

2.2 文件系统系统调用

根据 POSIX 定义, 文件系统应该提供的系统调用的最小功能子集如下:

```
signed16 dfs_open(signed8 *path, signed16 flag );
signed16 dfs_close(signed16 fd );
signed16 dfs_creat(signed8 *path, signed16 attrib );
signed16 dfs_delete (signed8 *path);
signed16 dfs_rename (signed8 *path1, signed8 *path2);
signed32 dfs_getfsize (signed16 fd );
signed16 dfs_read (signed16 fd, void *buffer, unsigned16
count );
signed16 dfs_write ( signed16 fd, void *buffer, unsigned16
count );
signed32 dfs_lseek (signed16 fd, signed32 foffset, signed16
origin );
signed16 dfs_getfattr ( signed8 *name, signed8 attrp );
signed16 dfs_setfattr ( signed8 *name, signed8 attrp );
signed16 dfs_rmdir ( signed8 *path );
signed16 dfs_mkdir ( signed8 *dir);
```

2.3 文件系统的实现方式

在这一节中我们给出各个模块实现方式的大致描述, 为接下来的三个章节制定轮廓, 而不讨论的实现细节。

首先是与文件存储设备相关的驱动程序层。实现设备驱动程序的首要任务是划定支持硬件的范围。在文件系统设计中, 将支持 IDE 硬盘、软盘、RAMDisk 和 DOC 等存储设备, 这是出于以下考虑: 选定硬盘不是因为它适合嵌入式系统使用, 而是 IDE 接口通过几十年的发展已经成为了一种标准, 不仅硬盘采用, 在其它类型的存储设备上也得到了广泛的应用, 如 DOM(disk on module); 软盘到现在为止仍然是广泛使用的嵌入式系统存储介质; RAM disk 可以使系统在没