/sda1 on /boot type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
none on /run/credentials/systemd-tmpfiles-setup.service type ramfs (ro,nosuid,nodev,noexec,relatime,seclabel,mode=700)
public on /media/sf_public type vboxsf (rw,nodev,relatime,ioccharset=utf8,uid=0,gid=978,dmode=0770,fmode=0770,tag=VBBox,Automounter)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=374648k,nr_inodes=93662,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/dev/sr0 on /run/media/aazhukoval/VBox_GAs_7.0.20 type iso9660 (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,uhelper=udisks2)
portal on /run/user/1000/doc type fuse.portal (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
```

Рис. 2.18: Монтирование раздела

2.9 Монтирование разделов с помощью /etc/fstab

1. Создайте точку монтирования для раздела XFS /dev/sdb1. Просматриваем информацию об идентификаторах блочных устройств (UUID). Вводим `blkid /dev/sdb1` и затем используйте мышшь, чтобы скопировать значение идентификатора UUID для устройства /dev/sdb1 (рис. 2.19).

```
[root@aaazhukoval ~]# mkdir -p /mnt/data
[root@aaazhukoval ~]# blkid
/dev/mapper/rl-swap: UUID="587781c7-0da7-4cfb-8ed0-8bd4c052e88e" TYPE="swap"
/dev/sdb5: LABEL="ext4disk" UUID="fd1958a6-9920-47ac-a7e3-1c040e0754a3" TYPE="ext4" PARTUUID="62ab59c3-05"
/dev/sdb1: LABEL="xfsdisk" UUID="87d4e259-91df-4fc7-925d-828ee537b2dd" TYPE="xfs" PARTUUID="62ab59c3-01"
/dev/sdb6: UUID="e54faf6b-4011-4bfe-88f3-339b7b3f19d1" TYPE="swap" PARTUUID="62ab59c3-06"
/dev/sr0: UUID="2024-07-10-14-17-04-74" LABEL="VBox_GAs_7.0.20" TYPE="iso9660"
/dev/mapper/rl-root: UUID="9a08951b-1031-43f4-bbfc-f8b9eb5173f8" TYPE="xfs"
/dev/sdc1: PARTLABEL="Linux filesystem" PARTUUID="102da2b6-b004-4ff3-93ed-01a143690b07"
/dev/sda2: UUID="9k12Wn-ivLU-ZmoZ-gelZ-Bc4h-87u2-0lbdMu" TYPE="LVM2_member" PARTUUID="7c41a9df-02"
/dev/sda1: UUID="abafd482-bfe0-4a6c-8783-722831625527" TYPE="xfs" PARTUUID="7c41a9df-01"
[root@aaazhukoval ~]# blkid /dev/sdb1
/dev/sdb1: LABEL="xfsdisk" UUID="87d4e259-91df-4fc7-925d-828ee537b2dd" TYPE="xfs" PARTUUID="62ab59c3-01"
[root@aaazhukoval ~]#
```

Рис. 2.19: Создание точки монтирования

2. Редактируем файл /etc/fstab (рис. 2.20).

```
fstab  [-M--]  0 L: [ 1+15 16/ 16] *(648 / 648b) <EOF>
#
# /etc/fstab
# Created by anaconda on Fri Sep  6 09:15:12 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-root / xfs defaults 0 0
UUID=abafd482-bfe0-4a6c-8783-722831625527 /boot xfs defaults 0 0
/dev/mapper/rl-swap none swap defaults 0 0
UUID=87d4e259-91df-4fc7-925d-828ee537b2dd /mnt/data xfs defaults 1 2
```

Рис. 2.20: Редактирование файла

3. Монтируем всё что указано в файле, проверяем что раздел примонтирован правильно (рис. 2.21).

```
[root@aazhukoval ~]# mount -a
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
[root@aazhukoval ~]# df -h
Файловая система    Размер  Использовано  Дост  Использовано%  Смонтировано в
devtmpfs             4,0M      0             4,0M      0% /dev
tmpfs                1,8G      0             1,8G      0% /dev/shm
tmpfs                732M     1,3M          731M      1% /run
/dev/mapper/rl-root   35G      6,4G          29G      19% /
/dev/sda1             960M     521M          440M      55% /boot
public              342G      71G           271G      21% /media/sf_public
tmpfs                366M     104K          366M      1% /run/user/1000
/dev/sr0              52M      52M           0         100% /run/media/aazhukoval/VBox_GAs_7.0.20
/dev/sdb1             95M      6,0M          89M       7% /mnt/data
[root@aazhukoval ~]#
```

Рис. 2.21: Проверка примонтирование

2.10 Самостоятельная работа

1. Добавляем две партии на диск с разбиением GPT. Создаём оба раздела размером 100 MiB. Один из этих разделов настроен как пространство подкачки, другой раздел отформатирован файловой системой ext4 (рис. 2.22-2.24).

```
[root@aazhukoval ~]# 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):
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):
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.
[root@aazhukoval ~]# partprobe /dev/sdc
```

Рис. 2.22: Создание партий

```
[root@aazhukoval ~]# mkfs.ext4 /dev/sdc2
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 102400 1k blocks and 25584 inodes
Filesystem UUID: b7fe946c-e5c8-4e32-bae4-b4f3439ab7eb
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. 2.23: Форматирование файловой системой

```
[root@aazhukoval ~]# mkswap /dev/sdc3
Setting up swapspace version 1, size = 100 MiB (104853504 bytes)
без метки, UUID=69279edd-25c4-4aa0-bd22-f378325fc09c
[root@aazhukoval ~]# swapon /dev/sdc3
[root@aazhukoval ~]# free -m
```

	total	used	free	shared	buff/cache	available
Mem:	3658	1301	1772	15	820	2357
Swap:	4243	0	4243			

Рис. 2.24: Настройка как пространство прокачки

2. Настройте сервер для автоматического монтирования этих разделов (рис. 2.25).

```
fstab [---] 0 L: [ 1+15 16/ 16] *(648 / 648b) <EOF>

#
# /etc/fstab
# Created by anaconda on Fri Sep  6 09:15:12 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-root / xfs defaults 0 0
UUID=abafd482-bfe0-4a6c-8783-722831625527 /boot xfs defaults 0 0
/dev/mapper/rl-swap none swap defaults 0 0
UUID=87d4e259-91df-4fc7-925d-828ee537b2dd /mnt/data xfs defaults 1 2
```

Рис. 2.25: Настройка сервера

2.11 Ответы на контрольные вопросы

1. Какой инструмент используется для создания разделов GUID? Для создания разделов GUID (GUID Partition Table - GPT) обычно используется gdisk (GNU Parted) или подобные инструменты, такие как fdisk с опцией создания GPT

таблицы разделов. Выбор инструмента зависит от операционной системы и предпочтений.

2. Какой инструмент применяется для создания разделов MBR? Для создания разделов MBR (Master Boot Record) чаще всего используется fdisk. Также можно использовать parted, но fdisk более традиционный и распространённый инструмент для работы с MBR.
3. Какой файл используется для автоматического монтирования разделов во время загрузки? Файл /etc/fstab (на системах типа Linux/Unix) содержит информацию о файловых системах, которые должны быть автоматически смонтированы при загрузке.
4. Какой вариант монтирования целесообразно выбрать, если необходимо, чтобы файловая система не была автоматически примонтирована во время загрузки? В файле /etc/fstab необходимо опустить поле или указать noauto в опциях монтирования для данной строки.
5. Какая команда позволяет форматировать раздел с типом 82 с соответствующей файловой системой? Тип 82 обычно ассоциируется с разделами Linux Swap. Для форматирования swap-раздела используется команда mkswap. После создания swap-раздела его необходимо активировать командой swapon.
6. Вы только что добавили несколько разделов для автоматического монтирования при загрузке. Как можно безопасно проверить, будет ли это работать без реальной перезагрузки? Можно использовать команду `sudo mount -a`. Эта команда попытается смонтировать все файловые системы, указанные в /etc/fstab. Однако это не полностью имитирует загрузку системы, так как некоторые сервисы могут быть недоступны.
7. Какая файловая система создаётся, если вы используете команду mkfs без какой-либо спецификации файловой системы? Команда mkfs сама по себе

не создает файловую систему без указания типа. Вам необходимо указать тип файловой системы (например, `mkfs.ext4`, `mkfs.vfat`).

8. Как форматировать раздел EXT4? Для форматирования раздела EXT4 используется команда `mkfs.ext4 /dev/`. Замените `/dev/` на путь к вашему устройству (например, `/dev/sda1`).
9. Как найти UUID для всех устройств на компьютере? Команда `blkid` отобразит UUID для всех доступных блочных устройств.

3 Выводы

Получила навыки создания разделов на диске и файловых систем. Получила навыки монтирования файловых систем.

Список литературы