

# Linux 磁盘管理

Linux磁盘管理好坏直接关系到整个系统的性能问题。

Linux磁盘管理常用三个命令为df、du和fdisk。

- df：列出文件系统的整体磁盘使用量
- du：检查磁盘空间使用量
- fdisk：用于磁盘分区

## df

df命令参数功能：检查文件系统的磁盘空间占用情况。可以利用该命令来获取硬盘被占用了多少空间，目前还剩下多少空间等信息。

语法：

```
df [-ahikHTm] [目录或文件名]
```

选项与参数：

- -a：列出所有的文件系统，包括系统特有的 /proc 等文件系统；
- -k：以 KBytes 的容量显示各文件系统；
- -m：以 MBytes 的容量显示各文件系统；
- -h：以人们较易阅读的 GBytes, MBytes, KBytes 等格式自行显示；
- -H：以 M=1000K 取代 M=1024K 的进位方式；
- -T：显示文件系统类型, 连同该 partition 的 filesystem 名称 (例如 ext3) 也列出；
- -i：不用硬盘容量，而以 inode 的数量来显示

## 实例 1

将系统内所有的文件系统列出来！

```
[root@www ~]# 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
```

在 Linux 底下如果 df 没有加任何选项，那么默认会将系统内所有的 (不含特殊内存内的文件系统与 swap) 都以 1 Kbytes 的容量来列出来！

## 实例 2

将容量结果以易读的容量格式显示出来

```
[root@www ~]# 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

## 实例 3

将系统内的所有特殊文件格式及名称都列出来

```
[root@www ~]# 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

## 实例 4

将 /etc 底下的可用的磁盘容量以易读的容量格式显示

```
[root@www ~]# df -h /etc
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/hdc2	9.5G	3.7G	5.4G	41%	/

## du

Linux du命令也是查看使用空间的，但是与df命令不同的是Linux du命令是对文件和目录磁盘使用的空间的查看，还是和df命令有一些区别的，这里介绍Linux du命令。

语法：

```
du [-ahskm] 文件或目录名称
```

选项与参数：

- `-a`：列出所有的文件与目录容量，因为默认仅统计目录底下的文件量而已。
- `-h`：以人们较易读的容量格式 (G/M) 显示；
- `-s`：列出总量而已，而不列出每个各别的目录占用容量；
- `-S`：不包括子目录下的总计，与 `-s` 有点差别。
- `-k`：以 KBytes 列出容量显示；
- `-m`：以 MBytes 列出容量显示；

## 实例 1

列出目前目录下的所有文件容量

```
[root@www ~]# du
8      ./test4      <==每个目录都会列出来
8      ./test2
....中间省略....
12     ./gconfd     <==包括隐藏文件的目录
220    .            <==这个目录(.)所占用的总量
```

直接输入 `du` 没有加任何选项时，则 `du` 会分析当前所在目录的文件与目录所占用的硬盘空间。

## 实例 2

将文件的容量也列出来

```
[root@www ~]# du -a
12     ./install.log.syslog  <==有文件的列表了
8      ./bash_logout
8      ./test4
8      ./test2
....中间省略....
12     ./gconfd
220    .
```

## 实例 3

检查根目录下每个目录所占用的容量

```
[root@www ~]# du -sm /*
7      /bin
6      /boot
.....中间省略....
0      /proc
.....中间省略....
```

1	/tmp	
3859	/usr	<==系统初期最大就是他了啦！
77	/var	

通配符 \* 来代表每个目录。

与 df 不一样的是，du 这个命令其实会直接到文件系统内去搜寻所有的文件数据。

## fdisk

fdisk 是 Linux 的磁盘分区表操作工具。

语法：

```
fdisk [-l] 装置名称
```

选项与参数：

- -l：输出后面接的装置所有的分区内容。若仅有 fdisk -l 时，则系统将会把整个系统内能够搜寻到的装置的分区均列出来。

### 实例 1

列出所有分区信息

```
[root@AY120919111755c246621 tmp]# fdisk -l

Disk /dev/xvda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

    Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *           1         2550       2048000    83  Linux
/dev/xvda2           2550         2611        490496    82  Linux swap / Solaris

Disk /dev/xvdb: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x56f40944

    Device Boot      Start         End      Blocks   Id  System
/dev/xvdb2           1         2610       20964793+    83  Linux
```

### 实例 2

找出你系统中的根目录所在磁盘，并查阅该硬盘内的相关信息

```
[root@www ~]# df /          <==注意：重点在找出磁盘文件名而已
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/hdc2        9920624    3823168   5585388   41% /
```

```
[root@www ~]# fdisk /dev/hdc <==仔细看，不要加上数字喔！
The number of cylinders for this disk is set to 5005.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
```

```
Command (m for help):      <==等待你的输入！
```

输入 m 后，就会看到底下这些命令介绍

```
Command (m for help): m    <== 输入 m 后，就会看到底下这些命令介绍
```

```
Command action
```

```
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition          <==删除一个partition
  l  list known partition types
  m  print this menu
  n  add a new partition         <==新增一个partition
  o  create a new empty DOS partition table
  p  print the partition table   <==在屏幕上显示分割表
  q  quit without saving changes <==不储存离开fdisk程序
  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 <==将刚刚的动作写入分割表
  x  extra functionality (experts only)
```

离开 fdisk 时按下 q，那么所有的动作都不会生效！相反的，按下 w 就是动作生效的意思。

```
Command (m for help): p    <== 这里可以输出目前磁盘的状态
```

```
Disk /dev/hdc: 41.1 GB, 41174138880 bytes          <==这个磁盘的文件名与容量
255 heads, 63 sectors/track, 5005 cylinders       <==磁头、扇区与磁柱大小
Units = cylinders of 16065 * 512 = 8225280 bytes <==每个磁柱的大小
```

```
Device Boot      Start          End      Blocks   Id  System
```

```

/dev/hdc1    *           1           13           104391    83    Linux
/dev/hdc2           14           1288    10241437+   83    Linux
/dev/hdc3           1289           1925    5116702+   83    Linux
/dev/hdc4           1926           5005    24740100     5    Extended
/dev/hdc5           1926           2052    1020096    82    Linux swap / Solaris
# 装置文件名 启动区否 开始磁柱    结束磁柱  1K大小容量 磁盘分区槽内的系统

Command (m for help): q

```

想要不储存离开吗？按下 q 就对了！不要随便按 w 啊！

使用 p 可以列出目前这颗磁盘的分割表信息，这个信息的上半部在显示整体磁盘的状态。

## 磁盘格式化

磁盘分割完毕后自然就是要进行文件系统的格式化，格式化的命令非常的简单，使用 `mkfs` ( `make filesystem` ) 命令。

语法：

```
mkfs [-t 文件系统格式] 装置文件名
```

选项与参数：

- -t ：可以接文件系统格式，例如 ext3, ext2, vfat 等(系统有支持才会生效)

### 实例 1

查看 mkfs 支持的文件格式

```

[root@www ~]# mkfs[tab][tab]
mkfs          mkfs.cramfs  mkfs.ext2     mkfs.ext3     mkfs.msdos    mkfs.vfat

```

按下两个[tab]，会发现 mkfs 支持的文件格式如上所示。

### 实例 2

将分区 /dev/hdc6 ( 可指定你自己的分区 ) 格式化为 ext3 文件系统：

```

[root@www ~]# mkfs -t ext3 /dev/hdc6
mke2fs 1.39 (29-May-2006)
Filesystem label=          <==这里指的是分割槽的名称(label)
OS type: Linux
Block size=4096 (log=2)    <==block 的大小配置为 4K
Fragment size=4096 (log=2)
251392 inodes, 502023 blocks  <==由此配置决定的inode/block数量
25101 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=515899392
16 block groups

```

```
32768 blocks per group, 32768 fragments per group
15712 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (8192 blocks): done <==有日志记录
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 34 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
# 这样就创建起来我们所需要的 Ext3 文件系统了！简单明了！
```

## 磁盘检验

fsck ( file system check ) 用来检查和维护不一致的文件系统。

若系统掉电或磁盘发生问题，可利用fsck命令对文件系统进行检查。

语法：

```
fsck [-t 文件系统] [-ACay] 装置名称
```

选项与参数：

- -t：给定档案系统的型式，若在 /etc/fstab 中已有定义或 kernel 本身已支援的则不需加上此参数
- -s：依序一个一个地执行 fsck 的指令来检查
- -A：对/etc/fstab 中所有列出来的 分区 ( partition ) 做检查
- -C：显示完整的检查进度
- -d：打印出 e2fsck 的 debug 结果
- -p：同时有 -A 条件时，同时有多个 fsck 的检查一起执行
- -R：同时有 -A 条件时，省略 / 不检查
- -V：详细显示模式
- -a：如果检查有错则自动修复
- -r：如果检查有错则由使用者回答是否修复
- -y：选项指定检测每个文件是自动输入yes，在不确定那些是不正常的时候，可以执行 # fsck -y 全部检查修复。

### 实例 1

查看系统有多少文件系统支持的 fsck 命令：

```
[root@www ~]# fsck[tab][tab]
fsck      fsck.cramfs  fsck.ext2    fsck.ext3    fsck.msdos    fsck.vfat
```

## 实例 2

强制检测 /dev/hdc6 分区:

```
[root@www ~]# fsck -C -f -t ext3 /dev/hdc6
fsck 1.39 (29-May-2006)
e2fsck 1.39 (29-May-2006)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
vbird_logical: 11/251968 files (9.1% non-contiguous), 36926/1004046 blocks
```

如果没有加上 -f 的选项，则由于这个文件系统不曾出现问题，检查的经过非常快速！若加上 -f 强制检查，才会一项一项的显示过程。

## 磁盘挂载与卸载

Linux 的磁盘挂载使用 `mount` 命令，卸载使用 `umount` 命令。

磁盘挂载语法：

```
mount [-t 文件系统] [-L Label名] [-o 额外选项] [-n] 装置文件名 挂载点
```

## 实例 1

用默认的方式，将刚刚创建的 /dev/hdc6 挂载到 /mnt/hdc6 上面！

```
[root@www ~]# mkdir /mnt/hdc6
[root@www ~]# mount /dev/hdc6 /mnt/hdc6
[root@www ~]# df
Filesystem            1K-blocks      Used Available Use% Mounted on
.....中间省略.....
/dev/hdc6              1976312    42072   1833836   3% /mnt/hdc6
```

磁盘卸载命令 `umount` 语法：

```
umount [-fn] 装置文件名或挂载点
```

选项与参数：

- -f：强制卸载！可用在类似网络文件系统 (NFS) 无法读取到的情况下；
- -n：不升级 /etc/mtab 情况下卸载。



卸载/dev/hdc6

```
[root@www ~]# umount /dev/hdc6
```

← Linux 用户和用户组管理

Linux vi/vim →



## 1 篇笔记



## 写笔记



Linux 操作系统的文件数据除了文件实际内容外，通常含有非常多的属性，例如 Linux 操作系统的文件权限（`rwX`）与文件属性（拥有者、群组、时间参数等）。文件系统通常会将这两部分的数据分别存放在不同的区块，权限与属性存放在 `inode` 中，至于实际数据则放置到 `data block` 区块中。另外，还有一个超级区块（`superblock`）会记录整个文件系统的整体信息，包括 `inode` 与 `block` 的总量、使用量、剩余量等。

`inode`：记录文件的属性，一个文件占用一个 `inode`，同时记录此文件的数据所在的 `block`。

在 Linux 中，可以使用 `stat` 命令查看某个文件的 `inode` 信息：

```
stat /etc/passwd
```

```
linux-peanut:~/Desktop # stat /etc/passwd
```

```
File: `/etc/passwd'
```

```
Size: 2269          Blocks: 8          IO Block: 4096   regular file
```

```
Device: 802h/2050d  Inode: 1149740     Links: 1
```

```
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
```

```
Access: 2018-04-09 17:16:42.000000000 +0800
```

```
Modify: 2018-04-09 17:16:36.000000000 +0800
```

```
Change: 2018-04-09 17:16:36.000000000 +0800
```

```
Birth: -
```

可见，文件的绝大部分信息都存储在 `inode` 中。

花生 11个月前 (04-10)