



图4-35 查找/usr/ast/mbox的过程

相对路径名的查找同绝对路径的查找方法相同，只不过是从当前工作目录开始查找而不是从根目录开始。每个目录都有`.`和`..`项，它们是在目录创建的时候同时创建的。`.`表项是当前目录的i节点号，而`..`表项是父目录（上一层目录）的i节点号。这样，查找`./dick/prog.c`的过程就成为在工作目录中查找`..`，寻找父目录的i节点号，并查询dick目录。不需要专门的机制处理这些名字。目录系统只要把这些名字看作普通的ASCII字符串即可，如同其他的名字一样。这里惟一的巧妙之处是`..`在根目录中指向自身。

4.6 有关文件系统的研究

文件系统总是比操作系统的其他部分吸引了更多的研究，现在也是这样。当标准的文件系统被完全理解后，现在还有很多后续的关于优化高速缓存管理的研究（Burnett等人，2002；Ding等人，2007；Gnaidy等人，2004；Kroeger和Long，2001；Pai等人，2000，以及Zhou等人，2001）。后续的工作还有关于新类型的文件系统，例如用户级别的文件系统（Mazières，2001），闪存文件系统（Gal等人，2005），日志文件系统（Prabhakaran等人，2005；以及Stein等人，2001），版本控制（versioning）文件系统（Cornell等人，2004），对等（peer-to-peer）文件系统（Muthitacharoen等人，2002），以及其他。Google文件系统也不寻常，因为它有极好的容错性能（Ghemawat等人，2003）。文件系统内不同的查询方法也是很有意义（Padiou和Ridoux，2003）。

另一个受到关注的领域是起源（provenance）——追踪数据的历史，包括它们来自哪里，谁拥有它们，以及它们是如何转换的（Muniswamy-Reddy等人，2006；以及Shah等人，2007）。这个信息可以以不同的方式加以运用。备份也一直受到关注（Cox等人，2002；以及Rycroft等人，2006），如同恢复的相关主题一样（Keeton等人，2006）。与备份有关的还有，设法保持数据几十年，并仍旧可以使用（Baker等人，2006；Maniatis等人，2003）。可靠性与安全性也是需要解决的问题（Greenan和Miller，2006；Wiens和Foley，2007；Wright等人，2007；以及Yang等人，2006）。最后，性能始终是一个值得研究的主题（Caudill和Gavrikovska，2006；Chiang和Huang，2007；Stein，2006；Wang等人，2006a；以及Zhang和Ghose，2007）。

4.7 小结

从外部看，文件系统是一组文件和目录，以及对文件和目录的操作。文件可以被读写，目录可以被创建和删除，并可从文件从一个目录移到另一个目录中。大多数现代操作系统都支持层次目录系统，其中，目录中还有子目录，子目录中还可以有子目录，如此无限下去。

而在内部看，文件系统又是另一番景象。文件系统的设计者必须考虑存储区是如何分配的，系统如何记录哪个块分给了哪个文件。可能的方案有连续文件、链表、文件分配表和i节点等。不同的系统有不同的目录结构。属性可以存在目录中或存在别处（比如，在i节点中）。磁盘空间可以通过位图的空闲表来管理。通过增量转储以及用程序修复故障文件系统的方法，可以提高文件系统的可靠性。文件系统的性能非常重要，可以通过多种途径提高性能，包括高速缓存、预读取以及尽可能仔细地将一个文件中