

第二章 嵌入式实时文件系统体系结构设计

2.1 嵌入式文件系统体系结构

嵌入式文件系统由于功能和作用与普通桌面操作系统的文件系统不同,导致了二者在体系结构上的很大差异。例如:在普通桌面操作系统中,文件系统不仅要管理文件,提供文件系统调用 API,还要管理各种设备,支持对设备和文件操作的一致性(即:象操作文件一样操作各种 I/O 设备)。在嵌入式文件系统中,这种规则发生了很大的变化——在某些情况下,嵌入式系统可以针对特殊的目的定制,特别是随着 ASOS (Application Specific Operating Systems, 为应用定制的嵌入式操作系统)的发展,对嵌入式操作系统的系统功能规整性和可伸缩性提出了更高的要求——有的应用为了节省系统开销可能不会使用文件系统,但是会用到 I/O 管理。在普通桌面操作系统中,这个问题是难以解决的。

所以我们不得不采取一种灵活的体系结构——对于操作系统来说,文件系统是一个独立的自包含的整体,可以灵活地加载和卸载,这一切操作对操作系统内核的功能没有任何影响。

基于以上的考虑,我们采用了如图 2-1 的嵌入式文件系统体系结构,该结构定义的文件系统从上到下有三个层次:

- 最上一层是实时文件系统实现。这是整个系统设计的核心,也是嵌入式文件系统中用户唯一可见的部分——POSIX 中文件系统的标准系统调用全部在该部分中得以实现。
- 第二层是高速缓冲区管理。它是文件系统的效率之源——高效的高速缓冲区管理算法能大幅度提高文件系统执行的效率;同时它要为文件系统的实现提供与硬件无关的统一接口,是文件系统结构规整性的基础。
- 最下面一层是设备驱动程序,由于涉及到具体的硬件,这一层的各个模块实现的差异很大,具体编程的数量和难度也是三部分中最大的。作为设备驱动程序的另一重要任务就是提供统一的设备驱动程序接口,这一实现我们将在第三章详细讨论。