

存储结构与磁盘划分

1

Linux文件结构

2

物理设备的命名规则

3

文件系统

4

挂载硬件设备

5

添加交换分区

6

软硬链接方式

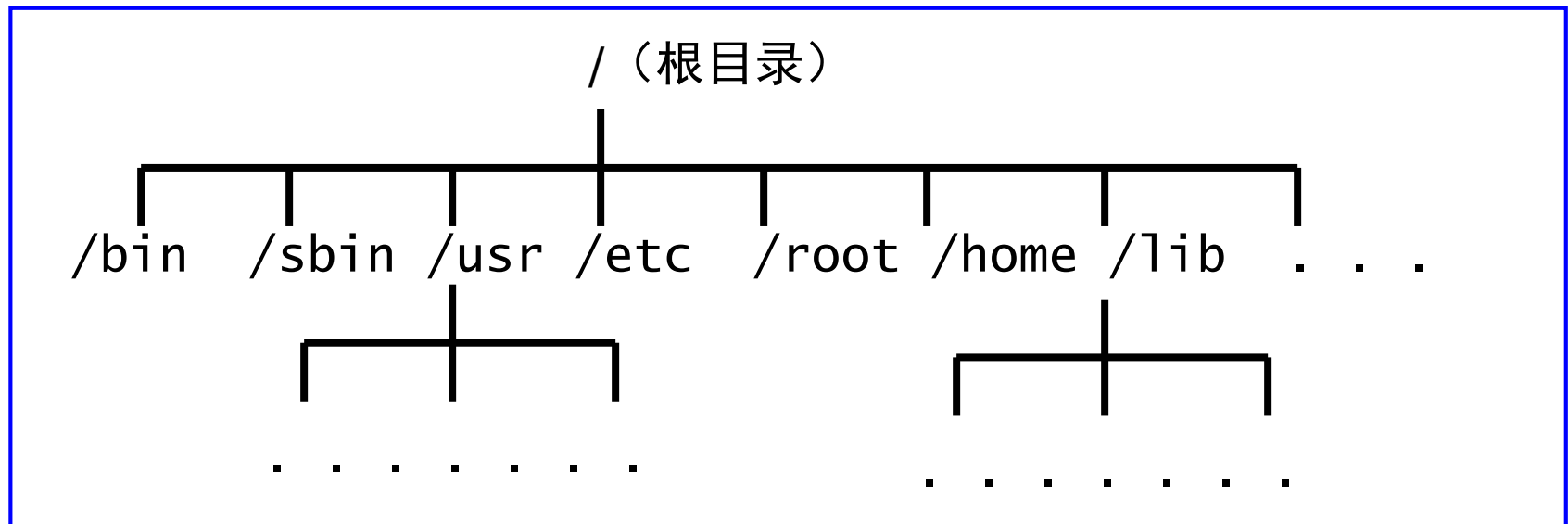


Linux文件结构

Linux文件结构

Linux 文件系统是一个目录树的结构，文件系统结构从一个根目录开始，根目录下可以有任意多个文件和子目录，子目录中又可以有任意多个文件和子目录。

Linux 的这种文件系统结构使得一个目录和它包含的文件/子目录之间形成一种层次关系。



/ → 文件系统结构的起始位置，称为根

-- bin → 存放基本命令程序(任何用户都可以调用)

-- boot → 存放系统启动时所读取的文件，包括系统核心文件

-- dev → 存放设备文件接口，如打印机、硬盘等外围设备

-- etc → 存放与系统设置和管理相关的文件，如用户帐号、密码等

| ...

-- home → 存放用户专属目录（用户主目录）

-- lib → 存放一些共享的函数库

-- misc → 一个空目录，供管理员存放公共杂物

-- proc → 存放系统核心和执行程序之间的信息

-- sbin → 系统管理员（超级用户）专用目录

-- tmp → 与/bin类似，存放用于系统引导和管理命令，通常供root使用。

-- usr → 临时目录，供任何用户存放临时文件。

| -- bin

| -- sbin → 此目录包含许多子目录，用来存放系统命令和程序等信息

| -- man

| ...

-- var → 存放经常变动的文件，如日志文件、临时文件、电子邮件等

...

物理设备命名规则

物理设备的命名规则

硬件设备	文件名称
IDE设备	/dev/hd[a-d]
SCSI/SATA/U盘	/dev/sd[a-p]
软驱	/dev/fd[0-1]
打印机	/dev/lp[0-15]
光驱	/dev/cdrom
鼠标	/dev/mouse
磁带机	/dev/st0或/dev/ht0

- 主分区或扩展分区的编号从1开始，到4结束；
- 逻辑分区从编号5开始。

文件系统

文件系统

- Ext3

一款日志文件系统，能够在系统异常宕机时避免文件系统资料丢失，并能自动修复数据的不一致与错误。然而，。它会把整个磁盘的每个写入动作的细节都预先记录下来，以便在发生异常宕机当硬盘容量较大时，所需的修复时间也会很长，而且也不能百分之百地保证资料不会丢失后能回溯追踪到被中断的部分，然后尝试进行修复。

- Ext4

Ext3的改进版本，作为RHEL 6系统中的默认文件管理系统，它支持的存储容量高达1EB（1EB=1,073,741,824GB），且能够有无限多的子目录。另外，Ext4文件系统能够批量分配block块，从而极大地提高了读写效率。

- XFS

一种高性能的日志文件系统，而且是RHEL 7中默认的文件管理系统，它的优势在发生意外宕机后尤其明显，即可以快速地恢复可能被破坏的文件，而且强大的日志功能只用花费极低的计算和存储性能。并且它最大可支持的存储容量为18EB，这几乎满足了所有需求。

挂载硬件设备

挂载硬件设备

mount命令：

mount命令用于挂载文件系统，格式为“mount 文件系统 挂载目录”。

```
[root@localhost ~]# mount /dev/sda2 /backup
```

这样挂载的文件系统是临时的，会在重启之后消失！

挂载硬件设备

编辑/etc/fstab使设备开机自动挂载

```
[root@linuxprobe ~]# vim /etc/fstab
#
# /etc/fstab
# Created by anaconda on Wed May 4 19:26:23 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/rhel-root / xfs defaults 1 1
UUID=812b1f7c-8b5b-43da-8c06-b9999e0fe48b /boot xfs defaults 1 1
/dev/mapper /rhel-swap swap swap defaults 0 0
/dev/cdrom /media/cdrom iso9660 defaults 0 0 /dev/sda2
/backup ext4 defaults 0 0
```

字段	意义
设备文件	一般为设备的路径+设备名称，也可以写唯一识别码（UUID，Universally Unique Identifier）
挂载目录	指定要挂载到的目录，需在挂载前创建好
格式类型	指定文件系统的格式，比如Ext3、Ext4、XFS、SWAP、iso9660（此为光盘设备）等
权限选项	若设置为defaults，则默认权限为：rw, suid, dev, exec, auto, nouser, async
是否备份	若为1则开机后使用dump进行磁盘备份，为0则不备份
是否自检	若为1则开机后自动进行磁盘自检，为0则不自检

挂载硬件设备

umount命令：

umount命令用于撤销已经挂载的设备文件，格式为 “umount [挂载点/设备文件]”

```
[root@localhost ~]# umount /dev/sda2
```

添加硬盘设备

fdisk工具：

在Linux系统中，管理硬盘设备最常用的方法就当属fdisk命令了。fdisk命令用于管理磁盘分区，格式为“fdisk [磁盘名称]”，它提供了集添加、删除、转换分区等功能于一身的“一站式分区服务”。

参数	作用
m	查看全部可用的参数
n	添加新的分区
d	删除某个分区信息
l	列出所有可用的分区类型
t	改变某个分区的类型
p	查看分区信息
w	保存并退出
q	不保存直接退出

```
[root@localhost ~]# fdisk /dev/sdb
命令(输入 m 获取帮助): p
磁盘 /dev/sdb: 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: 0x782338b8
  设备 Boot      Start          End      Blocks   Id  System
命令(输入 m 获取帮助): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
分区号 (1-4, 默认 1): 1
起始 扇区 (2048-41943039, 默认为 2048):
将使用默认值 2048
Last 扇区, +扇区 or +size{K,M,G} (2048-41943039, 默认为 41943039): +5G
分区 1 已设置为 Linux 类型, 大小设为 5 GiB
命令(输入 m 获取帮助): p
磁盘 /dev/sdb: 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: 0x782338b8
  设备 Boot      Start          End      Blocks   Id  System
/dev/sdb1          2048      10487807       5242880    83  Linux
命令(输入 m 获取帮助): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
正在同步磁盘。
```

在硬盘中更新分区信息

```
[root@localhost ~]# partprobe
[root@localhost ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                8:0      0   10G  0 disk
├─sda1                            8:1      0   500M  0 part /boot
├─sda2                            8:2      0    9.5G  0 part
│   ├─centos-root                 253:0     0    8.5G  0 lvm  /
│   └─centos-swap                 253:1     0     1G  0 lvm  [SWAP]
sdb                                8:16     0   20G  0 disk
└─sdb1                            8:17     0     5G  0 part
loop0                             7:0      0  100G  0 loop
└─docker-253:0-50989499-pool     253:2     0  100G  0 dm
loop1                             7:1      0     2G  0 loop
└─docker-253:0-50989499-pool     253:2     0  100G  0 dm
[root@localhost ~]# mkdir /newFS
[root@localhost ~]# mount /dev/sdb1 /newFS/
[root@localhost ~]# df -h
文件系统      容量  已用  可用  已用% 挂载点
/dev/mapper/centos-root 8.5G  1.5G  7.1G   18% /
devtmpfs        903M      0  903M    0% /dev
tmpfs           913M      0  913M    0% /dev/shm
tmpfs           913M  8.6M  904M    1% /run
tmpfs           913M      0  913M    0% /sys/fs/cgroup
/dev/sda1       497M  123M  374M   25% /boot
tmpfs          183M      0  183M    0% /run/user/0
/dev/sdb1       5.0G   33M  5.0G    1% /newFS
[root@localhost ~]#
```


du命令

du命令

用于查看文件数据占用量的du命令，其格式为 “du [选项] [文件]”。

```
[root@localhost ~]# cp -rf /etc/* /newFS/
[root@localhost ~]# du -sh /newFS/
33M      /newFS/
[root@localhost ~]# vi /etc/fstab

#
# /etc/fstab
# Created by anaconda on Wed May 18 18:52:19 2016
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root /                    xfs      defaults        0 0
UUID=2f1fac57-2ddf-46d6-be6d-49b82acaa541 /boot    xfs
defaults          0 0
/dev/mapper/centos-swap swap      swap        defaults        0 0
/dev/sdb1/newFS    xfs      defaults 0 0
[root@localhost ~]#
```

添加交换分区

添加交换分区

使用fdisk添加一个2G大小的分区，然后使用mkswap命令将其变为swap分区

```
[root@localhost newFS]#mkswap /dev/sdb2
正在设置交换空间版本 1, 大小 = 5242876 KiB
无标签, UUID=fda1528f-7003-46f0-9e09-26aadd5a9155
[root@localhost newFS]# swapon /dev/sdb2
[root@localhost newFS]# free -m
total used free shared buff/cache available
Mem: 1824 169 1264 8 391 1466
Swap: 6143 0 6143

# 添加到开机启动
[root@localhost newFS]# vi /etc/fstab
/dev/mapper/centos-root / xfs defaults 0 0
UUID=2f1fac57-2ddf-46d6-be6d-49b82acaa541 /boot xfs
defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
/dev/sdb1/newFS xfs defaults 0 0
/dev/sdb2 swap swap defaults 0 0
[root@localhost newFS]#
```

磁盘容量配额

- 软限制：当达到软限制时会提示用户，但仍允许用户在限定的额度内继续使用。
- 硬限制：当达到硬限制时会提示用户，且强制终止用户的操作

开启磁盘配额功能

```
[root@localhost ~]# vi /etc/fstab

#
/dev/mapper/centos-root / xfs defaults 0 0
UUID=2f1fac57-2ddf-46d6-be6d-49b82acaa541 /boot xfs defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
/dev/sdb1 /newFS xfs defaults,quota 0 0
/dev/sdb2 swap swap defaults 0 0
[root@localhost ~]# xfs_quota -x -c 'limit bsoft=3m bhard=6m isoft=3 ihard=6 eagle' /newFS
[root@localhost ~]# xfs_quota -x -c report /newFS/
User quota on /newFS (/dev/sdb1)

          Blocks
User ID      Used      Soft      Hard      Warn/Grace
-----
root          33764         0         0      00 [-----]
eagle           0      3072      6144      00 [-----]
[eagle@localhost ~]$ dd if=/dev/zero of=/newFS/eagle bs=5M count=1
记录了1+0 的读入
记录了1+0 的写出
5242880字节(5.2 MB)已复制, 0.0116712 秒, 449 MB/秒
[eagle@localhost ~]$ dd if=/dev/zero of=/newFS/eagle bs=8M count=1
dd: 写入"/newFS/eagle" 出错: 超出磁盘限额
记录了1+0 的读入
记录了0+0 的写出
6291456字节(6.3 MB)已复制, 0.0141649 秒, 444 MB/秒
[eagle@localhost ~]$
```

edquota

edquota命令用于编辑用户的quota配额限制，格式为“edquota [参数] [用户]”。

```
[root@localhost eagle]# edquota -u eagle
Disk quotas for user eagle (uid 1000):
    Filesystem            blocks          soft          hard          inodes
    soft      hard
    /dev/sdb1            6144          3072          8192           1
    3           6
[eagle@localhost ~]$ dd if=/dev/zero of=/newFS/eagle bs=8M count=1
记录了1+0 的读入
记录了1+0 的写出
8388608字节(8.4 MB)已复制, 0.0225895 秒, 371 MB/秒
[eagle@localhost ~]$ dd if=/dev/zero of=/newFS/eagle bs=10M count=1
dd: 写入"/newFS/eagle" 出错: 超出磁盘限额
记录了1+0 的读入
记录了0+0 的写出
8388608字节(8.4 MB)已复制, 0.0193201 秒, 434 MB/秒
[eagle@localhost ~]$
```

软硬链接方式

软硬方式链接

- 硬链接 (hard link) :

可以将它理解为一个“指向原始文件inode的指针”，系统不为它分配独立的inode和文件。所以，硬链接文件与原始文件其实是同一个文件，只是名字不同。我们每添加一个硬链接，该文件的inode连接数就会增加1；而且只有当该文件的inode连接数为0时，才算彻底将它删除。换言之，由于硬链接实际上是指向原文件inode的指针，因此即便原始文件被删除，依然可以通过硬链接文件来访问。需要注意的是，由于技术的局限性，我们不能跨分区对目录文件进行链接。

- 软链接（也称为符号链接[symbolic link]）：

仅仅包含所链接文件的路径名，因此能链接目录文件，也可以跨越文件系统进行链接。但是，当原始文件被删除后，链接文件也将失效，从这一点上来说与Windows系统中的“快捷方式”具有一样的性质。

软硬方式链接参数

参数	作用
-s	创建“符号链接”（如果不带-s参数，则默认创建硬链接）
-f	强制创建文件或目录的链接
-i	覆盖前先询问
-v	显示创建链接的过程

