

ext2/ext3 文件系统管理

内容提要

1. 熟悉 ext2/ext3 文件系统管理工具
2. 学会使用 mke2fs 创建 ext2/ext3 文件系统
3. 学会使用 e2fsck 检查 ext2/ext3 文件系统
4. 学会使用 tune2fs 调整 ext2/ext3 文件系统的属性

ext2/ext3 文件系统管理工具

在 e2fsprogs [<http://www.proxyserve.net/index.php?q=aHR0cDovL2UyZnNwcm9ncy5zb3VyY2Vmb3JnZS5uZXQ%3D>] 软件包中提供了 ext2/ext3 文件系统管理工具。下面列出常用工具的说明：

工具	说明
/sbin/fsck	文件系统检查的前端工具
/sbin/e2fsck	检查和修复 ext2 或 ext3 文件系统
/sbin/fsck.ext2	检查和修复 ext2 文件系统
/sbin/fsck.ext3	检查和修复 ext3 文件系统
/sbin/mke2fs	创建 ext2 或 ext3 文件系统
/sbin/mkfs.ext2	创建 ext2 文件系统
/sbin/mkfs.ext3	创建 ext3 文件系统
/sbin/badblocks	检查磁盘分区坏块
/sbin/tune2fs	调整 ext2/ext3 文件系统的可调属性参数
/sbin/dumpe2fs	显示 ext2/ext3 文件系统的超级块和块组信息
/sbin/debugfs	ext2/ext3 文件系统调试器
/sbin/e2label	显示或者修改 ext2/ext3 文件系统的卷标
/sbin/findfs	根据 ext2/ext3 文件系统的卷标或 UUID（全局唯一标识符，Universally Unique Identifier）查找对应的设备
/sbin/resize2fs	更改 ext2/ext3 文件系统的容量

创建 ext2/ext3 文件系统

mke2fs [<http://www.proxyserve.net/index.php?q=aHR0cDovL21hbi5jeC9ta2UyZnM%3D>] 命令用于创建 ext2/ext3 文件系统。mkfs.ext2 和 mkfs.ext3 命令都是 mke2fs 的硬链接，当使用 man mkfs.ext2 和 man mkfs.ext3 命令查看手册页时都定向到 mke2fs。

mke2fs 命令的格式如下：

```
格式1: mke2fs [<选项>...] <设备名> [blocks-count]
格式2: mke2fs -j [<选项>...] <设备名> [blocks-count]
```

说明：

- 格式1用于创建 ext2 文件系统；格式2用于创建 ext3 日志文件系统。

- **blocks-count**用于指定要创建的文件系统的块数，此值应该小于 **fdisk** 命令查看的此分区或逻辑卷的块数，若省略此参数将使用整个分区或逻辑卷创建文件系统。
- 常用选项说明如下：

选项	说明
-b <block-size>	指定区块大小，单位为字节。
-c	在创建文件系统之前检查是否有损坏的区块。一个c做只读检查；两个c做读写检查。
-f <fragment-size>	指定不连续区段的大小，单位为字节。
-F	强制执行 mke2fs ，不建议使用。
-i <bytes-per-inode>	指定 “字节/inode” 的比例。
-N <number-of-inodes>	指定要建立的inode数目。
-j	建立一个ext3日志文件系统。
-J <journal-options>	指定 ext3 日志文件系统的参数
-l <filename>	从指定的文件中，读取文件系统中损坏区块的信息。
-L <volume-label>	设置文件系统的卷标（最长11个字符）。
-m <reserved-blocks-percentage>	指定给管理员保留区块的比例，预设为5%。
-M <last-mounted-directory>	记录最后一次挂装的目录。
-q	执行时不显示任何信息。
-v	执行时显示详细信息。
-V	显示 mke2fs 版本信息。

使用举例：

1、在系统第二块 SCSI 接口的硬盘的第5个分区上创建 ext2 类型的文件系统

```
# mke2fs /dev/sdb5
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
244320 inodes, 487974 blocks
24398 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=503316480
15 block groups
32768 blocks per group, 32768 fragments per group
16288 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 33 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

2、在系统第二块 SCSI接口的硬盘的第6个分区上创建 ext3 类型的文件系统，检查磁盘坏块，卷标为设置“data”

```
# mke2fs -j -c -L data /dev/sdb6
mke2fs 1.39 (29-May-2006)
Filesystem label=data
OS type: Linux
Block size=4096 (log=2)
```

```
Fragment size=4096 (log=2)
123648 inodes, 246991 blocks
12349 blocks (5.00%) reserved for the super user
First data block=0
8 block groups
32768 blocks per group, 32768 fragments per group
15456 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376
Checking for bad blocks (read-only test): done          991
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 27 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

3、在系统第三块 **SCSI**接口的硬盘的第1个分区上创建 **ext3** 类型的文件系统，检查磁盘坏块，保留 **3%** 的空间

```
# mke2fs -j -c -m 3 /dev/sdc1
```

4、对软盘创建 **ext2** 类型的文件系统，同时检查磁盘坏块

```
# mke2fs -c -T floppy /dev/fd0
```

检查 ext2/ext3 文件系统

e2fsck 命令用于检查 **ext2/ext3** 文件系统的正确性并进行修复。**fsck.ext2** 和 **fsck.ext3** 命令都是 **e2fsck** 的硬链接，当使用 **man fsck.ext2** 和 **man fsck.ext3** 命令查看手册页时都定向到 **e2fsck**。

e2fsck 命令的格式如下：

```
e2fsck [<选项>] <设备名>
```

选项	说明
-a 或 -p	不询问使用者意见，便自动修复文件系统，建议使用 -p
-b <superblock>	指定超级块，而不使用预设的 superblock ，用于当前文件系统超级块损坏时
-B <blocksize>	指定区块的大小，单位为字节
-c	一并执行 badblocks ，以标识损坏的区块
-C <fd>	将检查过程的信息完整记录在 file descriptor 中，使得整个检查过程都能完整监控
-d	显示排错信息
-f	强制对该文件系统进行全面检查
-F	执行前先清除设备的缓冲区
-j <ext-journal>	为 ext3 日志文件系统指定日志路径
-l <filename>	将文件中指定的区块加到损坏区块列表
-L <filename>	先清除损坏区块列表，再将文件中指定的区块加到损坏区块列表
-n	以只读模式开启文件系统，并采取非互动方式执行，所有的问题对话均设置以“no”回答
-y	采取非互动方式执行，所有的问题均设置以“yes”回答。
-s	如果文件系统的字节顺序不适当，就交换字节顺序，否则不做任何动作

-S	不管文件系统的字节顺序，一律交换字节顺序
-t	显示执行顺序信息
-v	执行时显示详细的信息
-V	显示 e2fsck 版本信息

选项 **-p/-y/-n** 应该互斥使用。

在实际的文件系统修复经验中，我遇到了两难的问题。根文件系统损坏了，需要修复，系统会提示你要用**e2fsck**来修复。当然你可以按照他的提示去做，但是很可能，你会发现，等你修复完了，文件系统也能正常工作了，但是往往最重要的文件被修复得不加了。这不是开玩笑，而是有大量的案例。如何在修复和损坏之间取得一种平衡需要根据具体的情况来定夺。

e2fsck 执行后的传回值及代表意义如下：

- 0 — 没有任何错误发生。
- 1 — 文件系统发生错误，并且已经修正。
- 2 — 文件系统发生错误，并且已经修正。
- 4 — 文件系统发生错误，但没有修正。
- 8 — 运作时发生错误。
- 16 — 使用的语法发生错误。
- 128 — 共享的函数库发生错误。
- 一般情况下，无需用户手动执行 **e2fsck** 命令。在系统启动过程中，一旦系统检测到了不一致就会自动运行 **e2fsck** 命令。
- 手动执行 **e2fsck** 命令，建议在单用户模式下且文件系统被卸装的情况下进行。

使用举例：

1、自动检查并修复文件系统

```
# e2fsck -p /dev/sda5
```

2、强行进行文件系统检查，标识损坏区块，对所有问题回答“yes”

```
# e2fsck -f -y /dev/hda2
```

调整 ext2/ext3 文件系统的属性

可以使用 **tune2fs** [<http://www.proxyserve.net/index.php?q=aHR0cDovL21hbi5jeC90dW5lMmZz>] 命令调整 **ext2/ext3** 文件系统的可调属性参数。

tune2fs 命令的格式如下：

格式1: **tune2fs -l** <设备名>
格式2: **tune2fs** [**<选项>**] <设备名>

格式1：用于显示 **ext2/ext3** 文件系统的属性；格式2：用于条整 **ext2/ext3** 文件系统的属性。

常用选项说明如下：

选项	说明
-c <max-mount-counts>	设置每两次进行文件系统完整性检查的最大挂装次数, -1 表示从不进行完整性检查

-C <mount-count>	设置当前文件系统已经挂装的次数
-i <interval-between-checks>	设置每两次进行文件系统完整性检查的时间间隔，可以使用时间单位后缀（d-天，m-月，w-周），0 表示从不进行完整性检查
-T <time-last-checked>	设置最后一次进行文件系统完整性检查的时间，（YYYYMMDD[HH[MM[SS]]] 或 now）
-j	为 ext2 文件系统添加 ext3 文件系统所需的日志（即：将 ext2 文件系统无损地转换为 ext3 文件系统）
-l	列出指定设备的超级块信息
-L <volume-label>	为指定设备设置卷标，不大于 16 字符
-m <reserved-blocks-percentage>	指定给管理员保留区块的比例，预设为 5%
-r <reserved-blocks-count>	指定给管理员保留区块的数量

为了保证文件系统的完整性，默认情况下系统每隔**180**天或文件系统挂装**27**次之后就要在挂装文件系统时调用 **e2fsck** 对文件系统进行一次文件系统完整性检查，这对于工作站及服务器是很有必要的。管理员可以用 **tune2fs** 工具对此进行调整。

使用举例：

1、查看指定文件系统的相关信息

```
# tune2fs -l /dev/sdb5
tune2fs 1.39 (29-May-2006)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         1dbcff1a-e408-42fb-a465-114cae993873
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      filetype sparse_super
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              123648
Block count:              246991
Reserved block count:     12349
Free blocks:              243096
Free inodes:              123637
First block:              0
Block size:               4096
Fragment size:            4096
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         15456
Inode blocks per group:    483
Filesystem created:       Fri Mar 23 23:45:47 2007
Last mount time:          n/a
Last write time:          Fri Mar 23 23:45:48 2007
Mount count:              0
Maximum mount count:      27
Last checked:             Fri Mar 23 23:45:47 2007
Check interval:           15552000 (6 months)
Next check after:         Wed Sep 19 23:45:47 2007
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
Default directory hash:   tea
Directory Hash Seed:      a301fd0e-0841-45a5-a1a6-ccd5ff82ba5a
```

2、将 ext2 文件系统转换成 ext3 文件系统

```
# tune2fs -j /dev/sdb5
tune2fs 1.39 (29-May-2006)
Creating journal inode: done
This filesystem will be automatically checked every 27 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

再来看看文件系统信息的变化:

```
# tune2fs -l /dev/sdb5
tune2fs 1.39 (29-May-2006)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         1dbcff1a-e408-42fb-a465-114cae993873
Filesystem magic number: 0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal filetype sparse_super
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              123648
Block count:              246991
Reserved block count:     12349
Free blocks:              238994
Free inodes:              123637
First block:              0
Block size:               4096
Fragment size:            4096
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         15456
Inode blocks per group:   483
Filesystem created:       Fri Mar 23 23:45:47 2007
Last mount time:          n/a
Last write time:          Sat Mar 24 00:38:58 2007
Mount count:              0
Maximum mount count:      27
Last checked:             Fri Mar 23 23:45:47 2007
Check interval:           15552000 (6 months)
Next check after:         Wed Sep 19 23:45:47 2007
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               128
Journal inode:            8
Default directory hash:   tea
Directory Hash Seed:      a301fd0e-0841-45a5-a1a6-ccd5ff82ba5a
Journal backup:           inode blocks
```

3、调整指定文件系统每两次完整性检查的装载次数为10次

```
# tune2fs -c 10 /dev/sdb5
```

```
tune2fs 1.39 (29-May-2006)
Setting maximal mount count to 10
```

4、调整指定文件系统每两次完整性检查的时间间隔为2个月

```
# tune2fs -i 2m /dev/sdb5
tune2fs 1.39 (29-May-2006)
Setting interval between checks to 5184000 seconds
```

5、调整指定文件系统的保留空间为 2%

```
# tune2fs -m 2 /dev/sdb5
tune2fs 1.39 (29-May-2006)
Setting reserved blocks percentage to 2% (4939 blocks)
```

若要一个文件系统在下次挂装时进行文件系统完整性检查，可以使用带 **-C** 的命令设置已挂装次数的值大于 **-c** 指定的值。如：

```
# tune2fs -l /dev/sdb5 | grep 'Maximum mount count'
```

指定 **-C** 的值大于此值即可在下次挂装（通常是重新启动）时进行文件系统完整性检查

```
# tune2fs -C 28 /dev/sdb5
```

管理 ext2/ext3 文件系统的卷标

e2label

e2label [<http://www.proxyserve.net/index.php?q=aHR0cDovL21hbi5jeC9IMmxhYmVs>] 命令用来设定或显示 **ext2** 或 **ext3** 分区的卷标。命令格式如下：

```
e2label <设备文件名> [<新的卷标名称>]
```

例如：

1、显示 /dev/sdb6 文件系统的卷标

```
# e2label /dev/sdb6
```

2、将 /dev/sdb5 文件系统的卷标设置为 **www-data**

```
# e2label /dev/sdb5 www-data
```

findfs

findfs [<http://www.proxyservice.net/index.php?q=aHR0cDovL21hbi5jeC9maW5kZnM%3D>] 命令可以在系统中查找指定卷标所对应的文件系统。命令格式如下：

```
findfs LABEL=<label>
```

例如，要查找卷标为 **data** 所对应的文件系统，可以使用如下命令：

```
# findfs LABEL=data
```

- 显示源文件
- 登录