

09 【实操篇-磁盘查看和分区类】

1.du 查看文件和目录占用的磁盘空间

du: disk usage 磁盘占用情况

du 是统计目录或文件所占磁盘空间大小的命令。

需要注意的是，使用"ls -l"命令是可以看到文件的大小的。但是大家会发现，在使用"ls -l"命令查看目录大小时，目录的大小多数是 4KB，这是因为目录下的子目录名和子文件名是保存到父目录的 block（默认大小为 4KB）中的，如果父目录下的子目录和子文件并不多，一个 block 就能放下，那么这个父目录就只占了一个 block 大小。

大家可以将其想象成图书馆的书籍目录和实际书籍。如果我们用"ls -l"命令查看，则只能看到这些书籍占用了 1 页纸的书籍目录，但是实际书籍到底有多少是看不到的，哪怕它堆满了几个房间。

但是我们在统计目录时，不是想看父目录下的子目录名和子文件名到底占用了多少空间，而是想看父目录下的子目录和子文件的总磁盘占用量大小，这时就需要使用 du 命令才能统计目录的真正磁盘占用量大小。

du 命令的格式如下：

```
[root@localhost ~]# du [选项] [目录或文件名]
```

选项：

- -a：显示每个子文件的磁盘占用量。默认只统计子目录的磁盘占用量
- -h：使用习惯单位显示磁盘占用量，如 KB、MB 或 GB 等；
- -s：统计总磁盘占用量，而不列出子目录和子文件的磁盘占用量

【例 1】

```
[root@localhost ~]# du
#统计当前目录的总磁盘占用量大小，同时会统计当前目录下所有子目录的磁盘占用量大小，不统计子文件
#磁盘占用量的大小。默认单位为KB
20 ./gnupg
#统计每个子目录的大小
24 ./yum.bak
8 ./dtest
28 ./sh
188
#统计当前目录总大小
```

【例 2】

```
[root@localhost ~]# du -a
```

#统计当前目录的总大小，同时会统计当前目录下所有子文件和子目录磁盘占用量的大小。默认单位为 KB

```
4 ./bashjogout
36 ./install.log
4 ./bash_profile
4 ./cshrc
...省略部分输出...
188
```

【例 3】

```
[root@localhost ~]# du -sh
```

#只统计磁盘占用量总的大小，同时使用习惯单位显示

```
188K .
```

2.df 查看磁盘空间使用情况

df: disk free 空余磁盘

df 命令，用于显示 Linux 系统中各文件系统的硬盘使用情况，包括文件系统所在硬盘分区的总容量、已使用的容量、剩余容量等。

df 命令主要读取的数据几乎都针对的是整个文件系统。

df 命令的基本格式为：

```
[root@localhost ~]# df [选项] [目录或文件名]
```

选项	作用
-a	显示所有文件系统信息，包括系统特有的 /proc、/sysfs 等文件系统；
-m	以 MB 为单位显示容量；
-k	以 KB 为单位显示容量，默认以 KB 为单位；
-h	使用人们习惯的 KB、MB 或 GB 等单位自行显示容量；
-T	显示该分区的文件系统名称；
-i	不用硬盘容量显示，而是以含有 inode 的数量来显示。

【例 1】

```
[root@localhost ~]# df
```

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hdc2	9920624	3823112	5585444	41%	/
/dev/hdc3	4956316	141376	4559108	4%	/home
/dev/hdc1	101086	11126	84741	12%	/boot
tmpfs	371332	0	371332	0%	/dev/shm

不使用任何选项的 df 命令，默认会将系统内所有的文件系统信息，以 KB 为单位显示出来。

本例中，由 df 命令显示出的各列信息的含义分别是：

- Filesystem：表示该文件系统位于哪个分区，因此该列显示的是设备名称；
- 1K-blocks：此列表示文件系统的总大小，默认以 KB 为单位；
- Used：表示用掉的硬盘空间大小；
- Available：表示剩余的硬盘空间大小；
- Use%：硬盘空间使用率。如果使用率高达 90% 以上，就需要额外注意，因为容量不足，会严重影响系统的正常运行；
- Mounted on：文件系统的挂载点，也就是硬盘挂载的目录位置。

【例 2】

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hdc2       9.5G  3.7G  5.4G  41% /
/dev/hdc3       4.8G  139M  4.4G   4% /home
/dev/hdc1       99M   11M   83M  12% /boot
tmpfs          363M    0  363M   0% /dev/shm
```

同例 1 不同，这里使用了 -h 选项，因此文件系统的各种容量数据，会以人们习惯的单位（通常使用 GB 或 MB）显示出来。

【例 3】

```
[root@localhost ~]# df -h /etc
Filesystem      Size  Used Avail Use% Mounted on
/dev/hdc2       9.5G  3.7G  5.4G  41% /
```

同之前的 2 个例子不同，这里在 df 命令后添加了目录名，在这种情况下，df 命令会自动分析该目录所在的分区，并将所在分区的有关信息显示出来。由此，我们就可以知道，该目录下还可以使用多少容量。

【例 4】

```
[root@localhost ~]# df -at
Filesystem      Type 1k-blocks    Used Available Use% Mounted on
/dev/hdc2       ext3 9920624 3823112  5585444  41% /
proc            proc      0         0         0    -  /proc
sysfs           sysfs      0         0         0    -  /sys
devpts          devpts     0         0         0    -  /dev/pts
/dev/hdc3       ext3 4956316 141376  4559108   4% /home
/dev/hdc1       ext3 101086 11126  84741  12% /boot
tmpfs           tmpfs 371332      0  371332   0% /dev/shm
none            binfmt_misc 0         0         0    -  /proc/sys/fs/binfmt_misc
sunrpc          rpc_pipefs 0         0         0    -  /var/lib/nfs/rpc_pipefs
```

注意，使用 -a 选项，会将很多特殊的文件系统显示出来，这些文件系统包含的大多是系统数据，存在于内存中，不会占用硬盘空间，因此你会看到，它们所占据的硬盘总容量为 0。

3.du命令和df命令的区别

有时我们会发现，使用 du 命令和 df 命令去统计分区的使用情况时，得到的数据是不一样的。那是因为 df 命令是从文件系统的角度考虑的，通过文件系统中未分配的空间来确定文件系统中已经分配的空间大小。也就是说，在使用 df 命令统计分区时，不仅要考虑文件占用的空间，还要统计被命令或程序占用的空间（最常见的就是文件已经删除，但是程序并没有释放空间）。

而 du 命令是面向文件的，只会计算文件或目录占用的磁盘空间。也就是说，df 命令统计的分区更准确，是真正的空闲空间。

4.lsbblk 查看设备挂载情况

lsblk命令的英文是“list block”，即用于列出所有可用块设备的信息，而且还能显示他们之间的依赖关系，但是它不会列出RAM盘的信息。块设备有硬盘，闪存盘，CD-ROM等等。lsblk命令包含在util-linux-ng包中，现在该包改名为util-linux。

【例1】列出所有块设备

直接输入lsblk命令和lsblk -a输出相同

```
[root@test1 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 40G 0 disk
├─sda1 8:1 0 300M 0 part /boot
├─sda2 8:2 0 2G 0 part [SWAP]
└─sda3 8:3 0 37.7G 0 part /
sr0 11:0 1 1024M 0 rom
```

- NAME：这是块设备名。
- MAJ:MIN：本栏显示主要和次要设备号。
- RM：本栏显示设备是否可移动设备。注意，在本例中设备sdb和sr0的RM值等于1，这说明他们是可移动设备。
- SIZE：本栏列出设备的容量大小信息。例如298.1G表明该设备大小为298.1GB，而1K表明该设备大小为1KB。
- RO：该项表明设备是否为只读。在本案例中，所有设备的RO值为0，表明他们不是只读的。
- TYPE：本栏显示块设备是否是磁盘或磁盘上的一个分区。在本例中，sda和sdb是磁盘，而sr0是只读存储（rom）。
- MOUNTPOINT：本栏指出设备挂载的挂载点。

5.mount/umount 挂载/卸载

对于Linux用户来讲，不论有几个分区，分别分给哪一个目录使用，它总归就是一个根目录、一个独立且唯一的文件结构。

Linux中每个分区都是用来组成整个文件系统的一部分，它在用一种叫做“挂载”的处理方法，它整个文件系统中包含了一整套的文件和目录，并将一个分区和一个目录联系起来，要载入的那个分区将使它的存储空间在这个目录下获得。

5.1 mount 挂载

所有的硬件设备必须挂载之后才能使用，只不过，有些硬件设备（比如硬盘分区）在每次系统启动时会自动挂载，而有些（比如U盘、光盘）则需要手动进行挂载。

通过学习 Linux 文件系统，我们可以对挂载的含义进行引申，挂载指的是将硬件设备的文件系统和 Linux 系统中的文件系统，通过指定目录（作为挂载点）进行关联。而要将文件系统挂载到 Linux 系统上，就需要使用 mount 挂载命令。

mount 命令的常用格式有以下几种：

```
[root@localhost ~]# mount [-l]
```

单纯使用 mount 命令，会显示出系统中已挂载的设备信息，使用 -l 选项，会额外显示出卷标名称（读者可自行运行，查看输出结果）；

```
[root@localhost ~]# mount -a
```

-a 选项的含义是自动检查 /etc/fstab 文件中是否有疏漏被挂载的设备文件，如果有，则进行自动挂载操作。这里简单介绍一下 /etc/fstab 文件，此文件是自动挂载文件，系统开机时会主动读取 /etc/fstab 这个文件中的内容，根据该文件的配置，系统会自动挂载指定设备。

```
[root@localhost ~]# mount [-t 系统类型] [-L 卷标名] [-o 特殊选项] [-n] 设备文件名 挂载点
```

各选项的含义分别是：

- -t 系统类型：指定欲挂载的文件系统类型。Linux 常见的支持类型有 EXT2、EXT3、EXT4、iso9660（光盘格式）、vfat、reiserfs 等。如果不指定具体类型，挂载时 Linux 会自动检测。
- -L 卷标名：除了使用设备文件名（例如 /dev/hdc6）之外，还可以利用文件系统的卷标名称进行挂载。
- -n：在默认情况下，系统会将实际挂载的情况实时写入 /etc/mtab 文件中，但在某些场景下（例如单人维护模式），为了避免出现问题，会刻意不写入，此时就需要使用这个选项；
- -o 特殊选项：可以指定挂载的额外选项，比如读写权限、同步/异步等，如果不指定，则使用默认值（defaults）。具体的特殊选项参见表 1；

选项	功能
rw/ro	是否对挂载的文件系统拥有读写权限，rw 为默认值，表示拥有读写权限；ro 表示只读权限。
async/sync	此文件系统是否使用同步写入（sync）或异步（async）的内存机制，默认为异步 async。
dev/nodev	是否允许从该文件系统的 block 文件中提取数据，为了保证数据安全，默认是 nodev。
auto/noauto	是否允许此文件系统被以 mount -a 的方式进行自动挂载，默认是 auto。
suid/nosuid	设定文件系统是否拥有 SetUID 和 SetGID 权限，默认是拥有。
exec/noexec	设定在文件系统中是否允许执行可执行文件，默认是允许。
user/nouser	设定此文件系统是否允许让普通用户使用 mount 执行实现挂载，默认是不允许（nouser），仅有 root 可以。
defaults	定义默认值，相当于 rw、suid、dev、exec、auto、nouser、async 这 7 个选项。
remount	重新挂载已挂载的文件系统，一般用于指定修改特殊权限。

【例 1】

```
[root@localhost ~]# mount
#查看系统中已经挂载的文件系统，注意有虚拟文件系统
/dev/sda3 on / type ext4 (rw) <--含义是，将 /dev/sda3 分区挂载到了 / 目录上，文件系统是
ext4，具有读写权限
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw, gid=5, mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

【例 2】

修改特殊权限。通过例 1 我们查看到，/boot 分区已经被挂载了，而且采用的是 defaults 选项。这里我们重新挂载分区，并采用 noexec 权限禁止执行文件执行，看看会出现什么情况（注意不要用 / 分区做实验，否则系统命令也就不能执行了）。

```
[root@localhost ~]# mount -o remount noexec /boot
#重新挂载 /boot 分区，并使用 noexec 权限
[root@localhost ~]# cd /boot
#写一个 shell 脚本，看是否会运行
[root@localhost boot]# vi hello.sh
#!/bin/bash
echo "hello!!"
[root@localhost boot]# chmod 755 hello.sh
[root@localhost boot]# ./hello.sh
-bash: ./hello.sh: 权限不够
#虽然赋予了hello.sh执行权限，但是仍然无法执行
[root@localhost boot]# mount -o remount exec /boot
#记得改回来，否则会影响系统启动
```

对于特殊选项的修改，除非特殊场景下需要，否则不建议大家随意修改，非常容易造成系统出现问题，而且还找不到问题的根源。

【例 3】挂载分区。

```
[root@localhost ~]# mkdir /mnt/disk1
#建立挂载点目录
[root@localhost ~]# mount /dev/sdb1 /mnt/disk1
#挂载分区
```

/dev/sdb1 分区还没有被划分。我们在这里只看看挂载分区的方式，非常简单，甚至不需要使用 "-ext4" 命令指定文件系统，因为系统可以自动检测。

为什么使用 Linux 系统的硬盘分区这么麻烦，而不能像 Windows 系统那样，硬盘安装上就可以使用？

其实，硬盘分区（设备）挂载和卸载（使用 umount 命令）的概念源自 UNIX，UNIX 系统一般是作为服务器使用的，系统安全非常重要，特别是在网络上，最简单有效的方法就是“不使用的硬盘分区（设备）不挂载”，因为没有挂载的硬盘分区是无法访问的，这样系统也就更安全了。

另外，这样也可以减少挂载的硬盘分区数量，相应地，也就可以减少系统维护文件的规模，当然也就减少了系统的开销，即提高了系统的效率。

5.2 umount 卸载

硬盘分区是否需要卸载，取决于你下次是否还需要使用，一般不对硬盘分区执行卸载操作。

umount 命令用于卸载已经挂载的硬件设备，该命令的基本格式如下：

```
[root@localhost ~]# umount 设备文件名或挂载点
```

注意，卸载命令后面既可以加设备文件名，也可以加挂载点，不过只能二选一，比如：

```
[root@localhost ~]# umount /mnt/usb
#卸载U盘
[root@localhost ~]# umount /mnt/cdrom
#卸载光盘
[root@localhost ~]# umount /dev/sr0
#命令加设备文件名同样是可以卸载的
```

如果加了两个（如下所示），从理论上分析，会对光驱卸载两次，当然，卸载第二次的时候就会报错。

```
[root@localhost ~]# mount /dev/sr0 /mnt/cdrom/
```

另外，我们在卸载时有可能出现以下情况：

```
[root@localhost ~]# cd /mnt/cdrom/
#进入光盘挂载点
[root@localhost cdrom]# umount /mnt/cdrom/
umount: /mnt/cdrom: device is busy.
#报错，设备正忙
```

这种报错是因为我们已经进入了挂载点，因此，如果要卸载某硬件设备，在执行 umount 命令之前，用户须退出挂载目录。

卸载硬件设备成功与否，除了执行 umount 命令不报错之外，还可以使用 df 命令或 `mount -l` 来查看目标设备是否还挂载在系统中。

6.fdisk 分区

我们在安装操作系统的过程中已经对系统硬盘进行了分区，但如果新添加了一块硬盘，想要正常使用，难道需要重新安装操作系统才可以分区吗？

当然不是，在 Linux 中有专门的分区命令 fdisk 和 parted。其中 fdisk 命令较为常用，但不支持大于 2TB 的分区；如果需要支持大于 2TB 的分区，则需要使用 parted 命令，当然 parted 命令也能分配较小的分区。我们先来看看如何使用 fdisk 命令进行分区。

fdisk 命令的格式如下：

```
[root@localhost ~]# fdisk ~l
#列出系统分区
[root@localhost ~]# fdisk 设备文件名
#给硬盘分区
```

该命令必须在 root 用户下才能使用

使用 "fdisk -l" 查看分区信息，能够看到我们添加的两块硬盘 (/dev/sda 和 /dev/sdb) 的信息。我们解释一下这些信息，其上半部分态是硬盘的整体状态，/dev/sda 硬盘的总大小是 32.2 GB，共有 3916 个柱面，每个柱面由 255 个磁头读/写数据，每个磁头管理 63 个扇区。每个柱面的大小是 8225280 Bytes，每个扇区的大小是 512 Bytes。

信息的下半部分是分区的信息，共 7 列，含义如下：

- Device：分区的设备文件名。
- Boot：是否为启动引导分区，在这里 /dev/sda1 为启动引导分区。
- Start：起始柱面，代表分区从哪里开始。
- End：终止柱面，代表分区到哪里结束。
- Blocks：分区的大小，单位是 KB。
- id：分区内文件系统的 ID。在 fdisk 命令中，可以使用 "i" 查看。
- System：分区内安装的系统是什么。

在 fdisk 交互界面中输入 m 可以得到帮助，帮助里列出了 fdisk 可以识别的交互命令，我们来解释一下这些命令

命令	说 明
a	设置可引导标记
b	编辑 bsd 磁盘标签
c	设置 DOS 操作系统兼容标记
d	删除一个分区
l	显示已知的文件系统类型。82 为 Linux swap 分区，83 为 Linux 分区
m	显示帮助菜单
n	新建分区
O	建立空白 DOS 分区表
P	显示分区列表
q	不保存退出
s	新建空白 SUN 磁盘标签
t	改变一个分区的系统 ID
u	改变显示记录单位
V	验证分区表
w	保存退出