

Linux 101：分区、备份与批处理（2 上）

崔 灏

Linux User Group

2018 年 6 月 3 日

磁盘分区和文件系统（回顾）

- ▶ fdisk / parted 编辑分区表

```
$ parted /dev/sdz
```

```
(parted) mktable gpt
```

```
(parted) mkpart DataPart ext4 1MiB 100GiB
```

```
(parted) mkpart BackupPart xfs 100GiB 100%
```

- ▶ mkfs 创建文件系统

```
$ mkfs.ext4 /dev/sdz1
```

```
$ mkfs.xfs /dev/sdz2
```

- ▶ 检测分区表变化

```
$ partprobe
```

- ▶ mount 挂载文件系统

```
$ mount -t ext4 /dev/sdz1 /mnt/data
```

```
$ mount -t xfs /dev/sdz2 /mnt/backup
```

fstab 自动挂载

手动 mount 的文件系统，重启后就失效了。

非移动存储可以通过 /etc/fstab 配置自动挂载：

# Device	Path	FS-Type	Options	Dump	Pass
/dev/sda1	/	ext4	defaults	0	1
/dev/sda2	/home	xfs	defaults	0	1
/dev/sda3	none	swap	defaults	0	0

相当于开机时执行：

```
mount -t <FS-Type> -o <Options> <Device> <Path>
```

pass 不为 0 的，开机自动进行磁盘检查。

fstab 自动挂载

实际上一般不直接用 `/dev/sdXn` 设备名，因为编号可能会因为硬件配置变化而发生变化，甚至重启都可能改变其他设备标识：

- ▶ UUID：文件系统的随机编号（格式化时设置）
- ▶ LABEL：文件系统卷标（格式化时设置）
- ▶ PARTUUID：GPT 分区表的分区随机编号（创建分区时设置）

fstab 自动挂载

查看设备标识

```
% sudo blkid /dev/sda1
/dev/sda1: UUID="e2bf4816-1d85-484c-ab2b-e65b153386b5"
        TYPE="ext4" PARTUUID="8201755d-8c93-407a-8724-
        af59e60088f8"
```

配置 fstab:

```
UUID=e2bf4816-1d85-484c-ab2b-e65b153386b5    /    ext4
defaults    0 1
```

fstab 自动挂载

更多 fstab 例子：

备份盘

LABEL=Backup	/backup	xfs	defaults	0	1
--------------	---------	-----	----------	---	---

Windows 数据盘，只读挂载

LABEL=WinData	/mnt/windata	ntfs-3g	ro	0	0
---------------	--------------	---------	----	---	---

NFS 网络文件系统

10.23.45.67:/share	/mnt/share	nfs	_netdev	0	0
--------------------	------------	-----	---------	---	---

文件系统故障

Something went wrong.

常见原因：

- ▶ 硬盘物理损坏/坏道
- ▶ Linux 文件系统 BUG
- ▶ 危险操作处理失误

一些故障表现：

- ▶ 数据丢失
- ▶ 内核错误 (dmesg)
- ▶ 操作文件出现 I/O Error
- ▶ IO 性能下降

文件系统故障

Don't panic.

一般原则：

- ▶ 最好离线 + 只读操作
- ▶ 先备份未损坏的重要数据
 - ▶ ddrescue：全盘/分区备份
 - ▶ rsync：文件备份

文件系统检查/修复：

- ▶ `fsck/fsck.xfs/fsck.vfat -a` 自动检查和修复文件系统

磁盘检查：

- ▶ `smartctl`：检查磁盘 SMART 信息
- ▶ `badblock`：检查磁盘坏道

备份文件

Before it's too late.

解决方案：

- ▶ cp 到另一块盘上（注意处理权限等元数据）
- ▶ tar 打包后传到另一台服务器或云端
- ▶ rsync 支持增量传输的文件同步工具！

为什么用 rsync：

- ▶ 增量备份：只对上次备份后更新的文件进行备份，节约 IO 和流量（“断点续传”）
- ▶ 保留文件元数据，如权限、修改时间等

rsync 增量备份

基本格式: `rsync [OPTIONS] SRC DEST`

- ▶ SRC、DEST 可以都是本地文件路径
- ▶ SRC、DEST 其中一个可以是远程服务器 (SSH)
- ▶ 常用选项:
 - ▶ `-r` 递归同步文件夹 (不保留权限)
 - ▶ `-a` 递归同步, 保留权限、软链接、时间戳等
 - ▶ `-aHAXS` 保持更多文件属性, 适合全盘备份
 - ▶ `-x` 不同步挂载的文件系统
 - ▶ `-v --progress` 显示进度

rsync 增量备份

备份 /docs/ (注意路径的斜杠):

```
$ rsync -aHAXS /docs/ /mnt/backup/docs-201806/
```

通过 SSH 备份到远程服务器:

```
$ rsync -aHAXS /docs/ username@12.34.56.78:docs/
```

试试 Ctrl-C 中止后重新传。

文件批处理

find

find 文件查找: find < 文件夹 > [断言]

- ▶ 根据文件名查找 (后缀名 jpg, 忽略大小写)
`$ find img/ -iname '*.jpg'`
- ▶ 根据修改日期 (10 天以前)
`$ find data/ -mtime +10`
- ▶ 根据文件类型 (名为 tmp 的文件夹)
`$ find data/ -type d -name "tmp"`
- ▶ 根据用户归属 (不属于 cuihao)
`$ find data/ -not -user cuihao`

文件批处理

find

对查找到的文件批量执行命令：

- ▶ find 内置操作（删除修改时间在 100 天的 JPEG 文件）
`$ find img/ -iname '*.jpg' -mtime +100 -delete`
- ▶ 自定义命令（把 txt 文件都修改成 UNIX 换行符）
`$ find data/ -iname '*.txt' -exec dos2unix {}`

文件批处理

xargs

xargs 从 stdin 读取参数，批量调用命令执行

- ▶ 删除修改时间在 100 天的 JPEG 文件

```
$ find img/ -iname '*.jpg' -mtime +100 | xargs rm
```

- ▶ 把 txt 文件都修改成 UNIX 换行符

```
$ find data/ -iname '*.txt' | xargs dos2unix
```

注意 rm、dos2unix 都可以接受多个参数，xargs 读取足够多的参数后一次性执行。

一般比 find -exec 效率更高。

结束

Q & A