



# 目录页

Part 1

Part 2

Part 3

Part 4

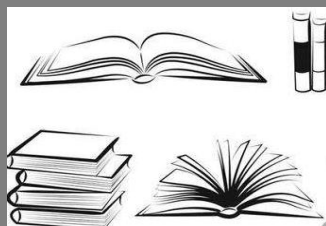
Linux命令格式



用户与用户组  
管理



文件管理



存储管理



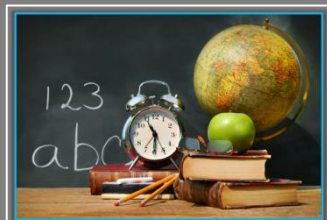
Part 5

Part 6

Part 7

Part 8

进程管理



服务管理



软件包管理



文本编辑器





## 2.4存储管理

Linux系统中磁盘的名称由系统根据设备类型自动识别，常用的存储设备类型有IDE、SATA、USB、SCSI等，其中IDE设备在Linux系统中被识别为hd，SATA、USB、SCSI设备在Linux系统中被识别为sd。





## 2.4存储管理

若系统中使用了多个同类型的设备，这些设备按照添加的顺序，使用小写字母依次编号。如系统中有两个sd设备，则第一个设备名为sda，第二个设备名为sdb，以此类推。





## 2.4.1 磁盘分区



磁盘分区有利于数据的分类存储，管理员可根据磁盘中将要存放的文件类型、数量和文件大小等因素，合理规划磁盘空间，以**提高**磁盘**使用率**与读取**速率**。常见的磁盘分区方式为MBR分区和GPT分区。



## 2.4.1 磁盘分区



### 1、MBR分区

MBR全称Master Boot Record，即主引导记录。磁盘中的空间以扇区为单位，采用MBR方式分区的磁盘第一个扇区中包含一个64字节的磁盘分区表，每个分区信息占用16字节，因此分区表最多可存储4项分区信息，也就是说磁盘只能划分出4个主分区。

磁盘 sda	sda1	sda2	sda3	sda4	剩余空间
--------	------	------	------	------	------

即便4个分区容量总和小于磁盘总容量，也无法再为剩余空间分区。



## 2.4.1 磁盘分区



### 1、MBR分区

但事实上，我们知道Windows中可以有不止4个分区，这是因为，MBR允许在基础分区中设置一个扩展分区，而扩展分区又可以划分为多个逻辑分区。

磁盘 sda	sda1	sda2	扩展分区 sda3		
			逻辑分区 sda5	逻辑分区 sda6	.....

- 在MBR分区中，编号1-4被预留给基础分区，所以逻辑分区的编号一定从5开始（即便基础分区数量不足4个）。
- 扩展分区也有自己的磁盘分区表，扩展分区的磁盘分区表存储于扩展分区的第一个扇区中。



## 2.4.1 磁盘分区



### 1、MBR分区

使用MBR方式创建的分区，可通过fdisk命令进行管理。

fdisk命令可以查看当前系统中的磁盘的分区情况，该命令的格式如下：

**fdisk [选项] [磁盘]**

选项	说明
-l	详细显示磁盘及其分区信息
-s	显示磁盘分区容量（单位为block）
-b	设置扇区大小（扇区大小取值为512、1024、2048或4096，单位MB）



## 2.4.1 磁盘分区



### 1、MBR分区

fdisk示例:

```
# fdisk -l /dev/sda
```

```
# fdisk -s /dev/sda
```

```
# fdisk -b 512 /dev/sda
```

以上示例只能实现磁盘与分区的简单管理，若要使用fdisk命令创建磁盘分区，需要在终端输入“fdisk 磁盘”命令进入fdisk的交互界面。





## 2.4.1 磁盘分区



### 1、MBR分区——fdisk交互使用

在终端输入 “**fdisk 磁盘**” ， 将会

进入如下所示的界面。

```
[root@localhost ~]# fdisk /dev/sda
欢迎使用 fdisk (util-linux 2.23.2)。
```

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。

命令(输入 m 获取帮助)： █

输入 “**m**” 查看fdisk命令可实现的操作。

命令(输入 m 获取帮助)： m  
命令操作

```
a  toggle a bootable flag
b  edit bsd disklabel
c  toggle the dos compatibility flag
d  delete a partition
g  create a new empty GPT partition table
G  create an IRIX (SGI) partition table
l  list known partition types
m  print this menu
n  add a new partition
o  create a new empty DOS partition table
p  print the partition table
q  quit without saving changes
s  create a new empty Sun disklabel
t  change a partition's system id
u  change display/entry units
v  verify the partition table
w  write table to disk and exit
x  extra functionality (experts only)
```



## 2.4.1 磁盘分区



### 1、MBR分区——fdisk交互使用

在终端输入 “**fdisk 磁盘**” ， 将会

进入如下所示的界面。

```
[root@localhost ~]# fdisk /dev/sda
欢迎使用 fdisk (util-linux 2.23.2)。
```

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。

命令(输入 m 获取帮助)： █

输入 “**m**” 查看fdisk命令可实现的操作。

命令(输入 m 获取帮助)： m  
命令操作

```
a  toggle a bootable flag
b  edit bsd disklabel
c  toggle the dos compatibility flag
d  delete a partition
g  create a new empty GPT partition table
G  create an IRIX (SGI) partition table
l  list known partition types
m  print this menu
n  add a new partition
o  create a new empty DOS partition table
p  print the partition table
q  quit without saving changes
s  create a new empty Sun disklabel
t  change a partition's system id
u  change display/entry units
v  verify the partition table
w  write table to disk and exit
x  extra functionality (experts only)
```



## 2.4.1 磁盘分区



### 1、MBR分区——查看磁盘使用情况

以/dev/sda磁盘为例展示fdisk命令的交互用法。

```
[root@localhost ~]# fdisk -l /dev/sda
```

查看磁盘使用情况

磁盘 /dev/sda: 21.5 GB, 21474836480 字节, 41943040 个扇区

Units = 扇区 of 1 \* 512 = 512 bytes

扇区大小(逻辑/物理): 512 字节 / 512 字节

I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘标签类型: dos

磁盘标识符: 0x000b1493

设备	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	616447	307200	83	Linux
/dev/sda2		616448	28944383	14163968	8e	Linux LVM



## 2.4.1 磁盘分区



### 1、MBR分区——创建新分区

#### 在磁盘/dev/sda上创建新分区

命令(输入 m 获取帮助): n

创建分区

Partition type:

p primary (2 primary, 0 extended, 2 free)

e extended

创建扩展分区

Select (default p): e

分区号 (3,4, 默认 3): 3

设置扩展分区编号

起始 扇区 (28944384-41943039, 默认为 28944384):

将使用默认值 28944384

Last 扇区, +扇区 or +size{K,M,G} (28944384-41943039, 默认为 41943039): +2G

分区 3 已设置为 Extended 类型, 大小设为 2 GiB

设置扩展分区大小



## 2.4.1 磁盘分区



### 1、MBR分区——创建新分区

#### 在磁盘/dev/sda上创建逻辑分区

命令(输入 m 获取帮助): n

创建分区

Partition type:

p primary (2 primary, 1 extended, 1 free)

l logical (numbered from 5)

创建逻辑分区

Select (default p): l

添加逻辑分区 5

设置逻辑分区编号

起始 扇区 (28946432-33138687, 默认为 28946432):

将使用默认值 28946432

Last 扇区, +扇区 or +size{K,M,G} (28946432-33138687, 默认为 33138687): +1G

分区 5 已设置为 Linux 类型, 大小设为 1 GiB

设置逻辑分区大小



## 2.4.1 磁盘分区



### 1、MBR分区——查看分区表

#### 查看磁盘/dev/sda此时的分区表

命令(输入 m 获取帮助): p

磁盘 /dev/sda: 21.5 GB, 21474836480 字节, 41943040 个扇区

Units = 扇区 of 1 \* 512 = 512 bytes

扇区大小(逻辑/物理): 512 字节 / 512 字节

I/O 大小(最小/最佳): 512 字节 / 512 字节

磁盘标签类型: dos

磁盘标识符: 0x000b1493

设备	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	616447	307200	83	Linux
/dev/sda2		616448	28944383	14163968	8e	Linux LVM
/dev/sda3		28944384	33138687	2097152	5	Extended
/dev/sda5		28946432	31043583	1048576	83	Linux

新扩展分区

新逻辑分区



## 2.4.1 磁盘分区



### 2、GPT分区

GPT (GUID Partition Table, 全局唯一标识分区表) 是一种较新的分区方式, 这种分区方式解决了MBR的很多缺点, 他**支持**超过**2TB**的磁盘, 向后**兼容MBR**。Windows7、Windows8系统下若想要使用GPT方式为数据盘分区, 可直接进行转换, 但若想要使用GPT方式为系统盘分区, 则必须采用支持UEFI的主板。此外, GPT只支持64位操作系统。



## 2.4.1 磁盘分区



### 2、GPT分区

在Linux系统中可用parted创建GPT格式的分区。parted是GNU组织开发的一款强大的磁盘管理工具，与fdisk不同，它既能为磁盘分区，亦能调整分区大小。parted也有命令行和交互这两种模式，命令行模式下其格式如下所示：

**parted** [选项] **设备** [命令]

(1) 修改分区表类型

使用parted工具的mklabel命令可以修改磁盘分区表格式，语法格式如下：

**parted** 磁盘 **mklabel** gpt





## 2.4.1 磁盘分区



### 2、GPT分区

在Linux系统中可用parted创建GPT格式的分区。parted是GNU组织开发的一款强大的磁盘管理工具，与fdisk不同，它既能为磁盘分区，亦能调整分区大小。parted也有命令行和交互这两种模式，命令行模式下其格式如下所示：

**parted** [选项] **设备** [命令]

#### (2) 查看分区表信息

使用parted工具的print命令可以查看磁盘的分区表信息，语法格式如下：

**parted** 磁盘 **print**



## 2.4.1 磁盘分区



### 2、GPT分区

在Linux系统中可用parted创建GPT格式的分区。parted是GNU组织开发的一款强大的磁盘管理工具，与fdisk不同，它既能为磁盘分区，亦能调整分区大小。parted也有命令行和交互这两种模式，命令行模式下其格式如下所示：

**parted** [选项] **设备** [命令]

(3) 创建分区

使用parted工具的mkpart命令可以创建分区，语法格式如下：

**parted** 磁盘 **mkpart** 分区类型 文件系统 起始 结束

primary

Logical

extended

fat16

fat32

ext2

linux-swap

reiserfs

示例： **parted /dev/sdc mkpart ext2 1 2G**



## 2.4.1 磁盘分区



### 2、GPT分区

在Linux系统中可用parted创建GPT格式的分区。parted是GNU组织开发的一款强大的磁盘管理工具，与fdisk不同，它既能为磁盘分区，亦能调整分区大小。parted也有命令行和交互这两种模式，命令行模式下其格式如下所示：

**parted** [选项] **设备** [命令]

#### (4) 删除分区

使用parted的rm命令可以删除分区，语法格式如下：

parted 磁盘 **rm** 分区编号

**示例：** parted /dev/sdc rm 2



## 2.4.1 磁盘分区



### 2、GPT分区

在Linux系统中可用parted创建GPT格式的分区。parted是GNU组织开发的一款强大的磁盘管理工具，与fdisk不同，它既能为磁盘分区，亦能调整分区大小。parted也有命令行和交互这两种模式，命令行模式下其格式如下所示：

**parted** [选项] **设备** [命令]      **示例：**

(5) 分区复制

```
parted /dev/sdc cp /dev/sdb sdb3 sdc1
```

使用parted工具的cp命令可将一个设备上的指定分区复制到当前设备的指定分区，语法格式如下：

```
parted 磁盘 cp 源设备 源分区 目标分区
```



## 2.4.2 格式化

磁盘给待存储的数据以硬件支持，但磁盘本身并不规定文件的存储方式，因此在使用磁盘之前，还需规定文件在磁盘中的组织方式，即格式化磁盘，为磁盘创建文件系统。

Linux系统中使用mkfs命令实现格式化分区的功能，该命令的格式如下：

**mkfs** [选项] [参数] 分区





## 2.4.2 格式化

mkfs命令的用法示例如下：

#将扩展分区/dev/sda5格式化，设置其文件系统为ext2

# mkfs -t ext2 /dev/sda5

也可使用“mkfs.文件系统”的方式格式化分区，示例如下：

#将扩展分区/dev/sda5格式化，设置其文件系统为ext2

# mkfs.ext2 /dev/sda5



## 2.4.2 格式化

1、mkfs命令的用法示例如下：

#将扩展分区/dev/sda5格式化，设置其文件系统为ext2

**# mkfs -t ext2 /dev/sda5**

2、也可使用“mkfs.文件系统”的方式格式化分区，示例如下：

#将扩展分区/dev/sda5格式化，设置其文件系统为ext2

**# mkfs.ext2 /dev/sda5**

3、也可使用“mkfs.文件系统”的方式格式化分区，示例如下：

#将扩展分区/dev/sda5格式化，设置其文件系统为ext2

**# mkfs.ext2 /dev/sda5**



## 2.4.3 挂载



挂载，是指将一个目录作为入口，把磁盘分区中的数据放置在以该目录为根结点的目录关系树中，这相当于为文件系统与磁盘进行了链接，指定了某个分区中文件系统访问的位置。





## 2.4.3 挂载



Linux系统中根目录是整个磁盘访问的基点，因此根目录必须要挂载到某个分区。Linux系统中通过 **mount** 命令和 **unmount** 命令实现分区的挂载和卸载。



## 2.4.3挂载



### 1、挂载

Linux系统中可以使用mount命令将某个分区挂载到目录，mount命令常用的格式如下：

**mount** [选项] [参数] 设备 挂载点

mount命令常用的选项有两个，分别为-t和-o，下面分别介绍这两个选项的功能。



## 2.4.3挂载



### 1、挂载

(1) -t

选项-t用于指定待挂载设备的文件系统类型，常见的类型如下：

- ❑ ISO9660：光盘/光盘镜像；
- ❑ MS-DOS：DOS fat16文件系统；
- ❑ VFAT：Windows 9x fat32文件系统；
- ❑ NTFS：Windows NT ntfs文件系统；
- ❑ SMBFS：Mount Windows文件网络共享；
- ❑ NFS：UNIX(LINUX)文件网络共享。



## 2.4.3挂载



### 1、挂载

(2) -o

选项-o主要用来描述设备的挂载方式，常用的挂载方式如表所示。

方式	说明
loop	将一个文件视为硬盘分区挂载到系统
ro	read-only，采用只读的方式挂载设备（即系统只能对设备进行读操作）
rw	采用读写的方式挂载设备
iocharset	指定访问文件系统所用的字符集
remount	重新挂载



## 2.4.3挂载



### 1、挂载

- mount的**参数**通常为设备文件名与挂载点。
- **设备**文件名即为要挂载的文件系统对应的设备名；
- **挂载点**指挂载点目录，设备必须被挂载到一个已经存在的目录上，其中的内容能通过目录访问，挂载的目录可以不为空，但将某个设备挂载到该目录后，目录中之前存放的内容不可再用。



## 2.4.3挂载



### 1、挂载——示例

#### (1) 挂载移动硬盘

- 确认设备连接到主机
- 使用挂载命令： `mount /dev/sdb5 /mnt/dir1`

#### (2) 挂载镜像文件

- 解压压缩包
- 使用挂载命令： `mount -o rw -t iso9660 /usr/test.iso /home/itheima`



## 2.4.3挂载



## 2、卸载

当需要挂载的分区只是一个移动存储设备（如移动硬盘）时，要进行的工作是在该设备与主机之间进行文件传输，那么在文件传输完毕之后，需要卸载该分区。Linux系统中卸载分区的命令是**umount**，该命令的格式如下：

`umount [选项] 参数`

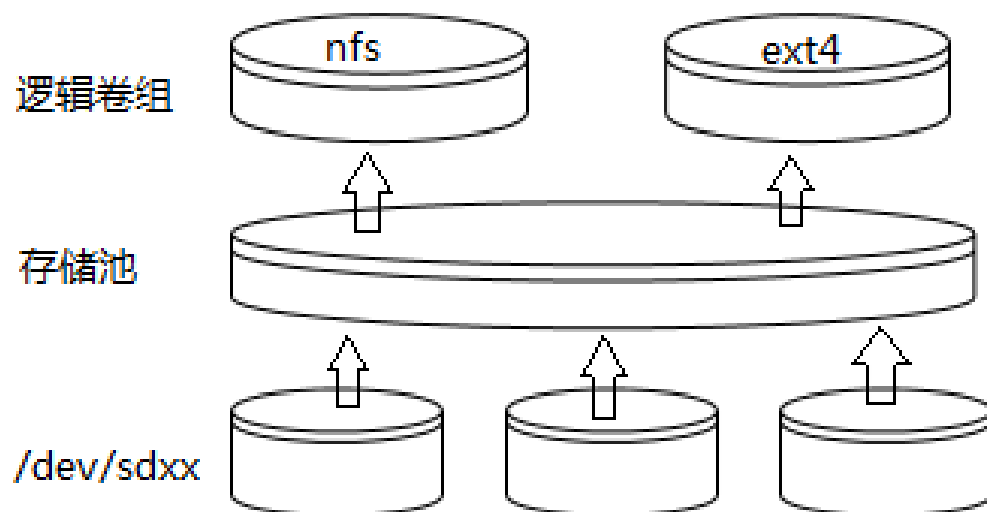
**示例：**

`umount /mnt`



## 2.4.4 LVM逻辑卷管理

LVM是Linux系统**管理磁盘**分区的一种**机制**，它建立在磁盘和分区之上，可以帮助管理员**动态地**管理磁盘，提高磁盘分区管理的灵活性。



LVM允许管理员调整存储卷组的大小，并按逻辑卷组进行命名、管理和分配。当系统中添加新的磁盘后，管理员不必将原有的文件移动到新的磁盘，只需直接扩展文件系统容量，即可使用新的磁盘。





## 2.4.4 LVM逻辑卷管理

下面对实现LVM机制常用的命令有：

- pvcreate
- vgcreate
- lvcreate
- vgdisplay
- lvextend
- lvremove



## 2.4.4 LVM逻辑卷管理



### 1、pvcreate

pvcreate命令用于将磁盘分区初始化为物理卷，该命令的语法格式如下：

pvcreate [选项] 参数

pvcreate命令的参数通常为磁盘分区名，常用选项如表所示：

选项	说明
-f	强制创建物理卷，不需要用户确认
-u	指定设备的UUID（Universally Unique Identifier，通用唯一识别码）
-y	所有的问题都用“yes”回答
-z	是否使用前4个扇区（y/n）

示例：pvcreate /dev/sdc2



## 2.4.4 LVM逻辑卷管理



### 2、vgcreate

vgcreate命令用于将物理卷整合为卷组，该命令的语法格式如下：

vgcreate [选项] 卷组名称 物理卷路径1 物理卷路径2 ...

vgcreate命令的常用选项如表2-28所示：

选项	说明
-l	设置卷组上允许创建的最大逻辑卷数
-p	设置卷组上允许添加的最大物理卷数
-s	设置卷组上的最小存储单元（PE）



## 2.4.4 LVM逻辑卷管理



### 3、lvcreate

lvcreate命令的功能是在已经存在的卷组中**创建逻辑卷**，该命令的语法格式如下：

**lvcreate** [选项] 卷组名/路径 物理卷路径

lvcreate命令的常用选项如表所示。

选项	说明
-l	以PE为单位指定逻辑卷容量
-L	指定逻辑卷的容量，单位为B/S/K/M/G/T/P/E
-n	指定逻辑卷的名称

**示例：**lvcreate -L 2G -n my\_lv1 itcast



## 2.4.4 LVM逻辑卷管理



### 4、vgdisplay

vgdisplay命令用于显示LVM卷组的信息，该命令的格式如下：

vgdisplay [选项] 卷组

vgdisplay命令的常用选项如表所示。

选项	说明
-s	使用短格式输出信息
-A	仅显示活动卷组的属性

**示例：**vgdisplay itcast



## 2.4.4 LVM逻辑卷管理



### 5、lvextend

当逻辑卷的可用存储空间不足时，我们需要为其拓展存储空间。使用LVM机制管理磁盘时可通过lvextend命令动态地调整分区的大小，该命令的格式如下：

lvextend [选项] 逻辑卷

lvextend命令的常用选项如表所示。

选项	说明
-l	以PE为单位指定逻辑卷容量
-L	指定逻辑卷的容量，单位为B/S/K/M/G/T/P/E



## 2.4.4 LVM逻辑卷管理



### 5、lvextend

当逻辑卷的可用存储空间不足时，我们需要为其拓展存储空间。使用LVM机制管理磁盘时可通过lvextend命令动态地调整分区的大小，该命令的格式如下：

lvextend [选项] 逻辑卷

lvextend命令的常用选项如表所示。

**示例：**

```
lvextend -l +1G /dev/itcast/my_lv1
```

```
lvextend -L 5G /dev/itcast/my_lv1
```



## 2.4.4 LVM逻辑卷管理



### 6、lvremove

lvremove命令用于删除指定的LVM逻辑卷，若逻辑卷已经被挂载到系统中，则应先使用umount命令将其卸载，再进行删除。lvremove命令的语法格式如下：

**lvremove [选项] 逻辑卷**

lvremove命令的常用选项为-f，其功能为强制删除指定逻辑卷。

**示例：**

```
lvremove /dev/itcast/mylv1
```





## 2.4.5 RAID磁盘阵列



RAID (Redundant Arrays of Independent Disks, 即磁盘阵列) 的核心思想是将多个独立的物理磁盘按照某些方式组合成一个逻辑磁盘, 这种技术早期的研究目的是使用多个廉价小磁盘替代大容量磁盘, 节约成本, 随着磁盘的发展, RAID技术更侧重于提高磁盘容错功能与传输速率, 提升磁盘性能。



## 2.4.5 RAID磁盘阵列

RAID分为**软件RAID**和**硬件RAID**，软件RAID和硬件RAID可以实现相同的功能。

### 1、软件RAID

主流操作系统都已集成了软件RAID功能，但由于软件RAID没有独立的硬件控制设备，性能略为低下。

### 2、硬件RAID

硬件RAID本身有独立的控制部件和内存，不占用系统资源，且效率高、性能强。

RAID卡价格低廉，目前市面上的多数主板已集成了RAID卡，因此硬件RAID存储系统常被用于生产环境中。

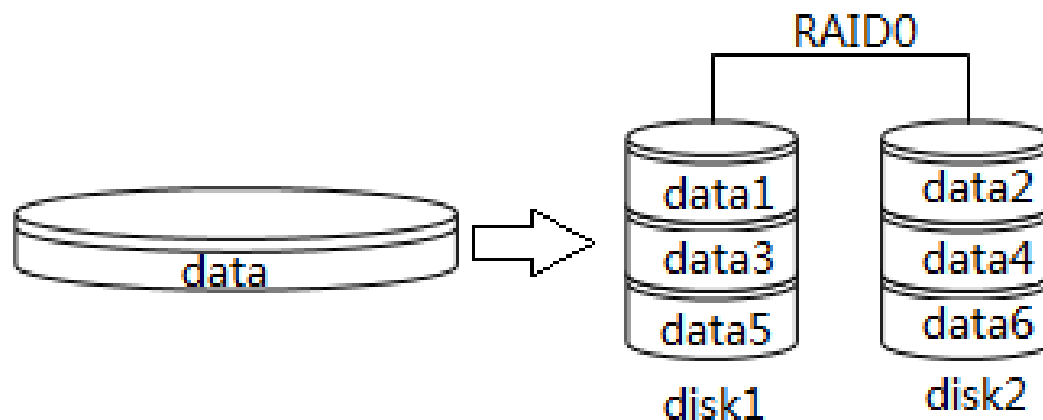


## 2.4.5 RAID磁盘阵列

根据磁盘中数据的存取方式，RAID分为多个级别，其中较为常用的级别有RAID0、RAID1、RAID10、RAID5等。

### 1、RAID0

- RAID0最少需要两块磁盘
- 多个磁盘中分散存放连续的数据，
- 多个磁盘并行执行数据存取



特点：

场合：

- ◆ 存取效率高
- ◆ 不具备容错能力

- ◆ 成本和效率要求较高
- ◆ 可靠性要求较低



## 2.4.5 RAID磁盘阵列

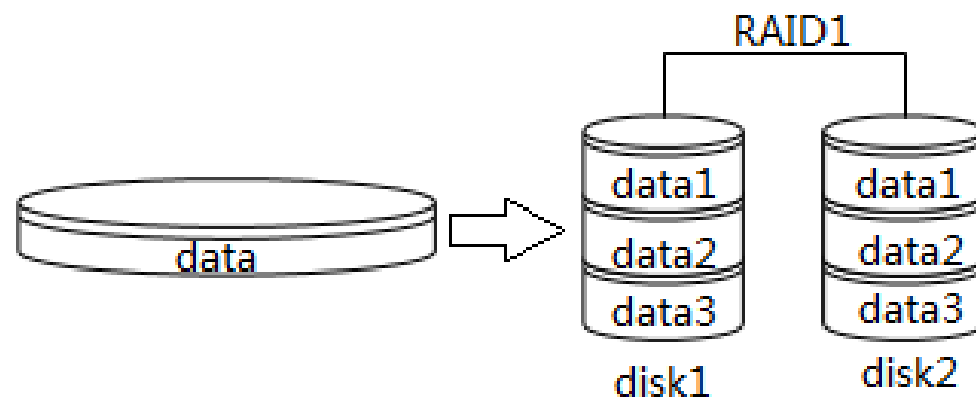
根据磁盘中数据的存取方式，RAID分为多个级别，其中较为常用的级别有RAID0、RAID1、RAID10、RAID5等。

### 2、RAID1

- RAID1至少需要两块磁盘
- 将数据同时复制到每块硬盘
- 不含校验

特点：

- ◆ 具备容错能力
- ◆ 磁盘利用率与写入效率较低





## 2.4.5 RAID磁盘阵列

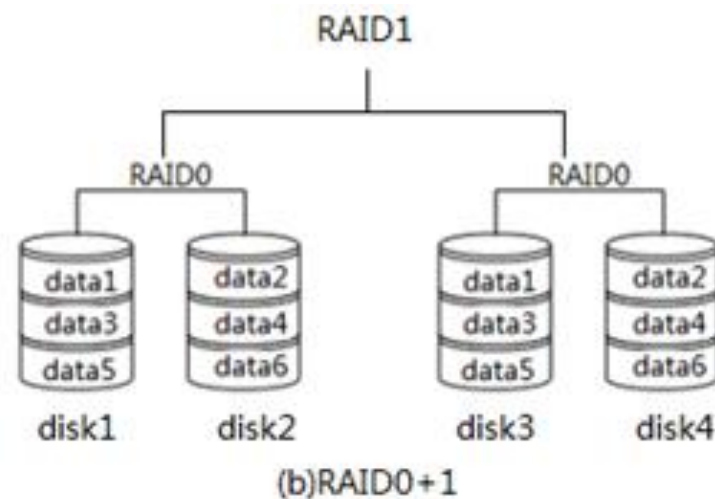
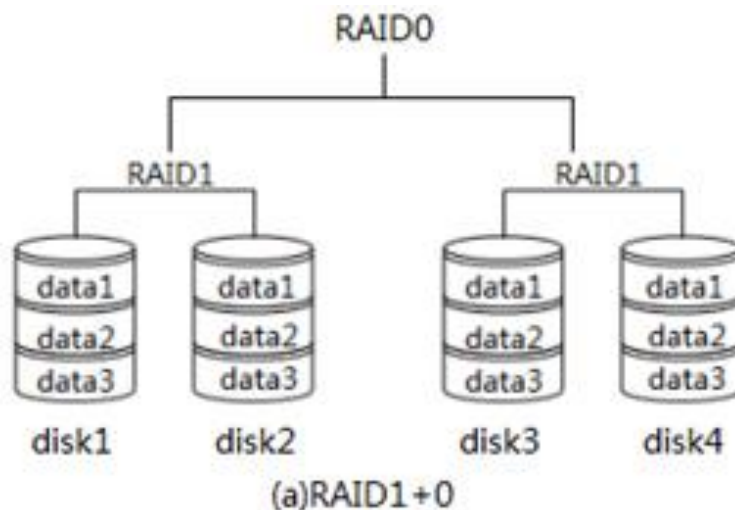
根据磁盘中数据的存取方式，RAID分为多个级别，其中较为常用的级别有RAID0、RAID1、RAID10、RAID5等。

### 3、RAID10

- RAID1+0
- RAID0+1
- 至少需要四块磁盘

特点：

- ◆ 高可靠性、高效率
- ◆ 高成本、低磁盘利用率较低





## 2.4.5 RAID磁盘阵列

根据磁盘中数据的存取方式，RAID分为多个级别，其中较为常用的级别有RAID0、RAID1、RAID10、RAID5等。

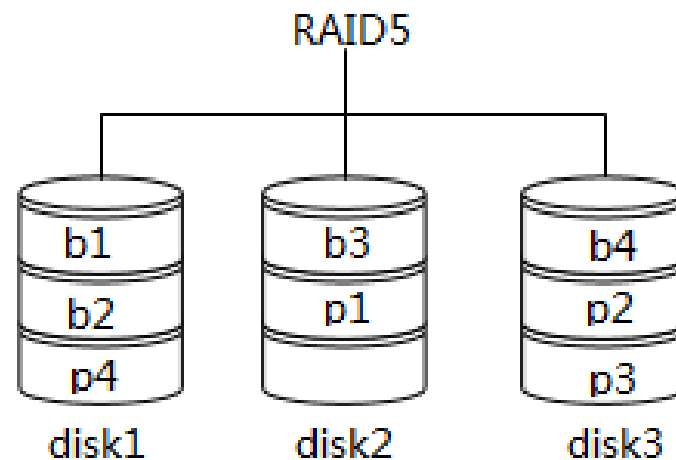
### 3、RAID5

- 至少需要三块磁盘
- 将数据以块为单位存储到各个磁盘中
- 将各块数据的校验信息分别存储到

RAID5的各个磁盘中

特点：

- ◆ 并行存取
- ◆ 存储成本与读写效率处于中间水准





## 2.4.6创建RAID

Linux系统中使用**mdadm**命令创建和管理RAID，该命令的全称是multiple devices admin，其语法格式如下：

**mdadm [模式] <RAID设备名> [选项] <组件设备名>**

工作模式	说明
-A/--assemble	组合。将一个本属于一个已存在阵列的每个设备组装为阵列
-B/--build	构建。构建一个不需要超级块的阵列，阵列中的每个设备都没有超级块
-C/--create	创建。创建一个新阵列
-F/--follow/--monitor	监视。监视一个或多个阵列的状态
-G/--grow	增长。更改RAID的容量或阵列中的设备数目
--auto-detect	自动侦测。请求内核启动任何自动检测的阵列
-I/--incremental	增加。向阵列中添加单个设备，或从阵列中删除单个设备



## 2.4.6创建RAID

Linux系统中使用`mdadm`命令创建和管理RAID，该命令的全称是multiple devices admin，其语法格式如下：

`mdadm [模式] <RAID设备名> [选项] <组件设备名>`

`mdadm`命令在-C模式下的常用选项如表所示。

选项	所需磁盘数	说明
-l		指定RAID级别
-n		指定设备数量
-a {yes no}		是否自动为其创建设备文件
-c		指定数据块大小
-x		指定空闲盘个数，空闲盘可自动顶替损坏的工作盘

**示例：** `mdadm -C /dev/md0 -l 0 -n 2 /dev/sda5 /dev/sda6`