

Linux文件系统的桌面应用 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/252/2021_2022_Linux_E6_96_87_E4_BB_c103_252846.htm 本文中要介绍一个所谓的"Linux 文件系统的守护神"，这是指一个能实时地观察 Linux 文件系统的变化情况的程序模块。能够实时的观察文件系统的变化情况，并做出及时的适当的反应，这对于应用 Linux 做桌面计算机系统来说，是十分的有趣，也是十分的重要的。本文还要介绍 Linux 文件系统的异步 I/O 的扩展。同样，这对于 Linux 系统的桌面应用也是关键的。

1、Linux 文件系统的守护神

传统的 Linux 文件系统呈现给用户程序的界面，确实是十分的干净利落。用户程序可以打开一个文件，向文件中线性的写入数据，从文件的某一位置开始，线性的读出数据，关闭一个文件，删除一个文件，创建一个文件，等等。请看，只有这么若干个简洁的操作原语，可是却能提供这么多丰富的应用。但是，我们注意到，用于访问 Linux 的文件系统的这些操作原语，并没有提供非常复杂的加锁解锁的功能。这是一件很奇妙的事情，如果来自不同的用户程序的请求发生了冲突怎么办呢？我们不妨走的再靠近一点，仔细的看看删除一个文件是怎样进行的。如果已经有一个用户程序在访问一个文件，而另外一个用户程序正好要删除这一个文件，这时会发生些什么呢？我们知道，Linux 的文件系统是基于所谓的 inode 的，每个文件都相伴有一个 inode。在 inode 中记录了关于这个文件的一些系统信息，比如文件的所有者，文件相关的一些权限记录，关于文件的若干个时间戳，等等。在内存中的 inode 还维持着一个关于自己的使用计数。每当一

个 inode 所代表的文件被打开一次，这个 inode 就把关于自己的使用计数加一。每当这个 inode 所代表的文件一被关闭，这个 inode 就把关于自己的使用计数减一。当用户程序删除一个文件的时候，相关的系统调用很快就返回到这个用户程序，告诉它，相应的文件已经被删除了。但是相应的 inode 还是保留在系统中，inode 首先要检查自己的使用计数，如果使用计数为零，那么 Linux Kernel 才可以真正的去删除这个文件。如果使用计数大于零，也就是说，还有其它的用户程序在访问这一个文件，那么 Linux Kernel 需要等待这些其他的用户程序一个个都完成对这一个文件的访问才行。也就是说，要等到这个 inode 的使用计数掉到零，才能真正的去删除这一个文件。我们可以设想一下，如果有一个 MP3 播放程序在播放一首 MP3 音乐，我们觉得它不好听，就到硬盘上找到这个文件，把它 rm 掉了。这时候，MP3 播放程序并不受到影响，还是可以继续播放这首 MP3 音乐，虽然这时候在文件系统上用 ls 已经找不到这个 MP3 音乐文件了。实际上，一直要到 MP3 播放程序停止播放这首 MP3 音乐，然后 Linux 文件系统才真正的从硬盘上删除这个 MP3 文件。这个经验和我们在 Windows 平台上遇到的截然不同。在 Windows 平台上，当我们试图在文件夹窗口中用鼠标点击右键菜单删除 Winamp 正在播放的一首 MP3 音乐的时候，Windows 系统会用一个弹出对话框告诉我们，这个文件正在被使用，没办法删除。Windows 系统的关于删除文件的这样一个解释，如果使用不当的话，会带来一个滑稽可笑的问题。我们可以设想一下，用户的一个 P2P 的文件共享程序提供了一个 MP3 文件以供别人下载，恰巧这个 MP3 音乐文件十分的热门，不断的有人

来下载，这个用户最终决定要节省一下带宽，想要把这个 MP3 音乐文件删除掉，但是 Windows 系统却不允许用户这样做，因为这个 P2P 的文件共享程序总是在使用这个 MP3 文件。用户要想删除这个文件，不得不先把 P2P 的文件共享程序给停下来！呵呵。但是 Linux 的文件系统的操作原语也有它自己的问题。我们知道，在一个 Linux Shell 的命令行上，先 rm，然后再 ls，非常的干净，被 rm 的文件没有了，被删除了。但是我们可以设想有一个图形界面的文件管理程序，当用户从 Shell 的命令行上 rm 掉一个文件的时候，这个图形界面的文件管理程序并没有收到任何人发给它的任何消息，它还以为什么都没有发生，被删除掉的文件还在那儿。这实在是很 U.G.L.Y. 啊。那么要想解决这个问题，一个明显的但是非常不好的办法，就是让一个后台进程 Daemon 每隔一个很短的时间间隔，就检查一下文件系统上这个目录的情况，看看有没有发生什么变化。这个办法的缺点真的是显而易见的，不但系统的性能受到影响，而且它的反应也还不是实时的。如果我们需要用户程序能够实时地了解文件系统上某一个目录的变化情况，从实时这个角度出发，显然，我们需要有一个中断机制。我们都知道，硬件中断能够实时地把系统某一个部件的情况反映给中央处理器，同样的，要想把位于系统内核中的文件系统的情况实时地反映给用户程序，我们也需要一个由操作系统内核到达用户进程的软件中断机制。熟悉 Linux 系统编程的读者朋友们立即就会想到，这个中断机制在 Linux 系统中早已就有了，这就是信号传递 signal。找到了信号传递这样一个中断用户进程的机制，一切似乎都已齐备，看来可以动手实现这样一个 Linux 文件系统的守护神，来实

时地监视文件系统的变化情况，并且及时地把消息通知给用户程序了。不过且慢，让我们搜索一下 Linux Kernel，看看是否有别人也在做同样的工作。哈哈，果不其然，原来这样一个实时地监视文件系统情况的机制早已在 Linux 内核中实现了。下面一段就是取自 Linux Kernel 文档的一段小小例程，说明了 Linux Kernel 中的 dnotify 功能的用法。dnotify 就是指 directory notification，监视文件系统上一个目录中的情况。

```
#define _GNU_SOURCE /* needed to get the defines */#include /* in glibc 2.2 this has the needed values defined */#include #include #include static volatile int event_fd ; // 信号处理例程 static void handler(int sig, siginfo_t *si, void *data){ event_fd = si->si_fd ; } int main(void){ struct sigaction act ; int fd ; // 登记信号处理例程 act.sa_sigaction = handler ; sigemptyset(&act, NULL) ; // 需要了解当前目录"."的情况 fd = open(".", O_RDONLY) ; fcntl(fd, F_SETSIG, SIGRTMIN) ; fcntl(fd, F_NOTIFY, DN_MODIFY|DN_CREATE|DN_MULTISHOT) ; /* we will now be notified if any of the files in "." is modified or new files are created */ while (1) { // 收到信号后，就会执行信号处理例程。 // 而 pause() 也就结束了。 pause() ; printf("Got event on fd=%d\n", event_fd) ; }}
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com