

## 16.挂载命令

### 挂载概念

Linux中的根目录以外的文件要想被访问，需要将其“关联”到根目录下的某个目录来实现，这种关联操作就是“**挂载**”，这个目录就是“**挂载点**”，解除此关联关系的过程称之为“**卸载**”。

注意：“挂载点”的目录需要以下几个要求：

- (1) 目录事先存在，可以用mkdir命令新建目录；
- (2) 挂载点目录不可被其他进程使用到；
- (3) 挂载点下原有文件将被隐藏。挂载点下最好是空的！

### 1. 挂载与卸载的基本使用方法

**mount命令的两个常用用法：**

```
# mount [-l]
#查询系统中已经挂载的设备，-l 会显示卷标名称，也可以直接查看/etc/mtab文件
# mount -a （前面用过一次了）把fstab里的命令执行一次，当然如果有配置错误，
也会报错，所以可以起到一定的检查作用
#依据配置文件/etc/fstab 的内容，自动挂载
```

**挂载的基本格式：** # mount [-t 文件系统] [-L 卷标名] [-o 特殊选项] 设备文件名  
挂载点

**选项：**

-t 文件系统： 加入文件系统类型来指定挂载的类型，可以 ext3、ext4、iso9660、xfs、fat32、ntfs  
等文件系统。

-L 卷标名： 挂载指定卷标的分区，而不是安装设备文件名挂载

-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，如果不指定它有一组默认值会生效，一般情况下就使用默认值，不建议更改。

**-o options：**（挂载文件系统的选项，这些特殊选项，了解一下就ok，一般情况下是不建议修改的，就defaults就好！切记！）

**async：** 异步模式；

**sync：** 同步模式；

**atime/noatime：** 包含目录和文件；

**diratime/nodiratime：** 目录的访问时间戳

**auto/noauto：** 是否支持自动挂载

**exec/noexec：** 是否支持将文件系统上应用程序运行进程

**dev/noddev**: 是否支持在此文件系统上使用设备文件;  
**suid/nosuid**: 是否支持在此文件系统上使用特殊权限  
**remount**: 重新挂载  
**ro**: 只读  
**rw**: 读写  
**user/nouser**: 是否允许普通用户挂载此设备  
**acl**: 启用此文件系统上的acl功能

注意: 上述选项可多个同时使用, 彼此使用逗号分隔;

默认挂载选项, 也就是fstab文件中defaults对应的值: **rw, suid, dev, exec, auto, nouser, async**

## 卸载命令: **umount**

命令使用格式:

**# umount 设备文件**

**# umount 挂载点**

上面两个是绑定关系, 所有用那个都一样

## 2. 光盘挂载

光盘挂载的前提依然是指定光盘的设备文件名, 不同版本的 Linux, 设备文件名并不相同:

CentOS 5.x 以前的系统, 光盘设备文件名是/dev/hdc

CentOS 6.x 以后的系统, 光盘设备文件名是/dev/sr0

不论哪个系统都有软连接/dev/cdrom, 与可以作为光盘的设备文件名

```
[root@localhost ~]# mount -t iso9660 /dev/cdrom /mnt #挂载光盘
```

```
[root@localhost ~]# umount /mnt/cdrom
```

注意1: 卸载的时候需要退出光盘目录, 才能正常卸载

注意2: 挂载点一定要是空目录

## 3. 挂载 U 盘 (FAT32)

U 盘会和硬盘共用设备文件名, 所以 U 盘的设备文件名不是固定的, 需要手工查询, 查询命令:

```
[root@localhost ~]# fdisk -l #查询硬盘
```

挂载命令:

```
[root@localhost ~]# mount -t vfat /dev/sdb1 /mnt/usb/
```

#挂载 U 盘。因为是 Windows 分区, 所以是 vfat 文件系统格式

如果 U 盘中有中文, 会发现中文是乱码。Linux 要想正常显示中文, 需要两个条件:

- 安装了中文编码和中文字体
- 操作终端需要支持中文显示 (纯字符终端, 是不支持中文编码的)

而我们当前系统是安装了中文编码和字体，而 crt 远程终端是 Windows 下的程序，当然是支持中文显示的。那之所以挂载 U 盘还出现乱码，是需要在挂载的时候，手工指定中文编码，例如：

```
[root@localhost ~]# mount -t vfat -o iocharset=utf8 /dev/sdb1  
/mnt/usb/
```

卸载

```
[root@localhost ~]# umount /mnt/usb/
```

## 4、挂载 NTFS 分区（U盘或在移动硬盘的挂载）

4.1 Linux 中硬件和 windows 一样也是需要驱动程序的，Linux 的驱动程序有这样几种：

驱动直接放入系统内核之中。这种驱动主要是系统启动加载必须的驱动，数量很少。

驱动以函数库的形式放入硬盘。大多数驱动都已这种方式保存，保存位置

在/lib/modules/3.10.0-862.el7.x86\_64/kernel/中。这里的東西都是Linux的核心模块

arch：与硬件平台有关的项目，例如 CPU 的等级等等；

crypto：核心所支持的加密的技术，例如 md5 或者是 des 等等；

drivers：一些硬件的驱动程序，例如显示适配器、网络卡、PCI 相关硬件等等；

fs：核心所支持的 filesystems，例如 vfat, reiserfs, nfs 等等；

lib：一些函数库；

net：与网络有关的各项协议数据，还有防火墙模块(net/ipv4/netfilter/\*) 等等；

sound：与音效有关的各项模块；

驱动可以被 Linux 识别，但是系统认为这种驱动一般不常用，默认不加载。如果需要加载这种驱动，需要重新编译内核，而 NTFS 文件系统的驱动就属于这种情况。

硬件不能被 Linux 内核识别，需要手工安装驱动。当然前提是厂商提供了该硬件针对 Linux 的驱动，否则就需要自己开发驱动了

### 4.2 使用

NTFS-3G 安装 NTFS 文件系统模块

NTFS-3G 是一个开源项目，NTFS-3G 是为 Linux, Android, Mac OS X, FreeBSD, NetBSD, OpenSolaris, QNX, Haiku, 和其他操作系统提供的一个稳定的，功能齐全，读写 NTFS 的驱动程序。

下载 NTFS-3G 插件

我们从网站 <https://www.tuxera.com/community/open-source-ntfs-3g/> 下载 NTFS-3G 插件到 Linux 服务器上。

安装 NTFS-3G 插件

在编译安装 NTFS-3G 插件之前，要保证 gcc 编译器已经安装 `yum install -y gcc`。具

体安装命令如下：

```
[root@localhost ~]# tar -zxvf ntfs-3g_ntfsprogs-2017.3.23.tgz
[root@localhost ~]# cd ntfs-3g_ntfsprogs-2017.3.23
[root@localhost ntfs-3g_ntfsprogs-2017.3.23]# ./configure    #编译器准备。没有指定安装目录，安装到默认位置中
[root@localhost ntfs-3g_ntfsprogs-2017.3.23]# make    #编译
[root@localhost ntfs-3g_ntfsprogs-2017.3.23]# make install    #编译安装
安装就完成了，已经可以挂载和使用 Windows 的 NTFS 分区了。不过需要注意挂载分区时的文件系统不是 ntfs，而是 ntfs-3g。
[root@localhost ~]# mount -t ntfs-3g /dev/sdb1 /mnt/win
```

**如果发现有中文乱码**，可能是语言环境就有关系，我们可以临时的更改一下语言环境，两条命令：

\$ locale 看一些系统运行时的语言环境

\$ export LC\_ALL="zh\_CN.GBK" 知道这句话的作用是：临时设定编码就行

export命令作用：新增、修改、删除环境变量，这个我们在后面linux系统的启动管理中详细讲解！

Linux运行时的语言环境，它包括语言(Language)，地域(Territory)和字符集(Codeset)。

一个locale的书写格式为：语言[\_地域[.字符集]]。完全的locale表达方式是 [语言[\_地域][.字符集] [@修正值]]。

zh\_CN.GB2312 = 中文\_中华人民共和国+国标2312字符集。

**locale：查看和临时设置语言环境的命令**

**#locale**

LANG=zh\_CN.UTF-8

LC\_CTYPE="zh\_CN.UTF-8"

LC\_NUMERIC="zh\_CN.UTF-8"

LC\_TIME="zh\_CN.UTF-8"

LC\_COLLATE="zh\_CN.UTF-8"

LC\_MONETARY="zh\_CN.UTF-8"

LC\_MESSAGES="zh\_CN.UTF-8"

LC\_PAPER="zh\_CN.UTF-8"

LC\_NAME="zh\_CN.UTF-8"

LC\_ADDRESS="zh\_CN.UTF-8"

LC\_TELEPHONE="zh\_CN.UTF-8"

LC\_MEASUREMENT="zh\_CN.UTF-8"

LC\_IDENTIFICATION="zh\_CN.UTF-8"

LC\_ALL=

locale把按照所涉及到的文化传统的各个方面分成12个大类，这12个大类分别是：

- 1、字符输入(LC\_CTYPE)
- 2、数字(LC\_NUMERIC)
- 3、比较和排序习惯(LC\_COLLATE)
- 4、时间显示格式(LC\_TIME)
- 5、货币单位(LC\_MONETARY)
- 6、信息：主要是提示信息, 错误信息, 状态信息, 标题, 标签, 按钮和菜单等(LC\_MESSAGES)
- 7、姓名书写方式(LC\_NAME)
- 8、地址书写方式(LC\_ADDRESS)
- 9、电话号码书写方式(LC\_TELEPHONE)
- 10、度量衡表达方式 (LC\_MEASUREMENT)
- 11、默认纸张尺寸大小(LC\_PAPER)
- 12、对locale自身包含信息的概述(LC\_IDENTIFICATION)

locale的设定：

LC\_ALL和LANG优先级的关系：LC\_ALL > LC\_\* > LANG

修改文件(debian系列)：/etc/default/locale

修改文件(redhat系列)：/etc/sysconfig/i18n (CentOS6版本的位置)

/etc/locale.conf (CentOS7改到了这里)

- 1、如果需要一个纯中文的系统的话，设定LC\_ALL=zh\_CN.XXXX，或者LANG=zh\_CN.XXXX都可以。
- 2、如果只想要一个可以输入中文的环境，而保持菜单、标题，系统信息等等为英文界面，那么只需要设定 LC\_CTYPE=zh\_CN.XXXX，LANG=en\_US.XXXX就可以了。
- 3、假如什么也不做的话，也就是LC\_ALL，LC\_\*和LANG均不指定特定值的话，系统将采用POSIX作为locale，也就是C locale。C 标准库系列之locale.h

### 总结一下：

设定了LC\_ALL就不用设置其他的了，他会强制覆盖掉LC\_\*. 设置部分LC\_\*和LANG的话，没有设置的LC\_\*采用的是LANG的值。

- 1、临时设定编码：\$ export LC\_ALL="zh\_CN.GBK"
- 2、永久设定编码：修改文件/etc/bashrc (对所有用户有效) 或者 ~/.bashrc (指定的用户有效)。在文件尾加入export LC\_ALL=zh\_CN.GBK
- 3、永久设定编码：/etc/locale.conf在这个配置文件中设定

## 5. 开机自动挂载里边的一些知识点

如果我们想实现开机自动挂载某设备，只要修改/etc/fstab文件即可。

## 要挂载的设备或伪文件系统

统类型    挂载选项    转储频率    自检次序

**要挂载的设备或伪文件系统：**设备文件、LABEL(LABEL="")、UUID(UUID="")、伪文件系统名称(proc, sysfs)

### 挂载选项: defaults

0: 不做备份

## 2: 每隔一天转储

0: 不自检

**文件系统**是对文件存储设备的空间进行组织和分配，负责文件存储并对存入的文件进行保护和检索的系统。具体地说，它负责为用户建立文件，存入、读出、修改、转储文件，控制文件的存取，当用户不再使用时撤销文件等。

## 6. /etc/fstab和/etc/mtab的区别

记录了计算机上硬盘分区的相关信息，启动 Linux 的时候，检查分区的 fsck 命令，和挂载分区的 mount 命令，都需要 fstab 中的信息，来正确的检查和挂载硬盘。

记载的是现在系统已经装载的文件系统，包括操作系统建立的虚拟文件等；而/etc/fstab是系统准备装载的。每当 mount 挂载分区、umount 卸载分区，都会动态更新 mtab，mtab 总是保持着当前系统中已挂载的分区信息，fdisk、df 这类程序，必须要读取 mtab 文件，才能获得当前系统中的分区挂载情况。当然我们自己还可以通过读取/proc/mount也可以来获取当前挂载信息

