

Linux培训系列

第四讲

将通过讨论多个主题（包括 Linux 文件系统、Linux 引导过程、运行级别、文件系统限额和系统日志），来巩固您在重要的 Linux 管理技能方面的知识。

内容基础，语言简短简洁

红联Linux论坛是致力于Linux技术讨论的站点，目前网站收录的文章及教程基本能满足不同水平的朋友学习。

红联Linux门户：www.linux110.com

红联Linux论坛：www.linuxdiyf.com/bbs

下载:Linux电子书籍：

<http://www.linux286.com/linux/linuxdzsj.htm>

目录

Linux 文件系统

- Linux 文件系统 - 块设备
- 完整的磁盘和分区
- 介绍 fdisk
- 使用 fdisk
- 使用 fdisk (续)
- fdisk 以及更多内容
- 创建文件系统
- 挂装文件系统
- 创建和使用交换分区
- 查看挂装的文件系统
- 挂装选项
- 介绍 fstab
- 样本 fstab
- 样本 fstab (续)
- 卸装文件系统
- 介绍 fsck
- fsck 的问题
- ext2 文件系统
- ext3 文件系统
- ReiserFS 文件系统
- XFS 文件系统
- JFS 文件系统
- VFAT

引导系统

- 引导系统
- MBR
- 内核引导过程
- /sbin/init
- 深入研究：LILO
- 使用 LILO
- 一个重要的 LILO 问题
- 深入研究：GRUB
- 使用 GRUB
- dmesg
- /var/log/messages
- 单用户方式
- 使用单用户方式
- 更改运行级别
- 恰当地关机
- 立即关机
- 缺省运行级别

运行级别

- 运行级别 - 单用户方式
- 单用户方式 (续)
- 运行级别
- telinit
- 运行级别规则

[“now”和停机](#)

[配置init](#)

文件系统限额

[文件系统限额](#)

[内核支持](#)

[文件系统支持](#)

[配置限额](#)

[配置限额（续）](#)

[quota 命令](#)

[查看限额](#)

[edquota](#)

[edquota（续）](#)

[理解edquota](#)

[进行更改](#)

[复制限额](#)

[组限制](#)

[repquota 命令](#)

[repquota 选项](#)

[监控限额](#)

[修改宽限时间](#)

[在引导时检查限额](#)

系统日志

[系统日志 - 介绍 syslogd](#)

[读取日志](#)

[跟踪日志文件](#)

[查找日志](#)

[日志概述](#)

[syslog.conf](#)

[syslog.conf](#)

[重新装入和附加信息](#)

[安全性注释](#)

[logrotate](#)

[高级主题 — klogd](#)

[高级主题 — 替代日志记录器](#)

Linux资源汇集

[海量Linux技术文章](#)

Linux 文件系统

Linux 文件系统 - 块设备

发布时间 :2007-01-27 11:22:39

在本节中，我们将仔细研究 Linux 文件系统，以便您熟悉管理员需要知道的所有具体的详细资料。我将从介绍“块设备”开始。最为人们熟知的块设备可能是表示 Linux 系统中第一个 IDE 驱动器的块设备：

`/dev/hda`

如果您的系统使用 SCSI 驱动器，那么您的第一个硬盘驱动器将是：

`/dev/sda`

以上的块设备表示磁盘的抽象接口。用户程序可以使用这些块设备与磁盘进行交互而无需担心驱动器是 IDE、SCSI 或其它东西。程序只需将磁盘上的存储信息视为一串大小为 512 字节、连续的、可随机访问的块来进行寻址。

完整的磁盘和分区

发布时间 :2007-01-27 11:23:08

在 Linux 下，我们可以使用特殊的“mkfs”命令并将特殊块设备指定为命令行参数来创建文件系统。

然而，尽管理论上可以使用一个象 /dev/hda 或 /dev/sda 那样的“完整的磁盘”的块设备（表示整个磁盘）来包含单个文件系统，但实际上从未采取过这种方法。而是将整个磁盘块设备分成更小、更可管理的、名为“分区”的块设备。可以使用名为 fdisk 的工具创建分区，该工具用来创建和编辑存储在磁盘上的分区表。分区表对如何分割整个磁盘进行了精确定义。

介绍 fdisk

发布时间 :2007-01-27 11:23:45

为了研究磁盘的分区表，我们可以运行 fdisk，其中将表示整个磁盘的块设备指定为参数：

```
# fdisk /dev/hda
```

```
# fdisk /dev/sda
```

请注意：如果磁盘的任何分区包含正在使用的文件系统或包含重要的数据，则不应该保存或作出任何对磁盘分区的更改。那样做会导致磁盘上的数据丢失。

使用 fdisk

发布时间 :2007-01-27 11:24:28

在 fdisk 中，您将看到与下面相似的提示符：

Command (m for help):

输入 p 以显示磁盘的当前分区配置：

Command (m for help): p

Disk /dev/hda: 240 heads, 63 sectors, 2184 cylinders
Units = cylinders of 15120 * 512 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	14	105808+	83	Linux
/dev/hda2		15	49	264600	82	Linux swap
/dev/hda3		50	70	158760	83	Linux
/dev/hda4		71	2184	15981840	5	Extended
/dev/hda5		71	209	1050808+	83	Linux
/dev/hda6		210	348	1050808+	83	Linux
/dev/hda7		349	626	2101648+	83	Linux
/dev/hda8		627	904	2101648+	83	Linux
/dev/hda9		905	2184	9676768+	83	Linux

Command (m for help):

我们将这个特殊磁盘配置成包含七个 Linux 文件系统（以“Linux”列出）和一个交换分区（以“Linux swap”列出）。请注意左边相应的分区块设备名称，从 /dev/hda1 开始直到 /dev/hda9。在 PC 的早期，分区软件最多只允许四个分区（称为“主”分区）。由于这限制过多，因此产生了一个名为扩展分区的变通方法。扩展分区和主分区非常相似，并且占据四个主分区限制中的一个。然而，扩展分区可以拥有任意数目的所谓逻辑分区，这有效地解决了四个分区的限制。

使用 fdisk (续)

发布时间 :2007-01-27 11:25:08

Command (m for help): p

Disk /dev/hda: 240 heads, 63 sectors, 2184 cylinders

Units = cylinders of 15120 * 512 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	14	105808+	83	Linux
/dev/hda2		15	49	264600	82	Linux swap
/dev/hda3		50	70	158760	83	Linux
/dev/hda4		71	2184	15981840	5	Extended
/dev/hda5		71	209	1050808+	83	Linux
/dev/hda6		210	348	1050808+	83	Linux
/dev/hda7		349	626	2101648+	83	Linux
/dev/hda8		627	904	2101648+	83	Linux
/dev/hda9		905	2184	9676768+	83	Linux

Command (m for help):

在我们的示例中，hda1 到 hda3 是主分区。hda4 是包含逻辑分区 hda5 到 hda9 的扩展分区。因此，在本示例中，您实际上永远不可能使用 /dev/hda4 直接存储任何文件系统 — 它只能作为分区 hda5 到 hda9 的容器。另外，请注意每个分区都有一个“标识 (Id)”，这也称做“分区类型”。无论您何时创建新的分区，您都必须确保正确设置了分区类型。‘83’是包含 Linux 文件系统的分区的正确类型，‘82’是用于 Linux 交换分区的正确类型。您可以使用 fdisk 中的 t 选项设置分区类型。Linux 内核在引导期间用分区类型设置来自动检测磁盘上的文件系统和交换设备。

fdisk 以及更多内容

发布时间 :2007-01-27 11:25:41

关于 fdisk 的内容还有很多，由于篇幅所限无法在这里全都讨论，这其中还包括新分区的创建（用 n 命令）以及将更改写入磁盘（用 w 命令）。请记住您可以输入 m 来获得帮助。如果您初次接触 fdisk，我建议您通过在没有数据丢失危险的空闲磁盘上创建一些分区来熟悉该程序的用法。一旦创建分区并将它们写入磁盘，您的新分区块设备就准备好可以使用了。我们马上将用这些新的块设备来存储新的 Linux 文件系统。

创建文件系统

发布时间 :2007-01-27 11:26:22

在新的块设备可用来存储文件以前，我们需要在上面创建新的文件系统。我们通过使用 mkfs 命令做到这一点—— 我们根据要创建的文件系统的类型来使用特殊的 mkfs。在本示例中，我们使用 mke2fs 在 /dev/hda6（一个空的、未使用的分区块设备）上创建 ext2 文件系统：

```
# mke2fs /dev/hdc6
mke2fs 1.25 (20-Sep-2001)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
1537088 inodes, 3072423 blocks
153621 blocks (5.00%) reserved for the super user
First data block=0
94 block groups
32768 blocks per group, 32768 fragments per group
16352 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208
```

```
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

This filesystem will be automatically checked every 22 mounts or 180 days, whichever comes first. Use tune2fs -c or -i to override.

通过上述命令，mke2fs 在 /dev/hda6 上创建了空白的 ext2 文件系统。

挂装文件系统

发布时间 :2007-01-27 11:26:54

创建文件系统之后，我们就可以使用 mount 命令挂装它：

```
# mount /dev/hdc6 /mnt
```

要挂装一个文件系统，需要将分区块设备指定为第一个参数，将“挂装点”指定为第二个参数。新的文件系统将在挂装点处“嫁接”。这样做的另一个效果是“隐藏”位于父文件系统上 /mnt 目录中的任何文件。以后卸载该文件系统时，这些文件将重新出现。执行挂装命令后，所有添加到 /mnt 的文件将存储在新的 ext2 文件系统中。

创建和使用交换分区

发布时间 :2007-01-27 11:27:29

如果我们刚刚创建了一个分区作为交换设备使用，我们要使用 `mkswap` 命令初始化该分区，并将这个分区块设备作为参数指定：

```
# mkswap /dev/hdc6
```

不同于常规文件系统，我们不挂装交换分区。而是用 `swapon` 命令启用交换分区：

```
# swapon /dev/hdc6
```

通常，Linux 系统的启动脚本将负责自动启用交换分区。因此，`swapon` 命令通常只在您要立即添加刚创建的交换分区时才需要。要查看当前启用的交换设备，请输入 `cat /proc/swaps`。

查看挂装的文件系统

发布时间 :2007-01-27 11:28:03

要查看挂装了什么文件系统，只需输入 mount 本身即可：

```
# mount
/dev/ide/host0/bus1/target0/lun0/part7 on / type xfs (rw,noatime,nodiratime)
proc on /proc type proc (rw)
none on /dev type devfs (rw)
tmpfs on /dev/shm type tmpfs (rw)
/dev/hdc6 on /mnt type ext2 (rw)
```

您也可以通过输入 `cat /proc/mounts` 查看类似信息。因为我的 Linux 系统使用 devfs，所以 mount 输出的第一行列出了较长的“根（root）”分区块设备路径。“根”文件系统将在引导期间由内核自动挂装。对 /dev 使用新的 devfs 设备管理文件系统的系统所用的正式的分区和磁盘块设备名称比 Linux 过去一直使用的更长。例如，/dev/ide/host0/bus1/target0/lun0/part7 是 /dev/hdc7 的正式名称，而 /dev/hdc7 本身只是指向正式块设备的符号链接。通过查看 /dev/.devfsd 文件是否存在，您可以确定文件系统是否在使用 devfs；如果该文件存在，则 devfs 是活动的。

挂装选项

发布时间 :2007-01-27 11:28:45

可以通过指定挂装选项来定制待挂装文件系统的各种属性。例如，您可以使用“ro”选项将文件系统挂装为“只读”：

```
# mount /dev/hdc6 /mnt -o ro
```

/dev/hdc6 挂装为只读后，就不能修改 /mnt 中的任何文件 — 只能读取。如果您的文件系统已经挂装为“读 / 写”方式，而您想把它切换为只读方式，您可以使用“remount”选项以避免再次卸装和重新挂装该文件系统：

```
# mount /mnt -o remount,ro
```

请注意：不需要指定分区块设备，因为已经挂装了文件系统而且 mount 知道 /mnt 与 /dev/hdc6 相关联。要使文件系统再次为可写，我们可以将它重新挂装为读 / 写方式：

```
# mount /mnt -o remount,rw
```

请注意：如果有任何进程打开了 /mnt 中的任何文件或目录，那么这些重新挂装命令将不会成功地完成。要熟悉 Linux 下所有可用的挂装选项，请输入 man mount。

介绍 fstab

发布时间 :2007-01-27 11:29:17

到目前为止，我们已经了解了如何手工挂装文件系统。一般而言，如果您要经常挂装一个文件系统，那么手工挂装往往有些麻烦。而且对于基本的文件系统（如一个单独的 /var 文件系统），手工挂装是不可能的。这些文件系统需要在引导期间自动挂装，只要向 /etc/fstab 文件添加合适的项，我们就可以告诉系统做这些事情。即使您不希望在引导期间自动挂装该文件系统，一个 /etc/fstab 项也可以使挂装变得比较容易，正如我们马上会看到的那样。

样本 fstab

发布时间 :2007-01-27 11:29:56

让我们考察一个样本 /etc/fstab 文件：

```
# <fs>      <mountpoint>  <type>      <opts>      <dump/pass>

/dev/hda1    /boot ext2 noauto,noatime 1 2
/dev/hdc7    /  xfs noatime,osyncisdsync,nodiratime 0 1
/dev/hdc5     none      swap sw 0 0
/dev/cdrom /mnt/cdrom iso9660 noauto,ro,user 0 0
# /proc should always be enabled
proc /proc      proc defaults 0 0
```

如上所示，/etc/fstab 中的每个未注释行指定一个分区块设备、一个挂装点、一个文件系统类型、挂装该文件系统时要用到的文件系统选项以及两个数字字段。第一个数字字段用来向 dump 备份命令指明应该备份的文件系统。当然，如果您不打算在系统上使用 dump 命令，那么忽略这个字段不会有问题。后一个字段由 fsck 文件系统完整性检查程序使用，并告诉该程序在引导时检查文件系统所应采用的顺序。我们将在下几屏再次接触 fsck。

观察 /dev/hda1 行；您会发现 /dev/hda1 是一个应该挂装在 /boot 挂装点的 ext2 文件系统。现在，观察 <opts> 列中 /dev/hda1 的挂装选项。noauto 选项告诉系统在引导期间不要自动挂装 /dev/hda1；如果没有这个选项，系统引导时会自动将 /dev/hda1 挂装到 /boot。

还要注意 noatime 选项，它关闭了磁盘上 atime（最近访问时间）信息的记录。该信息通常是不需要的，关闭 atime 更新对文件系统性能有积极作用。您还可以使用 nodiratime 挂装选项来关闭目录 atime 更新。

样本 fstab（续）

发布时间 :2007-01-27 11:30:41

```
# <fs>      <mountpoint>  <type>  <opts>      <dump/pass>
```

```
/dev/hda1    /boot          ext2 noauto,noatime 1 1
/dev/hdc7    /              xfs  noatime,osyncisdsync,nodiratime 0 0
/dev/hdc5    none           swap sw  0 0
/dev/cdrom   /mnt/cdrom     iso9660 noauto,ro,user  0 0
# /proc should always be enabled
proc         /proc          proc defaults  0 0
```

现在，观察 /proc 行并注意 defaults 选项。当希望只用标准挂装选项挂装文件系统时，请使用 defaults。因为 /etc/fstab 有多个字段，所以我们不能简单地让这个选项字段留作空白。

还要注意 /dev/hdc5 的 /etc/fstab 行。该行将 /dev/hdc5 定义为交换设备。因为交换设备不象文件系统那样被挂装，所以在挂装点字段指定 none。由于这个 /etc/fstab 项，系统启动时将自动启用 /dev/hdc5 交换设备。

如果有如上所示用于 /dev/cdrom 的 /etc/fstab 项，挂装 CD-ROM 驱动器会变得更加简单。我们不再输入：

```
# mount -t iso9660 /dev/cdrom /mnt/cdrom -o ro
```

我们现在可以输入：

```
# mount /dev/cdrom
```

事实上，使用 /etc/fstab 允许我们利用 user 选项。user 挂装选项告诉系统允许任何用户挂装这个特殊的文件系统。这对于可移动媒体设备（象 CD-ROM 驱动器）很方便。如果没有这个 fstab 挂装选项，则只有 root 用户能够使用 CD-ROM 驱动器。

卸装文件系统

发布时间 :2007-01-27 11:31:17

通常情况下，所有挂装的文件系统在系统重新引导或关机时都会自动卸装。当卸装文件系统时，内存中所有位于高速缓存中的文件系统数据都被刷新到磁盘中。

然而，也可以手工卸装文件系统。在可以卸装文件系统以前，您首先必须确保正在运行的进程在当前讨论的文件系统上没有打开的文件。然后，使用 `umount` 命令，将设备名称或者挂装点指定为参数：

```
# umount /mnt
```

或

```
# umount /dev/hdc6
```

卸装之后，`/mnt` 中被以前挂装的文件系统所“覆盖”的所有文件将重新出现。

介绍 fsck

发布时间 :2007-01-27 11:31:49

如果您的系统因某种原因崩溃或锁定，那么系统将无法彻底卸装您的文件系统。如果发生了这种情况，您的文件系统就处于不一致（不可预测）状态。当系统重新引导时，fsck 程序将检测出文件系统没有完全卸装，并希望对 /etc/fstab 中列出的文件系统的一致性检查。

有一点很重要 — 对于由 fsck 检查的文件系统，必须在 /etc/fstab 的 “ pass ” 字段（最后一个字段）中有非零数字。通常情况下，应将根文件系统的 passno 设置为 1，这指定应首先检查它。所有其它应在启动期间检查的文件系统的 passno 应为 2 或更高。

有时候，您会发现 fsck 在重新引导后不能完全修复一个部分损坏的文件系统。在这些情况下，您所能做的就是将系统降为单用户模式，然后手工运行 fsck，并将这个分区块设备作为参数提供。在 fsck 进行文件系统修复时，它可能会问您是否修复特殊的文件系统缺陷。通常情况下，您应该对所有这些问题回答 y (yes) 并允许 fsck 执行它的工作。

fsck 的问题

发布时间 :2007-01-27 11:32:27

fsck 扫描的问题之一是需要花较长的时间才能完成，因为它要扫描整个文件系统的元数据（内部数据结构）以确保其一致性。对于一些非常大的文件系统，用超过一小时的时间进行一次彻底的 fsck 并不罕见。

为了解决这个问题，人们设计了一种新的文件系统，名为日志记录文件系统（journaling filesystem）。日志记录文件系统记录一个近期对文件系统元数据所做更改的磁盘日志。如果发生崩溃，该文件系统驱动程序检查日志。因为日志含有磁盘上近期更改的精确记录，所以只需检查这部分文件系统元数据来找出错误。多亏这一重要的设计差异，对一个日志记录系统进行一致性检查通常只需大约几秒钟（不管文件有多大）。出于这个原因，日志记录文件系统正在 Linux 社区流行起来。

现在，让我们研究可用于 Linux 的各种文件系统。

ext2 文件系统

发布时间 :2007-01-27 11:33:38

做为参考，目前已到了ext4，现多用ext3。

ext2 文件系统多年来已经成为标准的 Linux 文件系统。对于大多数应用程序，它通常都有很好的性能，但它没有任何日志记录能力。这使得它不适合非常大的文件系统，因为执行 fscks 要花太多的时间。另外，由于每个 ext2 文件系统只能拥有固定数目的索引节点（inode），所以 ext2 有些内置的限制。可以这么说，通常认为 ext2 是一个非常健壮和有效的非日志记录文件系统。

内核：2.0+

日志记录：无

mkfs 命令：mke2fs

mkfs 示例：mke2fs /dev/hdc7

相关命令：debugfs、tune2fs 和 chattr

与性能相关的挂装选项：noatime 和 nodiratime

ext3 文件系统

发布时间 :2007-01-27 11:34:15

ext3 文件系统使用与 ext2 相同的磁盘格式，但增加了日志记录能力。事实上，在所有的 Linux 文件系统中，ext3 具有最广泛的日志记录支持，它不仅支持元数据日志记录，还支持有序日志记录（缺省）和完全的“元数据+数据”日志记录。这些“特殊”日志记录方式有助于确保数据完整性，而不象其它日志记录实现仅仅缩短 fsck 的运行时间。出于这个原因，如果数据完整性是绝对最重要的，那么 ext3 是可用的最佳文件系统。然而，这些数据完整性功能确实会在某种程度上影响性能。另外，因为 ext3 使用与 ext2 相同的磁盘格式，所以和它的非日志记录表亲（ext2）一样，它也受到同样的可伸缩性限制。如果您在寻找一个出色的、通用的同时又非常健壮的日志记录文件系统，那么 ext3 是很好的选择。

内核：2.4.16+

日志记录：元数据、有序数据写入和完全的“元数据+数据”

mkfs 命令：mke2fs -j

mkfs 示例：mke2fs -j /dev/hdc7

相关命令：debugfs、tune2fs 和 chattr

与性能相关的挂装选项：noatime 和 nodiratime

其它挂装选项：

data=writeback（禁用日志记录）

data=ordered（缺省值，将元数据日志记录和数据与元数据一起写到磁盘）

data=journal（用于数据和元数据完整性的完全数据日志记录。写操作性能降低一半。）

ReiserFS 文件系统

发布时间 :2007-01-27 11:34:47

ReiserFS 是一个相对较新的文件系统，它的设计目标是提供非常好的小文件性能、非常好的通用性能以及是非常可伸缩的。ReiserFS 使用元数据日志以避免长时间的 fsck，但日志实现可能使最近修改的数据在系统锁定时被毁坏。通常情况下，ReiserFS 有非常好的性能，但在装入特定种类的文件系统时可能出现某种性能反常的情况。另外，ReiserFS 的 fsck 工具还不成熟，因此从毁坏的文件系统恢复数据可能会有困难。这些问题中的很多是由于 ReiserFS 相对较新、仍在发展过程中。许多人因 ReiserFS 的速度和可伸缩性而喜爱它。

内核：2.4.0+（推荐 2.4.16+）

日志记录：元数据

mkfs 命令：mkreiserfs

mkfs 示例：mkreiserfs /dev/hdc7

与性能相关的挂装选项：noatime、nodiratime 和 notail

XFS 文件系统

发布时间 :2007-01-27 11:35:26

XFS 文件系统是正在由 SGI 移植到 Linux 的企业级日志记录文件系统。现成的内核中还没有 XFS，不过在 <http://oss.sgi.com/projects/xfs> 找到有关 XFS 的更多信息。

JFS 文件系统

发布时间 :2007-01-27 11:35:59

JFS 是一个由 IBM 移植到 Linux 的高性能日志记录文件系统。JFS 由 IBM 企业服务器使用，并且是为高性能应用程序而设计的。现成的内核中还没有 JFS。您可以在 JFS 项目网站学习更多有关 JFS 的知识。

VFAT

发布时间 :2007-01-27 11:36:27

VFAT 文件系统实际不是那种您可以选择用于存储 Linux 文件的文件系统。相反，它是一种与 DOS 兼容的文件系统驱动程序，允许您挂装基于 DOS 和 Windows FAT 的文件系统并与之交换数据。VFAT 文件系统驱动程序在标准 Linux 内核中存在。

引导系统

引导系统

发布时间 :2007-01-27 11:37:58

本节介绍 Linux 引导过程。我们将介绍引导装入程序的概念、如何在引导时设置内核选项以及如何检查引导日志以找出错误。

MBR

发布时间 :2007-01-27 11:38:41

不管安装的是哪个分发版（distribution），所有机器的引导过程都很相似。考虑下列硬盘示例：

```
+-----+
|  MBR  |
+-----+
| Partition 1: |
| Linux root (/) |
| containing |
| kernel and |
| system.  |
+-----+
| Partition 2: |
| Linux swap  |
+-----+
| Partition 3: |
| Windows 3.0 |
| (last booted |
| in 1992)  |
+-----+
```

首先，计算机的 BIOS 读取硬盘上的头几个扇区。这些扇区包含一个非常小的程序，名为“主引导记录（Master Boot Record）”，或简称为“MBR”。MBR 已经存储了 Linux 内核在硬盘上的位置（在上面的示例中是分区 1），因此它将内核装入内存，然后启动它。

内核引导过程

发布时间 :2007-01-27 11:39:23

接下来您看到的東西（尽管它可能一闪而过）是与下面类似的一行：

```
Linux version 2.4.16 ( root@time.flatmonk.org) (gcc version 2.95.3 20010315 (release)) #1 Sat Jan 12  
19:23:04 EST 2002
```

这就是内核开始运行时所打印的第一行信息。首先是内核版本，接着是构建该内核的用户标识（通常是 root 用户），然后是构建它的编译器，最后是构建时的时间戳记。

那一行之后是来自内核的关于系统硬件的大量输出：处理器、PCI 总线、磁盘控制器、磁盘、串口、软驱、USB 设备、网络适配器、声卡和其它可能的设备将依次报告它们的状态。

/sbin/init

发布时间 :2007-01-27 11:40:01

当内核装入结束时，它启动一个名为 init 的程序。该程序直到系统关机才停止运行。如您所见，始终给它分配进程标识 1：

```
$ ps --pid 1
PID TTY          TIME CMD
  1 ?        00:00:04 init.system
```

init 程序通过运行一系列脚本来引导分发版的余下部分。这些脚本通常位于 /etc/rc.d/init.d 或 /etc/init.d 中，它们执行的服务有设置系统主机名、检查文件系统是否有错误、挂装附加的文件系统、启用联网以及启动打印服务等。当脚本运行结束时，init 启动名为 getty 的程序，该程序会显示登录提示符，然后您就可以开始使用了！

深入研究： LILO

发布时间 :2007-01-27 11:40:31

既然我们已经快速了解了引导过程，现在让我们更仔细地研究第一个部分：MBR 和装入内核。维护 MBR 是“引导装入程序”的职责。基于 x86 的 Linux 的两个最流行的引导装入程序是“LILO”（Linux LOader 的缩写）和“GRUB”（GRand Unified Bootloader 的缩写）。

LILO 是二者中出现得较早、较常用的引导装入程序。引导时会出现简短的“LILO boot:”提示符，说明 LILO 存在于您的系统上。请注意，您可能要在引导时按住 shift 键才能看到提示符，因为系统常常被配置为不间断地快速运行，从而会很快闪过这个画面。

LILO 提示符没有太多华而不实的東西，不过如果您按一下 <tab> 键，您会看到一个可引导的可选内核（或其它操作系统）列表。通常列表中只有一项。您可以输入其中一项的名称然后按 <enter> 来引导它。或者您可以简单地按一下 <enter>，在缺省情况下将引导列表中的第一项。

使用 LILO

发布时间 :2007-01-27 11:41:09

有时候您希望在引导期间将一个选项传递到内核。一些较常用的选项是：root= 用来指定替代的引导文件系统，init= 用来指定替代的 init 程序（如指定 init=/bin/sh 来挽救被错误配置的系统），mem= 指定系统中的内存数量（例如，在 Linux 只自动识别了 128 M 的情况下使用 mem=512M）。您可以在 LILO 引导提示符下向内核传递下列选项：

LILO boot: linux root=/dev/hdb2 init=/bin/sh mem=512M

如果您需要经常指定命令行选项，您应该考虑把它们添加到 /etc/lilo.conf 文件。

一个重要的 LILO 问题

发布时间 :2007-01-27 11:41:48

在转向讨论 GRUB 以前，还要讨论一个重要的 LILO 问题。只要您对 /etc/lilo.conf 做了更改，或安装了新的内核，您都必须运行 lilo。lilo 程序会重写 MBR 以反映您所做的更改，包括记录内核的绝对磁盘位置。这里的示例利用 -v 标记进行详细信息描述。

```
# lilo -v
LILO version 21.4-4, Copyright (C) 1992-1998 Werner Almesberger
'lba32' extensions Copyright (C) 1999,2000 John Coffman
```

```
Reading boot sector from /dev/hda
Merging with /boot/boot.b
Mapping message file /boot/message
Boot image: /boot/vmlinuz-2.2.16-22
Added linux *
/boot/boot.0300 exists - no backup copy made.
Writing boot sector.
```

深入研究： GRUB

发布时间 :2007-01-27 11:42:38

GRUB 引导装入程序被认为是继 LILO 之后的下一代引导装入程序。对用户而言，最显而易见的是它用一个菜单界面取代了 LILO 的原始提示符。对于系统管理员而言，改变更为显著。GRUB 比 LILO 支持更多的操作系统，它在引导菜单中提供了基于密码的安全性，并且更易于管理。

通常用 grub-install 命令安装 GRUB。安装完毕后，可以编辑文件 /boot/grub/menu.lst 来管理 GRUB 菜单。这两个任务都超出了本文的讨论范围；在试图安装或管理 GRUB 之前，您应该阅读 GRUB 信息页。

使用 GRUB

发布时间 :2007-01-27 11:43:21

要向内核传递参数，您可以在引导菜单上按 e。这使您有机会编辑（再次按 e）要装入的内核名称或传递给它的参数。当您结束编辑后，按 <enter> 键，然后按 b 键以用您所做的更改来引导。

LILO 与 GRUB 之间值得一提的显著区别是：GRUB 不需要在每次更改配置或安装新内核后重新安装其引导装入程序。这是因为 GRUB 理解 Linux 文件系统，而 LILO 只是存储要装入内核的绝对磁盘位置。当系统管理员安装新内核后忘了输入 lilo 时，GRUB 的这一简单事实可以缓解他们的挫折感。

dmesg

发布时间 :2007-01-27 11:44:07

来自内核和 init 脚本的引导消息通常在屏幕上停留的时间很短，您可能会注意到一个错误消息，但在您能看清楚它以前，它已经消失了。在这种情况下，您可以在系统引导之后到两个地方去查看以找出错误（并且有希望找到修复它的方法）。

如果错误出现在内核正在装入或检测硬件设备的时候，您可以用 dmesg 命令获得一个内核日志的副本。

```
# dmesg | head -1
```

```
Linux version 2.4.16 ( root@time.flatmonk.org) (gcc version 2.95.3 20010315 (release)) #1 Sat Jan 12  
19:23:04 EST 2002
```

嘿，我们认得那一行！这就是内核在装入时打印的第一行信息。的确，如果您将 dmesg 的输出传送到分页程序上，您会看到内核在引导时打印的所有消息以及内核在那段时间已打印到控制台的所有消息。

/var/log/messages

发布时间 :2007-01-27 11:44:39

第二个查看信息的地方在 /var/log/messages 文件中。该文件由 syslog 守护程序记录，syslog 守护程序接受来自库、守护程序和内核的输入。消息文件中的每一行都被打上时间戳记。这个文件是查找引导期间的 init 脚本阶段出现的错误的好位置。例如，要查看来自名称服务器的最后几条消息：

```
# grep named /var/log/messages | tail -3
Jan 12 20:17:41 time /usr/sbin/named[350]: listening on IPv4 interface lo, 127.0.0.1#53
Jan 12 20:17:41 time /usr/sbin/named[350]: listening on IPv4 interface eth0, 10.0.0.1#53
Jan 12 20:17:41 time /usr/sbin/named[350]: running
```

单用户方式

发布时间 :2007-01-27 11:45:10

我们知道，可以在内核引导时向它传递参数。最常使用的参数之一是 s，它使系统以“单用户”方式启动。这一方式通常只挂装根文件系统，启动 init 脚本的最小子集，然后启动 shell 而不是提供登录提示符。另外，没有配置联网，因此外部因素没有机会影响您的工作。

使用单用户方式

发布时间 :2007-01-27 11:45:38

那么在这样的状态下我们能够“做什么”呢？要回答这个问题，我们必须认识到 Linux 和 Windows 之间的巨大差异。Windows 被设计成通常在同一时间内只能由坐在控制台前的一个人使用。它实际上一直处于“单用户”方式。相反，Linux 更多地用于为网络应用程序服务，或为网络上的远程用户提供 shell 或 X 会话。当您希望执行维护操作（如：从备份中恢复、创建或修改文件系统、从 CD 升级系统等等）时，您不希望有这些额外的变数。在这些情况下，您应该使用单用户方式。

更改运行级别

发布时间 :2007-01-27 11:46:17

事实上，要进入单用户方式不必重新引导。init 程序管理系统的当前方式（或称为“运行级别（runlevel）”）。Linux 系统的标准运行级别按如下分类并定义：

- 0：停止计算机
- 1 或 s：单用户方式
- 2：多用户，无网络
- 3：多用户，文本控制台
- 4：多用户，图形控制台
- 5：同 4
- 6：重新引导计算机

这些运行级别因分发版而异，所以请确保参考您的分发版的文档。要转至单用户方式，可以使用 telinit 命令，它指示 init 更改运行级别：

```
# telinit 1
```

从上表中您可以看到您还可以用这种方式关闭或重新引导系统。telinit 0 将停机；telinit 6 将重新引导计算机。当您发出 telinit 命令更改运行级别时，init 脚本的一个子集将运行以关闭或启动系统服务。

恰当地关机

发布时间 :2007-01-27 11:46:49

然而，如果还有用户在使用系统，那么此时关机是相当粗鲁的（用户可能会非常生气）。shutdown 命令提供了一种方法，以一种合理对待用户的方式来更改运行级别。类似于 kill 命令那种可以向一个进程发送多种信号的能力，shutdown 可以用来停机、重新引导或转至单用户方式。例如，要在 5 分钟内转至单用户方式：

```
# shutdown 5
Broadcast message from root (pts/2) (Tue Jan 15 19:40:02 2002):
The system is going DOWN to maintenance mode in 5 minutes!
```

如果此时您按 control-c 组合键，您可以取消切换至单用户方式前的延时等待。上面的消息将在系统的所有终端上出现，因此用户有合理数量的时间保存他们的工作并注销。（有些人可能会争论 5 分钟是不是“合理”的）。

立即关机

发布时间 :2007-01-27 11:47:21

如果您是唯一使用系统的人，您可以用“now”代替以分钟为单位的参数。例如，要立即重新引导系统：

```
# shutdown -r now
```

在此情况下，您没有机会按 control-c 组合键；因为系统已经在进行关机了。最后，-h 选项使系统停机：

```
# shutdown -h 1
Broadcast message from root (pts/2) (Tue Jan 15 19:50:58 2002):
The system is going DOWN for system halt in 1 minute!
```

缺省运行级别

发布时间 :2007-01-27 11:47:51

此时您可能已得出结论：init 程序在 Linux 系统上是非常重要的。您可以编辑文件 /etc/inittab 来配置 init，这在 inittab(5) 手册页中有描述。我们只讨论这个文件中的关键一行。

```
# grep ^id: /etc/inittab  
id:3:initdefault:
```

在我的系统上，运行级别 3 是缺省运行级别。如果您更喜欢您的系统立即引导至图形登录界面（通常为运行级别 4 或 5），那么更改这个值会有用。要那样做的话，只需简单地编辑文件并更改那一行的值。但要小心！如果您将它改为某个无效值，您可能不得不使用我们在前面提到的 init=/bin/sh 技巧。

运行级别

运行级别 - 单用户方式

发布时间 :2007-01-27 11:48:41

还记得关于引导装入程序一节所讲的吗？在引导内核时向它传递参数是可能的。最常用的参数之一是 s，它使系统以“单用户”方式启动。这一方式通常只挂装根文件系统，启动 init 脚本的最小子集，然后启动 shell 而不是提供登录提示符。另外，它不配置联网，因此外部因素没有机会影响您的工作。

单用户方式（续）

发布时间 :2007-01-27 11:49:13

那么在这样的状态下我们能够“做什么”呢？要回答这个问题，我们必须认识到 Linux 和 Windows 之间的巨大差异。Windows 被设计成通常在同一时间内只能由坐在控制台前的一个人使用。它实际上一直处于“单用户”方式。相反，Linux 更多地用于为网络应用程序服务，或为网络上的远程用户提供 shell 或 X 会话。当您希望执行维护操作（如：从备份中恢复、创建或修改文件系统、从 CD 升级系统等等）时，您不希望有这些额外的变数。在这些情况下，您应该使用单用户方式。

运行级别

发布时间 :2007-01-27 11:49:46

事实上，要进入单用户方式不必重新引导。init 程序管理系统的当前方式（或称为“运行级别”）。Linux 系统的标准运行级别按如下分类并定义：

- 0：停止计算机
- 1 或 s：单用户方式
- 2：多用户，无网络
- 3：多用户，文本控制台
- 4：多用户，图形控制台
- 5：同 4
- 6：重新引导计算机

这些运行级别因分发版而异，所以请确保参考您的分发版的文档。

telinit

发布时间 :2007-01-27 11:50:19

要转至单用户方式，可以使用 telinit 命令，它指示 init 更改运行级别：

```
# telinit 1
```

从上表中您可以看到您还可以用这种方式关机或重新引导系统。telinit 0 将停机；telinit 6 将重新引导计算机。当您使用 telinit 命令更改运行级别时，init 脚本的一个子集将运行以关闭或启动系统服务。

运行级别规则

发布时间 :2007-01-27 11:50:55

然而，如果还有用户在使用系统，那么此时关机是相当粗鲁的（用户可能会非常生气）。shutdown 命令提供了一种方法，以一种合理对待用户的方式来更改运行级别。类似于 kill 命令那种可以向一个进程发送多种信号的能力，shutdown 以用来停机、重新引导或转至单用户方式。例如，要在 5 分钟内转至单用户方式：

```
# shutdown 5
Broadcast message from root (pts/2) (Tue Jan 15 19:40:02 2002):
The system is going DOWN to maintenance mode in 5 minutes!
```

如果此时您按 control-c 组合键，您可以取消切换至单用户方式前的延时等待。上面的消息将在系统的所有终端上出现，因此用户有合理数量的时间保存他们的工作并注销。（有些人可能会争论 5 分钟是不是“合理”的）。

“ now ” 和停机

发布时间 :2007-01-27 11:51:34

如果您是唯一使用系统的人，您可以用 now 代替以分钟为单位的参数。例如，要立即重新引导系统：

```
# shutdown -r now
```

在此情况下，您没有机会按 control-c 组合键；因为系统已经在进行关机了。最后，-h 选项使系统停机：

```
# shutdown -h 1
Broadcast message from root (pts/2) (Tue Jan 15 19:50:58 2002):
The system is going DOWN for system halt in 1 minute!
```

配置 init

发布时间 :2007-01-27 11:52:10

此时您可能已得出结论：init 程序在 Linux 系统上是非常重要的。您可以编辑文件 /etc/inittab 来配置 init，这在 inittab(5) 手册页中有描述。我们只讨论这个文件中的关键一行。

```
# grep ^id: /etc/inittab  
id:3:initdefault:
```

在我的系统上，运行级别 3 是缺省运行级别。如果您更喜欢您的系统立即引导至图形登录（通常为运行级别 4 或 5），那么更改这个值会有用。要那样做的话，只需简单地编辑文件并在那一行更改那个值。但要小心！如果您将它改为某个无效值，您可能不得不使用我们在前面提到的 init=/bin/sh 技巧。

文件系统限额

文件系统限额

发布时间 :2007-01-27 11:52:48

限额（quota）是 Linux 的一个特性，它让您跟踪用户或组的磁盘使用情况。它们可用于防止任何单个用户或组不公平地使用文件系统的一部分或将它填满。限额只能由 root 用户启用和管理。在本节中，我将介绍如何在 Linux 系统上设置限额并有效地管理它们。

内核支持

发布时间 :2007-01-27 11:53:28

限额是文件系统的一个特性；因此，它们需要内核支持。首先需要做的是验证您的内核支持限额。您可以用 `grep` 做到这一点：

```
# cd /usr/src/linux
# grep -i quota .config
CONFIG_QUOTA=y
CONFIG_XFS_QUOTA=y
```

如果难以用该命令返回的信息作出结论（如未设置 `CONFIG_QUOTA`），那么您应该重新构建内核以包含限额支持。这一过程并不困难，但超出本教程这一节的讨论范围之外。如果您不熟悉构建和安装新内核的步骤，您应该考虑参考这个教程。

文件系统支持

发布时间 :2007-01-27 17:27:53

在深入研究限额管理以前，请注意 2.4.x 内核为止的 Linux 系列上的限额支持尚不完整。限额目前在 ext2 和 ext3 文件系统中还存在问题，而 ReiserFs 似乎根本不支持限额。本教程的示例使用的是 XFS，它似乎可以较好地支持 quota。

配置限额

发布时间 :2007-01-27 17:28:29

要开始在系统配置限额，您应该编辑 /etc/fstab 文件以挂装受启用限额影响的文件系统。在我们的示例中，我们使用挂装时启用用户和组限额的 XFS 文件系统。

```
# grep quota /etc/fstab
/usr/users /mnt/hdc1 xfs  usrquota,grpquota,noauto 0 0
# mount /usr/users
```

配置限额（续）

发布时间 :2007-01-27 17:29:11

请注意：usrquota 和 grpquota 选项不一定在文件系统上启用限额。要确保启用了限额，您可以使用 quotaon 命令：

```
# quotaon /usr/users
```

如果您想以后禁用限额，则有一个相应的 quotaoff 命令：

```
# quotaoff /usr/users
```

但在目前，如果您正在尝试本教程中的几个示例，请确保启用了限额。

quota 命令

发布时间 :2007-01-27 17:29:45

quota 命令显示了当前挂装的所有文件系统的用户磁盘使用情况和限制。-v 选项包括启用了限额的文件系统列表，但当前没有给用户分配存储空间。

```
# quota -v
```

Disk quotas for user root (uid 0):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hdc1	0	0	0	3	0	0		

第一列 blocks 显示 root 用户当前在每个列出的文件系统上正在使用多少磁盘空间。接下来的 quota 和 limit 列显示当前的磁盘空间限制。我们稍后将解释 quota 和 limit 之间的区别以及 grace 列的含义。files 列显示 root 用户在特定文件系统上拥有多少文件。其后的 quota 和 limit 列则显示对这些文件的限制。

查看限额

发布时间 :2007-01-27 17:30:18

任何用户都可以使用 quota 命令查看自己的限额报告，如前一个示例所示。但是只有 root 用户可以查看其他用户和组的限额。例如，假设我们在 /usr/users 上挂装了一个文件系统 /dev/hdc1，并且有两个用户：jane 和 john。首先，让我们看看 jane 的磁盘使用情况和限制。

```
# quota -v jane
```

Disk quotas for user jane (uid 1003):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hdc1	4100	0	0	6	0	0		

在这一示例中，我们看到 jane 的 quota 被设置为零，这表示没有限制。

edquota

发布时间 :2007-01-27 17:30:52

现在假设我们希望给用户 jane 一个限额。我们用 edquota 命令实现。在我们开始编辑限额之前，让我们看看我们在 /usr/users 上还有多少可用空间：

```
# df /usr/users
```

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hdc1	610048	4276	605772	1%	/usr/users

这不是一个特别大的文件系统，仅仅 600M 左右。给 jane 一个限额似乎是明智的，这样她所用的空间就不会超过她应得的空间。当您运行 edquota 时，会为您在命令行指定的每个用户或组创建一个临时文件。

edquota（续）

发布时间 :2007-01-27 17:31:25

edquota 命令为您提供一个编辑器，它使您能通过这个临时文件添加和 / 或修改 quota。

```
# edquota jane
```

Disk quotas for user jane (uid 1003):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hdc1	4100	0	0	6	0	0

与上面 quota 命令的输出相似，这个临时文件中的 blocks 和 inodes 列显示 jane 当前正在使用的磁盘空间和文件数目。您不能修改 blocks 或 inodes 的数量；任何这样的尝试都将立即被系统废弃。soft 和 hard 列显示 jane 的限额，我们可以看到当前对它没有限制（零表明没有限额）。

理解 edquota

发布时间 :2007-01-27 17:31:56

soft 限制是在文件系统上分配给 jane 的磁盘空间最大使用量（也就是她的限额）。如果 jane 使用的磁盘空间数量超过在 soft 限制中分配给她的空间数量，将通过电子邮件就她的违规行为提出警告。hard 限制表明对磁盘使用的绝对限制，用户不能超过该限制。如果 jane 试图使用的磁盘空间多于 hard 限制中指定的磁盘空间，她将得到 “ Disk quota exceeded ” 错误信息并且不能完成该操作。

进行更改

发布时间 :2007-01-27 17:32:33

那么我们在这里更改 jane 的 soft 和 hard 限制，然后保存该文件：

Disk quotas for user jane (uid 1003):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/hdc1	4100	10000	11500	6	2000	2500

运行 quota 命令，我们可以检查我们所做的修改：

```
# quota jane
```

Disk quotas for user jane (uid 1003):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hdc1	4100	10000	11500		6	2000	2500	

复制限额

发布时间 :2007-01-27 17:33:09

您会记得在这个文件系统上我们还有另一个用户 john。如果我们希望给 john 的限额与 jane 的限额相同，我们可以在 edquota 命令中使用 -p 选项，它会将 jane 的限额作为原型用于命令行上所有随后的用户。这是一个为用户组设置限额的简便方法。

```
# edquota -p jane john
# quota john
```

Disk quotas for user john (uid 1003):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hdc1	0	10000	11500		1	2000	2500	

组限制

发布时间 :2007-01-27 17:33:49

使用 edquota , 我们还可以根据文件的组所有权来限制磁盘空间分配。例如 , 要编辑 users 组的限额 :

```
# edquota -g users Disk quotas for group users (gid 100): Filesystem blocks soft hard inodes soft hard
/dev/hdc1 4100 500000 510000 7 100000 125000
```

然后要查看修改的 users 组的限额 :

```
# quota -g users Disk quotas for group users (gid 100): Filesystem blocks quota limit grace files quota limit
grace /dev/hdc1 4100 500000 510000 7 100000 125000
```

repquota 命令

发布时间 :2007-01-27 17:34:26

如果您的文件系统有很多用户，那么使用限额命令查看每个用户的限额会比较麻烦。repquota 命令将文件系统的限额以一个清楚易懂的报告进行总结。例如，要查看 /usr/users 上所有用户和组的限额：

```
# repquota -ug /usr/users
```

```
*** Report for user quotas on device /dev/hdc1
```

```
Block grace time: 7days; Inode grace time: 7days
```

User	Block limits				File limits			
	used	soft	hard	grace	used	soft	hard	grace
root	--	0	0	0	3	0	0	
john	--	0	10000	11500	1	2000	2500	
jane	--	4100	10000	11500	6	2000	2500	

```
*** Report for group quotas on device /dev/hdc1
```

```
Block grace time: 7days; Inode grace time: 7days
```

Group	Block limits				File limits			
	used	soft	hard	grace	used	soft	hard	grace
root	--	0	0	0	3	0	0	
users	--	4100	500000	510000	7	100000	125000	

repquota 选项

发布时间 :2007-01-27 17:34:59

repquota 有其它两个选项值得一提。repquota -a 将报告所有启用了限额并且当前已挂装的读-写文件系统。repquota -n 将不把 uid 和 gid 解析为名称。这可以加快大型列表的输出。

监控限额

发布时间 :2007-01-27 17:35:32

如果您是系统管理员，您会希望有一种方法可以监控限额以确保没有用户超过它们。实现它的一个简单方法是使用 `warnquota`。`warnquota` 命令会向超过 `soft` 限制的用户发送电子邮件。通常将 `warnquota` 作为 `cron` 作业运行。

当用户超过 `soft` 限制时，`quota` 命令输出中的 `grace` 列将指出宽限时间 — 在该文件系统强制执行 `soft` 限制之前有多少时间。

Disk quotas for user jane (uid 1003):

Filesystem	blocks	quota	limit	grace	files	quota	limit	grace
/dev/hdc1	10800*	10000	11500	7days	7	2000	2500	

缺省情况下，`blocks` 和 `inodes` 的宽限时间是 7 天。

修改宽限时间

发布时间 :2007-01-27 17:36:06

您可以用 equota 修改文件系统的宽限时间：

```
# edquota -t
```

这为您提供一个临时文件编辑器，它看起来与下面相似：

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem      Block grace period   Inode grace period
/dev/hdc1       7days                7days
```

该文件中的文本简单明了，无需解释。请确保给您的用户足够的时间来接收警告电子邮件并找出一些要删除的文件！

在引导时检查限额

发布时间 :2007-01-27 17:36:40

您可能还希望在引导时检查限额。要实现这一点，您可以使用一个脚本运行 quotacheck 命令；在 Quota Mini HOWTO 有一个脚本示例。quotacheck 命令还具有修复损坏的限额文件的能力；请阅读 quotacheck(8) 手册页来熟悉这一点。

还要记住我前面提到的关于 quotaon 和 quotaoff 的内容。您应该将 quotaon 合并到引导脚本中以启用限额。要在所有支持限额的文件系统上启用 quota，请使用 -a 选项：

```
# quotaon -a
```

系统日志

系统日志 - 介绍 syslogd

发布时间 :2007-01-27 17:37:31

syslog 守护程序为记录来自运行于系统之上的程序的消息提供了一种成熟的客户机-服务器机制。syslog 接收来自守护程序或程序的消息，根据优先级和类型将该消息分类，然后根据由管理员可配置的规则将它写入日志。结果是一个健壮而统一的管理日志的方法。

读取日志

发布时间 :2007-01-27 17:38:06

让我们跳过一些内容来直接研究一个 syslog 记录的日志文件的内容。之后我们再回过头研究 syslog 配置。FHS（请参阅本教程系列的第 2 部分）要求将日志文件放在 /var/log 中。我们在这里使用 tail 命令来显示 “messages” 文件中的最后 10 行：

```
# cd /var/log
# tail messages
Jan 12 20:17:39 bilbo init: Entering runlevel: 3
Jan 12 20:17:40 bilbo /usr/sbin/named[337]: starting BIND 9.1.3
Jan 12 20:17:40 bilbo /usr/sbin/named[337]: using 1 CPU
Jan 12 20:17:41 bilbo /usr/sbin/named[350]: loading configuration from '/etc/bind/named.conf'
Jan 12 20:17:41 bilbo /usr/sbin/named[350]: no IPv6 interfaces found
Jan 12 20:17:41 bilbo /usr/sbin/named[350]: listening on IPv4 interface lo, 127.0.0.1#53
Jan 12 20:17:41 bilbo /usr/sbin/named[350]: listening on IPv4 interface eth0, 10.0.0.1#53
Jan 12 20:17:41 bilbo /usr/sbin/named[350]: running
Jan 12 20:41:58 bilbo gnome-name-server[11288]: starting
Jan 12 20:41:58 bilbo gnome-name-server[11288]: name server starting
```

我们希望您能在文本处理的纷乱事件中记得 tail 命令显示文件的最后几行。在本示例中，我们可以看到最近在这个系统上启动了名为 bilbo 的名称服务器 named。如果我们在部署 IPv6，我们可能会注意到 named 找不到 IPv6 接口，这表明可能出现了问题。另外，我们可以看到最近可能有用户已经启动了 GNOME，这可以由 gnome-name-server 的存在看出。

跟踪日志文件

发布时间 :2007-01-27 17:38:38

有经验的系统管理员可能会使用 `tail -f` 以便当日志文件的输出出现时进行跟踪：

```
# tail -f /var/log/messages
```

例如，以调试理论上的 IPv6 问题为例，在停止和启动 `named` 时在一个终端运行上述命令会立即显示来自该守护程序的消息。这在调试时是一个有用的技术。有些管理员甚至喜欢在终端上一直运行 `tail -f messages`，这样他们可以随时关注系统事件。

查找日志

发布时间 :2007-01-27 17:39:14

另一种有用的技术是使用 grep 实用程序搜索日志文件，这在本教程系列的第 2 部分有描述。在上述示例中，我们可以使用 grep 找到 “ named ” 行为发生改变的地方：

```
# grep named /var/log/messages
```


日志概述

发布时间 :2007-01-27 17:39:43

以下内容概括了通常位于 /var/log 并由 syslog 维护的日志文件：

messages：来自一般系统程序和守护程序的信息性与错误消息

secure：认证消息与错误，为了额外的安全性而与“message”分隔

maillog：与邮件相关的消息与错误

cron：与 cron 相关的消息与错误

spooler：UUCP 和与新闻相关的消息与错误

syslog.conf

发布时间 :2007-01-27 17:41:26

事实上，现在是研究 syslog 配置文件 /etc/syslog.conf 的时候了。（注：如果您没有 syslog.conf，为了获得信息请继续阅读，不过您可以使用另一个 syslog 守护程序）。通过浏览那个文件，我们发现以上提到的每个常见日志文件都有项，可能还有其它项。该文件的格式为 facility.priority action，这些字段的定义如下：

facility

指定产生消息的子系统。facility 有效的关键字有 auth、authpriv、cron、daemon、kern、lpr、mail、news、syslog、user、uucp 以及 local0 到 local7。

priority

指定消息的最低严重性，即此优先级和高于此优先级的消息将由这个规则匹配。priority 的有效关键字有 debug、info、notice、warning、err、crit、alert 和 emerg。

action

action 字段可以是文件名、tty（如 /dev/console）、以 @ 为前缀的远程机器、以逗号分隔的用户列表，或是 * 以表明向所有登录用户发送消息。最常见的操作是一个简单的文件名。

syslog.conf

发布时间 :2007-01-27 17:41:44

事实上，现在是研究 syslog 配置文件 /etc/syslog.conf 的时候了。（注：如果您没有 syslog.conf，为了获得信息请继续阅读，不过您可以使用另一个 syslog 守护程序）。通过浏览那个文件，我们发现以上提到的每个常见日志文件都有项，可能还有其它项。该文件的格式为 facility.priority action，这些字段的定义如下：

facility

指定产生消息的子系统。facility 有效的关键字有 auth、authpriv、cron、daemon、kern、lpr、mail、news、syslog、user、uucp 以及 local0 到 local7。

priority

指定消息的最低严重性，即此优先级和高于此优先级的消息将由这个规则匹配。priority 的有效关键字有 debug、info、notice、warning、err、crit、alert 和 emerg。

action

action 字段可以是文件名、tty（如 /dev/console）、以 @ 为前缀的远程机器、以逗号分隔的用户列表，或是 * 以表明向所有登录用户发送消息。最常见的操作是一个简单的文件名。

重新装入和附加信息

发布时间 :2007-01-27 17:42:14

希望这个配置文件的概述有助于您体验 syslog 系统的长处。您应该在进行更改之前阅读 `syslog.conf(5)` 手册页来获得更多的信息。另外，`syslogd(8)` 手册页提供了更详细的信息。

请注意：在对该配置文件的更改生效前，您需要向 syslog 守护程序通知所做的更改。向它发送 SIGHUP 是个正确的办法，您可以用 `killall` 命令轻松地做到这一点：

```
# killall -HUP syslogd
```

安全性注释

发布时间 :2007-01-27 17:42:43

您应该清楚如果 syslogd 写的日志文件还不存在的话，程序将创建它们。无论您当前的 umask 如何设置，该文件将被创建为可被所有用户读取。如果您关心安全性，那么您应该用 chmod 命令将该文件设置为仅 root 用户可读写。此外，可以用适当的许可权配置 logrotate 程序（在下面描述）以创建新的日志文件。syslog 守护程序始终会保留现有日志文件的当前属性，因此一旦创建了文件，您就不需要担心它。

logrotate

发布时间 :2007-01-27 17:43:17

/var/log 中的日志文件将随时间而变大，并有可能填满文件系统。建议利用 logrotate 这样的程序来管理日志的自动归档。logrotate 程序通常作为日常 cron 作业运行，并且可以配置为对日志文件执行循环、压缩、除去或发送电子邮件等操作。

例如，logrotate 的缺省配置会每周循环日志，保留 4 周的备份日志（通过在文件名后附加序号），并且压缩备份日志以节省空间。另外，还可以将该程序配置成将 SIGHUP 发送到 syslogd，这样守护程序将注意到现在为空的日志文件，并会适当地给它们附加信息。

有关 logrotate 的更多信息，请参阅 logrotate(8) 手册页，它包含该程序的描述以及配置文件的语法。

高级主题 — klogd

发布时间 :2007-01-27 17:43:47

在结束 syslog 的讨论之前，我想为渴望了解更多知识的读者提及几个高级主题。在您尝试理解与 syslog 相关的主题时，这些提示会使您免去一些麻烦。

首先，syslog 守护程序实际上是 sysklogd 软件包的一部分，这个软件包有另一个守护程序，名为 klogd。klogd 的任务是接收来自内核的信息和错误消息，然后将它们传递到 syslogd 进行分类和日志记录。klogd 接收到的消息和您可以使用 dmesg 命令获得的消息完全相同。区别是 dmesg 打印内核中环形缓冲区（ring buffer）的当前内容，而 klogd 将消息传递到 syslogd 使它们不会因环形缓冲区被覆盖而丢失。

高级主题 — 替代日志记录器

发布时间 :2007-01-27 17:44:17

其次，标准 `sysklogd` 包有替代选项。这些替代选项试图比 `sysklogd` 更有效、更易于配置，而且功能尽可能更丰富。`Syslog-ng` 和 `Metalog` 似乎是较受欢迎的替代项中的几个；如果您觉得 `syslogd` 达不到您所需的功能级别，您可以研究它们。

最后，您可以在脚本中使用 `logger` 命令记录消息。请参阅 `logger(1)` 手册页以获得更多信息。

Linux资源汇集

海量 Linux 技术文章

发布时间 :2006-11-24 16:50:29

下面是linux技术文章快速入口。需要联网：

[Linux 技术交流](#)

<http://www.linuxdiyf.com/bbs/forum-3-1.html>

[Linux 应用](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=1>

[Linux 安装及学习指导](#)

<http://www.linuxdiyf.com/bbs/forum-45-1.html>

[Linux 系统安装](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=11>

[Linux 学习指导](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=12>

[Linux 软件安装](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=45&filter=type&typeid=13>

[shell](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=3>

[Linux 壁纸](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=4>

[红旗](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=5>

[Redhat](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=6>

[SuSE](#)

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=7>

Linux 认证

<http://www.linuxdiyf.com/bbs/forumdisplay.php?fid=3&filter=type&typeid=9>

Linux下载分享{酷件、书籍、视频分享 }

<http://www.linuxdiyf.com/bbs/forum-6-1.html>

服务器应用

<http://www.linuxdiyf.com/bbs/forum-7-1.html>

数据库应用

<http://www.linuxdiyf.com/bbs/forum-8-1.html>

Linux 编程与内核

<http://www.linuxdiyf.com/bbs/forum-9-1.html>

UniX 技术文章

<http://www.linuxdiyf.com/bbs/forum-32-1.html>

Linux 业界声音、新闻

<http://www.linuxdiyf.com/bbs/forum-11-1.html>

Linux 人才招聘信息

<http://www.linuxdiyf.com/bbs/forum-46-1.html>

网络转载，感谢原创作者！

制作：红联Linux论坛

祝您阅读愉快！