第 5 章 文件系统基本管理

本章首先介绍磁盘分区和格式化的概念，以 ext2 文件系统为例分析文件系统的工作原理。接着以案例的形式介绍文件系统的管理——包括传统的 MBR 分区和 GPT 分区。最后简单介绍 LVM 的概念。

5.1 磁盘分区和格式化简介

5.1.1 磁盘分区的概念

磁盘必须经过分区，操作系统才能读取分区中的文件系统。目前在 Linux 系统下的磁盘分区主要有两种格式，分别是 MBR 分区与 GPT 分区。MBR 分区的历史较为悠久，不足之处在于分区有 2TB 容量的限制，而当前的磁盘容量已经超过 2TB，所以 MBR 分区就不适用了，GPT 分区应运而生，GPT 分区支持大容量分区。不过 GPT 分区并没有完全取代 MBR 分区。例如在虚拟机环境中，大部分磁盘的容量还是小于 2TB 的，因此 MBR 分区也还有存在的价值。

1. MBR 磁盘分区

在 MBR 分区表中，第 1 个扇区最重要，包含以下两部分：

（1）主引导记录（MBR）446 字节（Byte）。

（2）分区表（Partition Table）64 字节（Byte）。

注意：扇区（Sector）是磁盘中最小的物理存储单位，目前主要有 512B（字节）和 4KB 两种格式。

由于 MBR 分区表中仅有 64 字节用于分区表，因此默认分区表只能记录 4 项分区信息。这 4 项分区信息记录主分区（Primary）与扩展分区（Extend）。主分区被格式化后可以直接存取数据，但是扩展分区不能直接写入数据，仅相当于一个容器，必须从扩展分区中分出逻辑分区（logical），对逻辑分区格式化后才能存取数据。通常使用 P 代表主分区，使用 E 代表扩展分区，L 代表逻辑分区，这三者之间的关系总结如下：

（1）主分区与扩展分区最多可以有 4 个，其中扩展分区最多只能有一个。分区的数量可用数学公式表示，即 $0<N_P+N_E\leqslant 4$，且 $N_E=0$ 或者 1。

（2）逻辑分区是在扩展分区上分出的分区。

（3）主分区和逻辑分区都可以被格式化后用作数据存取。扩展分区无法格式化。

（4）逻辑分区的数量依操作系统而不同。

2. GPT 磁盘分区

1）什么是 GPT

GPT 是全局唯一标识磁盘分区表，其全称是 GUID Partition Table，是源自 EFI 标准的一种全新磁盘分区表标准结构，与普通的 MBR 分区相比，更加灵活，更加强大。

常见的磁盘扇区有 512B 和 4KB 两种容量。为了解决兼容性问题，通常使用逻辑区块地址（Logical Block Address，LBA）来处理扇区的定义。GPT 将磁盘所有区块以 LBA（默认为 512B）来规划，第 1 个 LBA 称为 LBA0。

与 MBR 只使用第 1 个 512B 区块记录分区信息不同，GPT 使用了多个 LBA 区块记录分区信息。LBA2~LBA33 实际记录分区表，每个 LBA 记录 4 项数据，所以一共可以记录 32×4=128 项以上的分区信息。因为每个 LBA 为 512B，所以每条记录可占用 512/4=128B，因为每条记录主要记录开始和结束两个扇区的位置，因此记录的扇区位置最多可达 64 位（bit），若每个扇区容量是 512B，则单个分区的最大容量是 8ZB（1ZB=2^{30}TB）。另外，分区区域结束后就是分区表备份，分区表备份是对分区表 32 个扇区的完整备份。

值得一提的是，每个 GPT 分区都属于主分区，可以直接进行格式化。

2）GPT 分区的优点

（1）分区数量无限制，但是受到 Windows 系统的限制，最多只允许创建 128 个分区。一般够用了。

（2）分区容量支持超过 2TB 的硬盘。GPT 采用 64 位的整数表示扇区号，支持的容量非常大，而传统的 MBR 最高支持到 2TB。

（3）分区表有容灾功能，如果分区表坏了，系统则将自动读取分区表备份保证正常识别分区。

3）什么时候使用 GPT 分区

（1）如果硬盘容量大于 2TB，则无论是数据盘还是系统盘，直接选用 GPT 分区。

（2）若新买的计算机主板是集成 UEFI BIOS，GPT 分区和 UEFI 是一对好搭档，建议直接选用 GPT 分区。

5.1.2 格式化的概念

磁盘经过分区后，下一个步骤就是要对磁盘分区进行格式化。格式化是指根据用户选定的文件系统类型（如 ext4 或 xsf），在磁盘的特定区域写入特定数据，在分区中划出一片用于文件管理的磁盘空间，也就是说，将 inode 和 block 规划好。格式化的目的就是为了写入文件系统，但是需要注意，格式化操作会导致现有分区中的所有数据被清除，因此要慎重。

格式化的命令为 mkfs，在后面章节将详细介绍该命令。

5.1.3　Linux 的 ext2 文件系统简介[①]

ext2 是 Linux 早期比较流行的文件系统，很多文件系统的设计源自它。只要掌握了 ext2 文件系统，其他文件系统大同小异。

在 ext2 文件系统中，需要了解以下几个概念。

（1）block：即区块，实际记录文件的内容，如果文件比较大，则会占用多个 block。

（2）inode：即索引节点，记录文件的属性。一个文件占用一个 inode，并记录此文件数据所在的 block 号码。

（3）Super Block：又叫超级区块，记录文件系统的整体信息，包括 inode/block 的总量、已经使用的量、剩余的量，以及文件系统的格式和一些相关信息。

1. ext2 文件系统布局

ext2 文件系统的布局如图 5-1 所示。

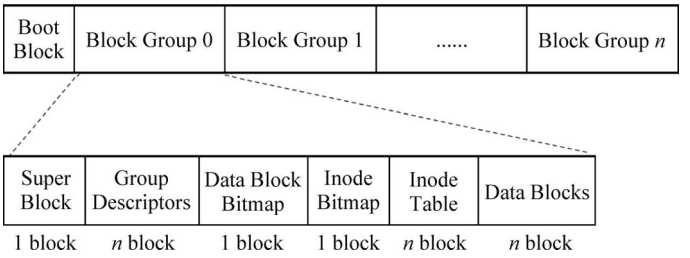

图 5-1　ext2 文件系统示意图

1）Boot Block（引导块）

每个硬盘分区的开头 1024 字节，即 0Byte 到 1023Byte 是分区的启动扇区。存放由 ROM BIOS 自动读入的引导程序和数据，但这只对引导设备有效，而对于非引导设备，该引导块不含代码。这个块与 ext2 没有任何关系。如果这个块损坏，则整个文件系统也就启动不起来了。

2）Super Block（超级块）

Super Block 是整个文件系统的综合概要信息。超级块在每个块组的开头都有一份备份。要读取文件系统一定要从 Super Block 读起。Super Block 主要记录的数据如下：

（1）block 与 inode 的总量。

（2）已经使用或者还尚未使用的 inode/block 数量。

（3）block 与 inode 的大小（block 可以是 1KB、2KB、4KB，inode 可以是 128B 或 256B）。

（4）文件系统的挂载时间、最近一次写入数据的时间等相关信息。

（5）一个 Valid Bit（有效位）。Valid Bit 标记该文件系统是否已经被挂载，若 Valid Bit

① 关于 ext2 文件系统的部分内容参考了《鸟哥的 Linux 基础学习实训教程》。

值为 0，则表示已挂载；若 Valid Bit 值为 1，则表示没有挂载。

3）Group Descriptors（块组描述符）

整个分区分成多少个块组就对应有多少个块组描述符。每个块组描述符存储一个块组的描述信息，包括 inode 表从哪里开始，数据块从哪里开始，空闲的 inode 和数据块还有多少个等。块组描述符表在每个块组的开头也都有一份备份，这些信息是非常重要的，因此它们都有多份备份。

4）Data Block Bitmap（块位图）

Data Block Bitmap 是用来描述整个块组中哪些块已用及哪些块空闲的，本身占一个块，其中的每个 bit 代表本块组中的一个块，如果这个 bit 为 1，则表示该块已用，如果这个 bit 为 0，则表示该块空闲可用。假设块大小为 b 字节，则可以区别的块数为 b×8 个。

5）Inode Bitmap（inode 索引节点位图）

和 Data Block Bitmap 类似，本身占一个块，其中每个 bit 表示一个 inode 是否空闲可用。

6）Inode Table（inode 索引节点表）

Inode Table 由若干个 inode 数据结构组成，需要占用若干个块。每个 inode 即对应一个文件或目录。inode 的内容是对除文件名（目录名）外的所有属性的描述，包括文件类型（普通文件、目录、符号链接等）、权限、文件创建时间/修改时间/访问时间、文件所占数据块的个数及指向数据块的指针。

7）Data Block（数据块）

原则上，Data Block 的大小与数量在格式化完成后就不能再修改了；如果需要修改，则必须重新进行格式化。

每个 block 内最多只能存放一个文件的数据。如果文件较大，则一个文件会占用多个 block。如果文件的大小小于 block，则该 block 的剩余容量也不能被其他文件使用了。这样方便管理，但也一定程度上使磁盘空间浪费了。

2. ext2 文件系统原理

对于一个文件来讲，除了它的数据保存在数据块中，其他信息都保存在自己的 inode 表中，但需要注意的是，inode 表和数据块都不保存文件的名字，它的名字保存在目录的数据块中（当然目录也有自己的 inode 表）。

对于目录来讲，目录的数据块中存储的就是文件名及其这些文件名的 inode 地址。

其他文件系统会通过虚拟文件系统的方法，演绎出和 ext2 相似的做法。

1）普通文件相关操作

一般而言，当读取文件系统中的一个文件时，大致流程如下：

（1）读出文件的 inode 编号。

（2）根据 inode 内的权限判定用户能否存取该文件。

（3）若用户有足够的权限，则开始读取 inode 内所记录的数据存放在哪些编号的 block 中。

（4）读出这些编号 block 内的数据，将这些数据按逻辑次序组合起来即为一个文件的实

际内容。

新建文件的大致流程如下：

（1）当需要新建一个文件时，先到 metadata 区块找到尚未使用的一个 inode。

（2）将权限和属性相关数据写入该 inode，并在 metadata 区块将该 inode 设置为已使用，同时更新 Super Block 信息。

（3）到 metadata 区块找到尚未使用的 block，将实际数据写入 block，若文件内容较多，则继续在 metadata 区块找尚未使用的 block，持续写入，直到写完数据为止。

删除文件的大致流程如下：

（1）将要删除文件的 inode 编号与所属相关的 block 编号抹除取消。

（2）将 metadata 区块相对应的 inode 与 block 设置为尚未使用，并同步更新 Super Block 的数据。

2）目录相关操作

当用户创建一个目录时，文件系统会分配一个 inode 与对应的至少一块 block 给该目录，其中，inode 用于记录该目录的权限与属性，并记录分配到的 block 的编号；block 则记录在这个目录下的文件的文件名和这些文件对应的 inode 编号。

前面提到，读取文件数据时，最重要的一步是要先读到该文件的 inode 编号，而我们在实际操作时，是通过“文件名”来读写数据的，并没有直接用 inode 编号，因此，目录的重要意义就在于此——记录文件名与该文件名对应的 inode 编号。

5.2　文件系统的管理

某同学在使用虚拟机做 openEuler 系统实验的过程中发现硬盘容量不足，他考虑增加新硬盘，但是自己又刚接触 openEuler 系统，所以不知如何下手。下面为这位同学提供解决问题的思路。

（1）首先，为名为 openEuler-server 的 openEuler 虚拟机增加一块新硬盘，硬盘大小为 20GB，如图 5-2 所示，选择【编辑虚拟机设置】或者双击【设备】列表中的任一设备打开虚拟机设置对话框。

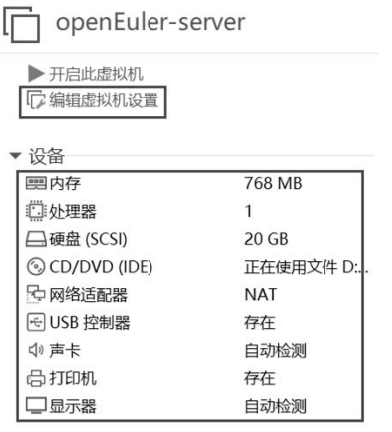

图 5-2　编辑虚拟机设置

（2）在虚拟机设置对话框中，如图 5-3 所示，单击【添加（A）】按钮，接着依次按照如图 5-4~图 5-9 所示完成新硬盘的设置。

注意：若指定的虚拟磁盘大小超过硬盘上实际可用磁盘空间，则会出错，从而导致创建磁盘失败。

在图 5-8 中，指定磁盘文件使用默认文件名即可，然后单击【完成】按钮就完成了新硬盘的创建。磁盘创建成功后即可出现在虚拟机的主页上，如图 5-9 所示。

图 5-3　虚拟机设置对话框

图 5-4　添加硬盘

图 5-5　选择硬盘类型

图 5-6　创建新虚拟磁盘

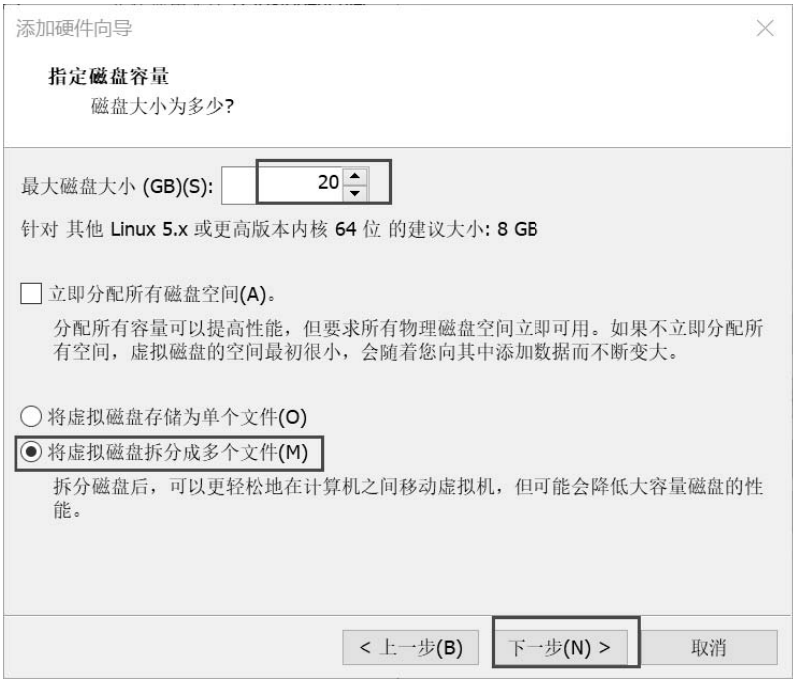

图 5-7　设置磁盘容量

图 5-8　指定磁盘文件

图 5-9　新硬盘出现在虚拟机主页

（3）重复步骤（2）中的操作，再创建一个大小为 20GB 的新硬盘。

（4）重启虚拟机。以上两块新硬盘将分别被系统识别为/dev/sdb 和/dev/sdc。

（5）/dev/sdb 和/dev/sdc 分别使用了不同的分区方式。具体规划如下：/dev/sdb 使用 MBR 分区，共划分为 5 个分区，容量分别为 2GB、2GB、5GB、5GB、6GB；/dev/sdc 使用 GPT 分区，共分为 5 个分区，容量分别为 2GB、2GB、5GB、5GB、6GB。

（6）以普通用户 chen 身份登录系统，启动虚拟终端，切换至 root 用户，工作目录为 root 的家目录。

5.2.1　MBR 分区管理

1. 创建磁盘分区

使用 fdisk 命令可以对磁盘进行 MBR 分区操作，包括增加分区、删除分区、查看分区，以及转换分区类型等。命令语法如下：

```
fdisk [选项] [设备]
```

fdisk 命令的常用选项及含义见表 5-1。

表 5-1　fdisk 命令的常用选项及含义

选　项	含　义
-b	显示扇区计数和大小
-l	列出指定磁盘的分区表信息
-s	显示分区大小，单位为块
-v	显示命令版本信息

fdisk 命令采用传统的问答式界面，需要用户输入子命令，fdisk 命令的常用子命令的功能见表 5-2。

表 5-2 fdisk 命令的常用交互子命令

子命令	功 能
m	显示 fdisk 的子命令
n	创建一个新分区，通常接下来系统会询问创建新分区类型：p 表示主分区；e 表示扩展分区；l 表示逻辑分区
d	删除磁盘分区
T	更改分区类型
v	检查校验分区表
p	打印分区表，显示磁盘分区信息
q	退出 fdisk 而不保存磁盘分区配置
w	保存磁盘分区配置并退出 fdisk

注意：为了方便读者理解，本节对 fdisk 命令的使用进行分步骤讲解，但实际上这是一个连贯的 fdisk 命令。如果中途退出 fdisk 命令，则前面的操作可能无效。

使用 fdisk 命令为/dev/sdb 创建磁盘分区。下面 fdisk 命令代码中的<--表示后面为用户输入，起提示作用，实际中不存在。

（1）进入 fdisk 界面，显示磁盘分区信息。

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

欢迎使用 fdisk (util-Linux 2.36.1)。
更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

设备不包含可识别的分区表。
创建了一个磁盘标识符为 0x62b675ce 的新 DOS 磁盘标签。

命令(输入 m 获取帮助)：<--m

帮助：

  DOS (MBR)
   a   开关可启动标志
   b   编辑嵌套的 BSD 磁盘标签
   c   开关 DOS 兼容性标志

常规
   d   删除分区
```