番外篇 6 Linux文件系统

一、硬盘高级格式化

1.1 准备工作

1. 转换成root身份：

```
$ su      #如果没有设置过root密码的，先使用 sudo passwd root 设置root密码
```

2. 查看当前系统的磁盘数据：

```
$ fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6728fa32

Device     Boot Start      End  Sectors Size Id Type
/dev/sda1  *     2048 41943039 41940992  20G 83 Linux
```

3. 关闭虚拟机，在虚拟机控制台中增加一块硬盘，视频中以VMWare为例，增加了一块5G大小的硬盘。
4. 重启虚拟机，再利用fdisk -l命令观察数据变化。

```
Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

1.2 MBR分区

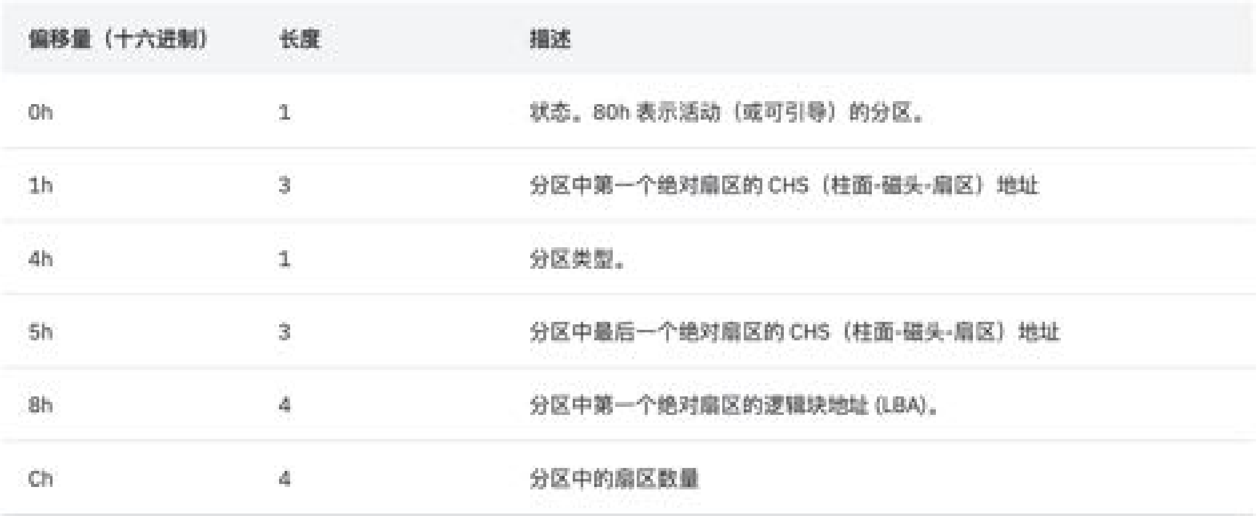
1. **维基参考链接**
2. 将 ***/dev/sdb*** 硬盘设备设置为MBR分区（disklabel type），并创建1个分区

```
$ fdisk /dev/sdb
Welcome to fdisk (util-linux 2.29.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help):  m
Help:

  Generic
```

```
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table


Command (m for help): o #设置该设备为MBR分区
Created a new DOS disklabel with disk identifier 0x87807b6a.

Command (m for help): p #打印出分区情况
Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x87807b6a

Command (m for help): n #增加一个分区
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): #以下全部采用了默认值，即将5G空间全部设置为primary主分区，分区号为1
Using default response p.
Partition number (1-4, default 1):
First sector (2048-10485759, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-10485759, default 10485759):

Created a new partition 1 of type 'Linux' and of size 5 GiB.

Command (m for help): p #重新输出sdb的分区情况
```

```
Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos #MBR分区
Disk identifier: 0x7f1f925f #设备ID号
#分区号      起始扇区号    终止扇区号   总扇区数   容量   分区类型编号   分区类型
Device      Boot Start       End  Sectors  Size   Id          Type
/dev/sdb1         2048 10485759 10483712   5G    83          Linux

Command (m for help): w #用该命令应用上述分区修改，如果放弃可以使用'q'命令
The partition table has been altered.
```

3. 分区的起始扇区号为2048，前面0~2047扇区为保留扇区，第0号扇区为MBR。
4. 观察/dev/sdb是否符合MBR的特征：主引导记录 (MBR) 是硬盘驱动器上的第一个扇区。MBR 包含引导程序代码（**440字节**），可能还包含其他一些信息，紧接着是 64 字节的分区表和一个 2 字节的引导签名。**64 字节的分区表**有 4 个 16 字节的条目，从偏移量 ***446 (1BEh)*** 开始。下表给出了每个 16 字节条目的布局。



偏移量（十六进制）	长度	描述
0h	1	状态。80h 表示活动（或可引导）的分区。
1h	3	分区中第一个绝对扇区的 CHS（柱面-磁头-扇区）地址
4h	1	分区类型。
5h	3	分区中最后一个绝对扇区的 CHS（柱面-磁头-扇区）地址
8h	4	分区中第一个绝对扇区的逻辑块地址 (LBA)。
Ch	4	分区中的扇区数量

```
#通过下面的命令，将MBR以16进制形式打印出来
$ dd if=/dev/sdb bs=512 count=1 2>/dev/null | hexdump -C
00000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
*                                                  #↓446字节的开始位置
000001b0  00 00 00 00 00 00 00 00  5f 92 1f 7f 00 00 00 20  |......._...... |
#               ↓           ↓         ↓           ↓
000001c0  21 00 83 b4 a8 8c 00 08  00 00 00 f8 9f 00 00 00  |!...............|
000001d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*                                                  #↓分区有效标志
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00000200
```

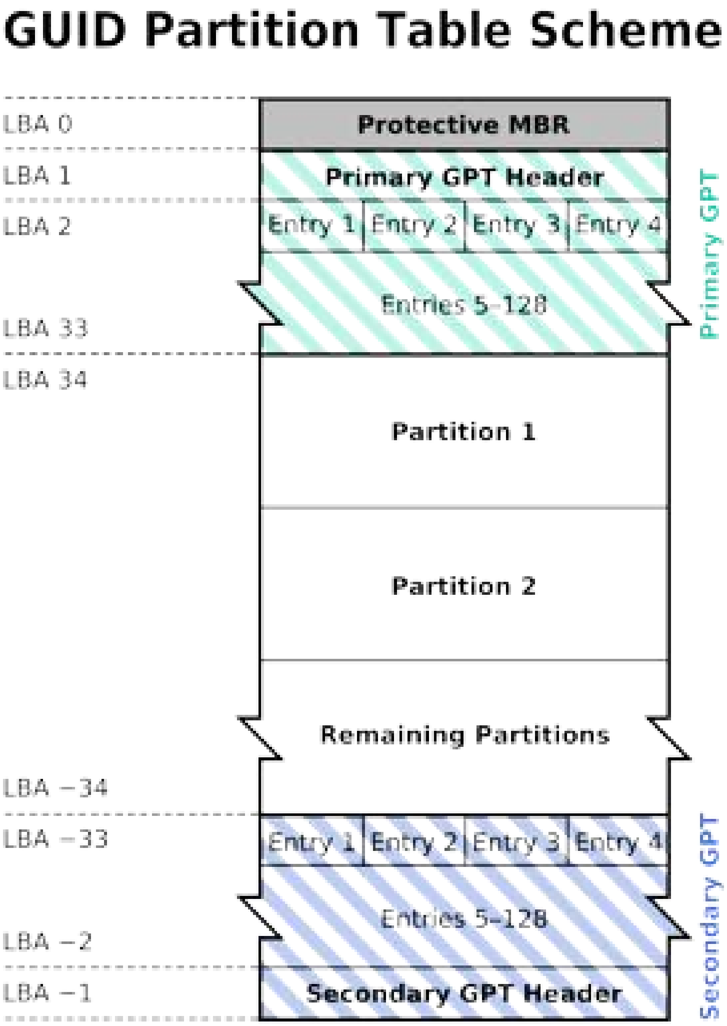
5. 使用下面的命令只显示64字节的分区表信息:

```
$ dd if=/dev/sdb bs=510 count=1 2>/dev/null | tail -c 64 | hexdump -C
00000000  00 20 21 00 83 b4 a8 8c  00 08 00 00 00 f8 9f 00  |. !.............|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000040
```

6. 扇区LBA号使用4个字节来编排，所以分区的最大扇区数为2^{32}个，扇区大小为512字节，则最大支持容量为$2^{32} * 512 = 2TBytes.$
7. 分区数量最多是4个，为了超过这个限制，这帮天才又想出来**扩展分区**这玩意，在此就不再展开，有兴趣的自行研究。

1.3 GPT分区

1. **维基参考链接**



2. 扇区按LBA模式编排，第0号扇区存放一个叫Protective MBR的数据结构，是为了兼容MBR分区，里面的分区类型为0xEE，不支持GPT的系统读到这个标志就会报错。
3. 每个分区信息占用128字节，因此1个扇区可以存放4个分区信息，从2~33号扇区都是存放分区信息的，故GPT可以支持最大128个分区。
4. 扇区号使用8字节进行编制，因此每个分区最大支持容量为：$2^{64} * 512 = 8ZBytes.$（注：TB→PB→EB→ZB→YB→BB→NB→DB）

5. 使用fdisk将/dev/sdb设备重新设置为GPT分区，并且按2：3的比例划分出2个分区

```
$ fdisk /dev/sdb

Welcome to fdisk (util-linux 2.29.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): m

Help:

  DOS (MBR)
   a   toggle a bootable flag
   b   edit nested BSD disklabel
   c   toggle the dos compatibility flag

  Generic
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   u   change display/entry units
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table


Command (m for help): g
Created a new GPT disklabel (GUID: 7CE8EAC3-AC64-4EC6-BE11-711A5AF46A05).

Command (m for help): p
```

```
Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 7CE8EAC3-AC64-4EC6-BE11-711A5AF46A05

Command (m for help): n
Partition number (1-128, default 1):
First sector (2048-10485726, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-10485726, default 10485726): +2G

Created a new partition 1 of type 'Linux filesystem' and of size 2 GiB.

Command (m for help): n
Partition number (2-128, default 2):
First sector (4196352-10485726, default 4196352):
Last sector, +sectors or +size{K,M,G,T,P} (4196352-10485726, default 10485726):

Created a new partition 2 of type 'Linux filesystem' and of size 3 GiB.

Command (m for help): p
Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 7CE8EAC3-AC64-4EC6-BE11-711A5AF46A05

Device       Start      End Sectors Size Type
/dev/sdb1     2048  4196351 4194304   2G Linux filesystem
/dev/sdb2  4196352 10485726 6289375   3G Linux filesystem

Command (m for help): m

Help:

  Generic
   d   delete a partition
   F   list free unpartitioned space
   l   list known partition types
   n   add a new partition
   p   print the partition table
   t   change a partition type
   v   verify the partition table
   i   print information about a partition

  Misc
   m   print this menu
   x   extra functionality (experts only)

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file

  Save & Exit
   w   write table to disk and exit
   q   quit without saving changes

  Create a new label
   g   create a new empty GPT partition table
   G   create a new empty SGI (IRIX) partition table
   o   create a new empty DOS partition table
   s   create a new empty Sun partition table


Command (m for help): t
Partition number (1,2, default 2): 2
Hex code (type L to list all codes): L
  1 EFI System                     C12A7328-F81F-11D2-BA4B-00A0C93EC93B
  2 MBR partition scheme           024DEE41-33E7-11D3-9D69-0008C781F39F
  3 Intel Fast Flash               D3BFE2DE-3DAF-11DF-BA40-E3A556D89593
  4 BIOS boot                      21686148-6449-6E6F-744E-656564454649
  5 Sony boot partition            F4019732-066E-4E12-8273-346C5641494F
  6 Lenovo boot partition          BFBFAFE7-A34F-448A-9A5B-6213EB736C22
  7 PowerPC PReP boot              9E1A2D38-C612-4316-AA26-8B49521E5A8B
  8 ONIE boot                      7412F7D5-A156-4B13-81DC-867174929325
  9 ONIE config                    D4E6E2CD-4469-46F3-B5CB-1BFF57AFC149
 10 Microsoft reserved             E3C9E316-0B5C-4DB8-817D-F92DF00215AE
 11 Microsoft basic data           EBD0A0A2-B9E5-4433-87C0-68B6B72699C7
 12 Microsoft LDM metadata         5808C8AA-7E8F-42E0-85D2-E1E90434CFB3
 13 Microsoft LDM data             AF9B60A0-1431-4F62-BC68-3311714A69AD
 14 Windows recovery environment   DE94BBA4-06D1-4D40-A16A-BFD50179D6AC
 15 IBM General Parallel Fs        37AFFC90-EF7D-4E96-91C3-2D7AE055B174
 16 Microsoft Storage Spaces       E75CAF8F-F680-4CEE-AFA3-B001E56EFC2D
 17 HP-UX data                     75894C1E-3AEB-11D3-B7C1-7B03A0000000
 18 HP-UX service                  E2A1E728-32E3-11D6-A682-7B03A0000000
 19 Linux swap                     0657FD6D-A4AB-43C4-84E5-0933C84B4F4F
 20 Linux filesystem               0FC63DAF-8483-4772-8E79-3D69D8477DE4
 21 Linux server data              3B8F8425-20E0-4F3B-907F-1A25A76F98E8
 22 Linux root (x86)               44479540-F297-41B2-9AF7-D131D5F0458A
 23 Linux root (ARM)               69DAD710-2CE4-4E3C-B16C-21A1D49ABED3
 24 Linux root (x86-64)            4F68BCE3-E8CD-4DB1-96E7-FBCAF984B709
 25 Linux root (ARM-64)            B921B045-1DF0-41C3-AF44-4C6F280D3FAE
 26 Linux root  (IA-64)             993D8D3D-F80E-4225-855A-9DAF8ED7EA97
 27 Linux reserved                 8DA63339-0007-60C0-C436-083AC8230908
 28 Linux home                     933AC7E1-2EB4-4F13-B844-0E14E2AEF915
 29 Linux RAID                     A19D880F-05FC-4D3B-A006-743F0F84911E
 30 Linux extended boot            BC13C2FF-59E6-4262-A352-B275FD6F7172
 31 Linux LVM                      E6D6D379-F507-44C2-A23C-238F2A3DF928
 32 FreeBSD data                   516E7CB4-6ECF-11D6-8FF8-00022D09712B
 33 FreeBSD boot                   83BD6B9D-7F41-11DC-BE0B-001560B84F0F
```

```

Hex code (type L to list all codes): 11

Changed type of partition 'Linux filesystem' to 'Microsoft basic data'.

Command (m for help): p
Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 7CE8EAC3-AC64-4EC6-BE11-711A5AF46A05

Device       Start      End Sectors Size Type
/dev/sdb1     2048  4196351 4194304   2G Linux filesystem
/dev/sdb2  4196352 10485726 6289375   3G Microsoft basic data

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

root@youngyt-PC:/home/youngyt# fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6728fa32

Device     Boot Start      End  Sectors Size Id Type
/dev/sda1  *     2048 41943039 41940992  20G 83 Linux


Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 7CE8EAC3-AC64-4EC6-BE11-711A5AF46A05

Device       Start      End Sectors Size Type
/dev/sdb1     2048  4196351 4194304   2G Linux filesystem
/dev/sdb2  4196352 10485726 6289375   3G Microsoft basic data
#Done
```

6. 用dd命令验证一下protective MBR扇区中的0xEE标志，若不支持GPT的系统读到这个标志会报错，支持GPT的话就知道这个分区用的是GPT而不是MBR。

```
root@youngyt-PC:/mnt/linux# dd if=/dev/sdb bs=510 count=1 2>/dev/null | tail -c 64 | hexdump -C
00000000  00 00 01 00 ee fe ff ff  01 00 00 00 ff ff 9f 00  |................|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000040
```

1.4 格式化

1. 任务：构建文件系统（高级格式化）
2. 命令mkfs

```
root@youngyt-PC:/home/youngyt# mkfs.ext4 /dev/sdb1
mke2fs 1.43.4 (31-Jan-2017)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: c5648df2-5619-48b7-b7e3-3d8fcf7dae91
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@youngyt-PC:/home/youngyt# mkfs.ntfs /dev/sdb2
Cluster size has been automatically set to 4096 bytes.
Initializing device with zeroes: 100% - Done.
Creating NTFS volume structures.
mkntfs completed successfully. Have a nice day.
root@youngyt-PC:/home/youngyt#
```

1.5 挂载分区

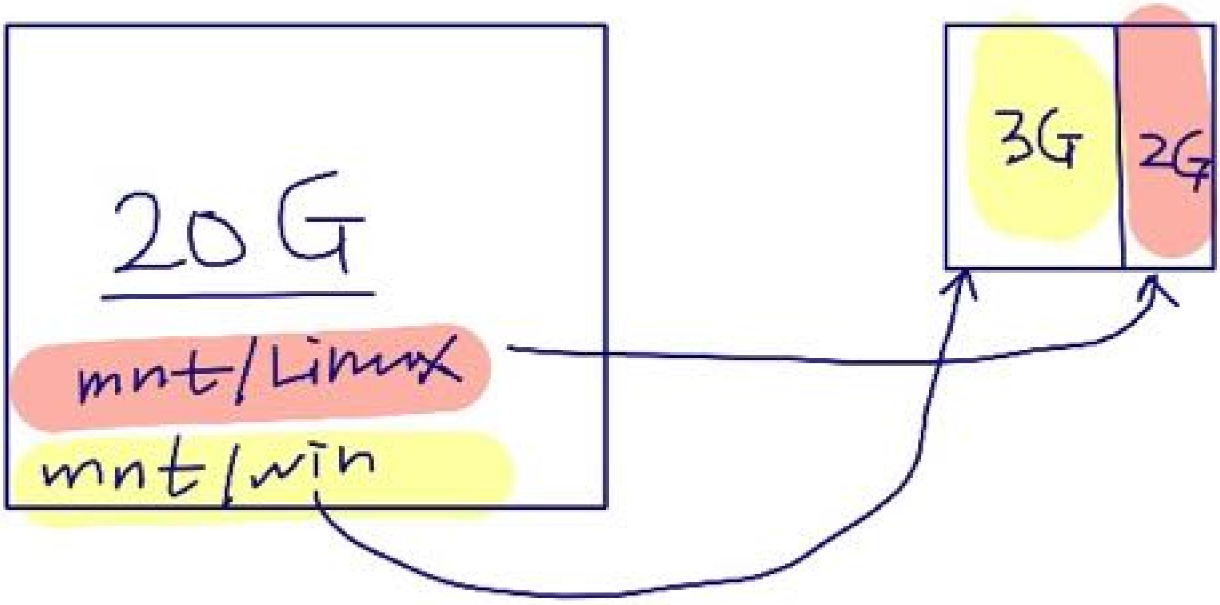
1. 目的：为了让分区可用
2. 命令：mount

```
root@youngyt-PC:/media# cd /mnt
root@youngyt-PC:/mnt# mkdir linux
root@youngyt-PC:/mnt# mkdir windows
root@youngyt-PC:/mnt# fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6728fa32

Device     Boot Start      End  Sectors Size Id Type
/dev/sda1  *     2048 41943039 41940992  20G 83 Linux


```

```
Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 7CE8EAC3-AC64-4EC6-BE11-711A5AF46A05

Device       Start      End Sectors Size Type
/dev/sdb1     2048  4196351 4194304   2G Linux filesystem
/dev/sdb2  4196352 10485726 6289375   3G Microsoft basic data
root@youngyt-PC:/mnt# mount /dev/sdb1 /mnt/linux/
root@youngyt-PC:/mnt# mount /dev/sdb2 /mnt/windows/
#Done
```

3. 查看分区挂载情况

```
root@youngyt-PC:/mnt/linux# lsblk -f
NAME   FSTYPE LABEL UUID                                 MOUNTPOINT
sda
└─sda1 ext4         c8993682-0699-4cbe-8688-58d73bbc49af /
sdb
├─sdb1 ext4         c5648df2-5619-48b7-b7e3-3d8fcf7dae91 /mnt/linux
└─sdb2 ntfs         24AEF58C71C9751B                     /mnt/windows
```



二、 Linux文件系统

1. 树形目录结构

```
root@youngyt-PC:/mnt/linux# tree
.
├── Lecture
|   ├── L01
|   └── L02
├── lost+found
└── test
```

2. 文件占用空间的大小

```
root@youngyt-PC:/mnt/linux# ls -l
total 24
drwxr-xr-x 4 root root  4096 Apr  7 11:26 Lecture
drwx------ 2 root root 16384 Apr  7 11:13 lost+found
-rw-r--r-- 1 root root    12 Apr  7 11:20 test

root@youngyt-PC:/mnt/linux# cat test
hello world

root@youngyt-PC:/mnt/linux# du -h test
4.0K    test
```

- 疑问：test文件大小只有12字节，但是占用了4K字节（8个扇区）的磁盘空间。
- Cluster：簇。文件系统是以簇为单位进行空间分配的。簇的大小是可以调节的。

3. 查看文件的目录项

```
root@youngyt-PC:/mnt/linux# stat test
  File: test
#     实际大小        占用的扇区数量         簇的大小
  Size: 12          Blocks: 8          IO Block: 4096   regular file
#                   Inode编号            链接数
Device: 811h/2065d  Inode: 12          Links: 1
#    ACL  (644是8进制)                  用户编号                     用户组编号
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
#      2进制     110 100 100
#      8进制      6   4   4
Access: 2020-04-07 11:25:04.477016832 +0800
Modify: 2020-04-07 11:20:47.387464472 +0800
Change: 2020-04-07 11:20:47.387464472 +0800
```

其中iNode存放了文件的物理扇区位置，因为一个文件大多数需要若干扇区（簇），为了保证目录大小的一致性，将这些占用的扇区编号统一保存在了一个iNode的结构当中，每个文件都有一个对应的iNode，每个iNode都有一个唯一的编号，所有的iNode都存放在分区开始部分的一个叫"superblock（超级块）"地方，可以用iNode编号在里面进行索引。

4. 使用chmod改变文件的ACL

```
root@youngyt-PC:/mnt/linux# chmod 464  test
root@youngyt-PC:/mnt/linux# ls -l
total 8
drwxr-xr-x 4 root root 4096 Apr  7 11:26 Lecture
-r--rw-r-- 1 root root   12 Apr  7 11:20 test
root@youngyt-PC:/mnt/linux# chmod 777 test
root@youngyt-PC:/mnt/linux# ls -l
total 8
drwxr-xr-x 4 root root 4096 Apr  7 11:26 Lecture
-rwxrwxrwx 1 root root   12 Apr  7 11:20 test
```

5. 使用debugfs命令观察文件的扇区内容

```
root@youngyt-PC:/mnt/linux# debugfs /dev/sdb1
debugfs 1.43.4 (31-Jan-2017)
debugfs:  help
Available debugfs requests:

show_debugfs_params, params
                         Show debugfs parameters
open_filesys, open       Open a filesystem
close_filesys, close     Close the filesystem
freefrag, e2freefrag     Report free space fragmentation
feature, features        Set/print superblock features
dirty_filesys, dirty     Mark the filesystem as dirty
init_filesys             Initialize a filesystem (DESTROYS DATA)
show_super_stats, stats  Show superblock statistics
ncheck                   Do inode->name translation
icheck                   Do block->inode translation
change_root_directory, chroot
                         Change root directory
change_working_directory, cd
                         Change working directory
list_directory, ls       List directory
show_inode_info, stat    Show inode information
dump_extents, extents, ex
                         Dump extents information
blocks                   Dump blocks used by an inode
filefrag                 Report fragmentation information for an inode
link, ln                 Create directory link
unlink                   Delete a directory link
mkdir                    Create a directory
rmdir                    Remove a directory
rm                       Remove a file (unlink and kill_file, if appropriate)
kill_file                Deallocate an inode and its blocks
copy_inode               Copy the inode structure
clri                     Clear an inode's contents
freei                    Clear an inode's in-use flag
seti                     Set an inode's in-use flag
testi                    Test an inode's in-use flag
freeb                    Clear a block's in-use flag
setb                     Set a block's in-use flag
```

```
testb                    Test a block's in-use flag
modify_inode, mi         Modify an inode by structure
find_free_block, ffb     Find free block(s)
find_free_inode, ffi     Find free inode(s)
print_working_directory, pwd
                         Print current working directory
expand_dir, expand       Expand directory
mknod                    Create a special file
list_deleted_inodes, lsdel
                         List deleted inodes
undelete, undel          Undelete file
write                    Copy a file from your native filesystem
dump_inode, dump         Dump an inode out to a file
cat                      Dump an inode out to stdout
lcd                      Change the current directory on your native filesyste
m
rdump                    Recursively dump a directory to the native filesystem
set_super_value, ssv     Set superblock value
set_inode_field, sif     Set inode field
set_block_group, set_bg  Set block group descriptor field
logdump                  Dump the contents of the journal
htree_dump, htree        Dump a hash-indexed directory
dx_hash, hash            Calculate the directory hash of a filename
dirsearch                Search a directory for a particular filename
bmap                     Calculate the logical->physical block mapping for an
inode
fallocate                Allocate uninitialized blocks to an inode
punch, truncate          Punch (or truncate) blocks from an inode by deallocat
ing them
symlink                  Create a symbolic link
imap                     Calculate the location of an inode
dump_unused              Dump unused blocks
set_current_time         Set current time to use when setting filesystem field
s
supported_features       Print features supported by this version of e2fsprogs
dump_mmp                 Dump MMP information
set_mmp_value, smmp      Set MMP value
extent_open, eo          Open inode for extent manipulation
zap_block, zap           Zap block: fill with 0, pattern, flip bits etc.
block_dump, bdump, bd    Dump contents of a block
ea_list                  List extended attributes of an inode
ea_get                   Get an extended attribute of an inode
ea_set                   Set an extended attribute of an inode
ea_rm                    Remove an extended attribute of an inode
list_quota, lq           List quota
get_quota, gq            Get quota
inode_dump, idump, id    Dump the inode structure in hex
journal_open, jo         Open the journal
journal_close, jc        Close the journal
journal_write, jw        Write a transaction to the journal
journal_run, jr          Recover the journal
help                     Display info on command or topic.
```

```
list_requests, lr, ?     List available commands.
quit, q                  Leave the subsystem.

debugfs:  blocks test  # 查看test文件占用的扇区号
33025
debugfs:  bdump 33025  # 将指定编号的扇区打印出来
0000  6865 6c6c 6f20 776f 726c 640a 0000 0000  hello world.....
0020  0000 0000 0000 0000 0000 0000 0000 0000  ................
*

debugfs:  q  #退出debugfs

#我们试着将test文件删除
root@youngyt-PC:/mnt/linux# rm test
root@youngyt-PC:/mnt/linux# ls -l
total 4
drwxr-xr-x 4 root root 4096 Apr  7 11:26 Lecture

#再次进入debugfs 查看之前的扇区内容
root@youngyt-PC:/mnt/linux# debugfs /dev/sdb1
debugfs 1.43.4 (31-Jan-2017)
debugfs:  bdump 33025    #发现文件虽然删除了，但是扇区中的文件内容还在，可以被用来反删除
0000  6865 6c6c 6f20 776f 726c 640a 0000 0000  hello world.....
0020  0000 0000 0000 0000 0000 0000 0000 0000  ................
*
#Done
```