为了增强文件系统的健壮性，Linux依靠日志文件系统。Ext3，作为Ext2文件系统的改进，就是一个日志文件系统的例子。

这种文件系统背后的基本思想是维护一个日志，该日志顺序记录所有文件系统操作。通过顺序写出文件系统数据或元数据（i节点，超级块等）的改动，该操作不必忍受随机磁盘访问时磁头移动带来的开销。最后，这些改动将被写到适当的磁盘地址，而相应的日志项可以被丢弃。如果系统崩溃或电源故障在改动提交之前发生，那么在重启动过程中，系统将检测到文件系统没有被正确地卸载。然后系统遍历日志，并执行日志记录所描述的文件系统改动。

Ext3设计成与Ext2高度兼容，事实上，两个系统中所有的核心数据结构和磁盘布局都是相同的。此外，一个作为ext2系统被卸载的文件系统随后可以作为ext3系统被加载并提供日志能力。

日志是一个以环形缓冲器形式组织的文件。日志可以存储在主文件系统所在的设备上也可以存储在其他设备上。由于日志操作本身不被日志记录，这些操作并不是被日志所在的ext3文件系统处理的，而是使用一个独立的日志块设备（Journaling Block Device，JBD）来执行日志的读/写操作。

JBD支持三个主要数据结构：日志记录、原子操作处理和事务。一个日志记录描述一个低级文件系统操作，该操作通常导致块内变化。鉴于系统调用（如write）包含多个地方的改动——i节点、现有的文件块、新的文件块、空闲块列表等，所以将相关的日志记录按照原子操作分成组。Ext3将系统调用过程的起始和结束通知JBD，这样JBD能够保证一个原子操作中的所有日志记录或者都被应用，或者没有一个被应用。最后，主要从效率方面考虑，JBD将原子操作的汇集作为事务对待。一个事务中日志记录是连续存储的。仅当一个事务中的所有日志记录都被安全提交到磁盘后，JBD才允许日志文件的相应部分被丢弃。

把每个磁盘改动的日志记录项写到磁盘可能开销很大，ext3可以配置为保存所有磁盘改动的日志或者仅仅保存文件系统元数据（i节点、超级块、位映射等）改动的日志。只记录元数据会使系统开销更小，性能更好，但是不能保证文件数据不会损坏。一些其他的日志文件系统仅仅维护关于元数据操作的日志（例如，SGI的XFS）。

4. /proc文件系统

另一个Linux文件系统是/proc（process）文件系统。其思想来自于Bell实验室开发的第8版UNIX，后来被4.4BSD和System V采用。不过，Linux在几个方面对该思想进行了扩充。其基本概念是为系统中的每个进程在/proc中创建一个目录。目录的名字是进程PID的十制数值。例如，/proc/619是与PID为619的进程相对应的目录。在该目录下是进程信息的文件，如进程的命令行、环境变量和信号掩码等。事实上，这些文件在磁盘上并不存在。当读取这些文件时，系统按需从进程中抽取这些信息，并以标准格式将其返回给用户。

许多Linux扩展与/proc中其他的文件和目录相关。它们包含各种各样的关于CPU、磁盘分区、设备、中断向量、内核计数器、文件系统、已加载模块等信息。非特权用户可以读取很多这样的信息，于是就可以通过一种安全的方式了解系统的行为。其中的部分文件可以被写入，以达到改变系统参数的目的。

10.6.4 NFS：网络文件系统

网络在Linux中起着重要作用，在UNIX中也是如此——自从网络出现开始（第一个UNIX网络是为了将新的内核从PDP-11/70转移到Interdata 8/32上而建立的）。本节将考察Sun Microsystem的NFS（网络文件系统）。该文件系统应用于所有的现代Linux系统中，其作用是将不同计算机上的不同文件系统连接成一个逻辑整体。当前主流的NFS实现是1994年提出的第3版。NFS第4版在2000年提出，并在前一个NFS体系结构上做了一些增强。NFS有三个方面值得关注：体系结构、协议和实现。我们现在将依次考察这三个方面，首先是简化的NFS第3版，然后简要探讨第4版所做的增强。

1. NFS体系结构

NFS背后的基本思想是允许任意选定的一些客户端和服务器共享一个公共文件系统。在很多情况下，所有的客户端和服务器都在同一个局域网中，但这并不是必需的。如果服务器距离客户端很远，NFS也可以在广域网上运行。简单起见，我们还是说客户端和服务器，就好像它们位于不同的机器上，但实际上，NFS允许一台机器同时既是客户端又是服务器。

每一个NFS服务器都导出一个或多个目录供远程客户端访问。当一个目录可用时，它的所有子目录也