FCB\* pre;//指向前一个FCB，第一个孩子的pre指向父亲

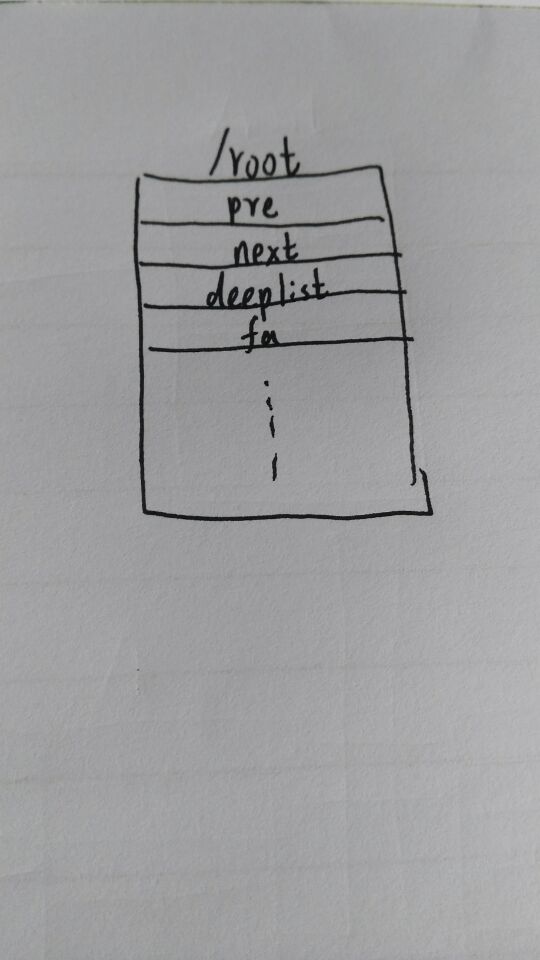
FCB\* next;//指向后一个FCB，最后一个孩子的next指向父亲

FCB\* deepQList;//指向第一个孩子FCB

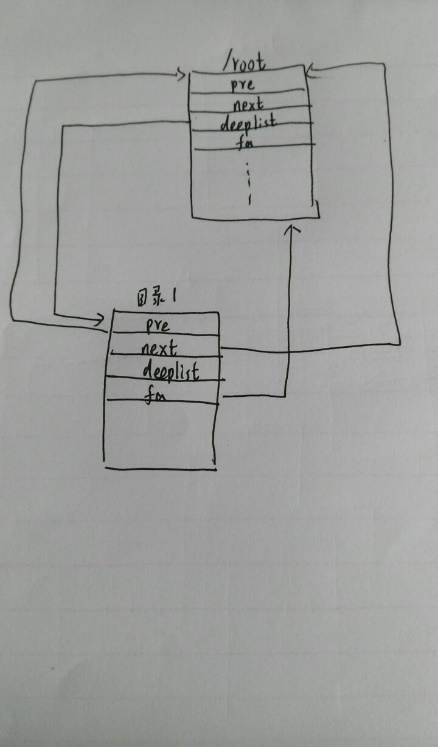
FCB\* fa;//指向父亲FCB

整体数据结构类似于一棵树，只是比树多了一些指针

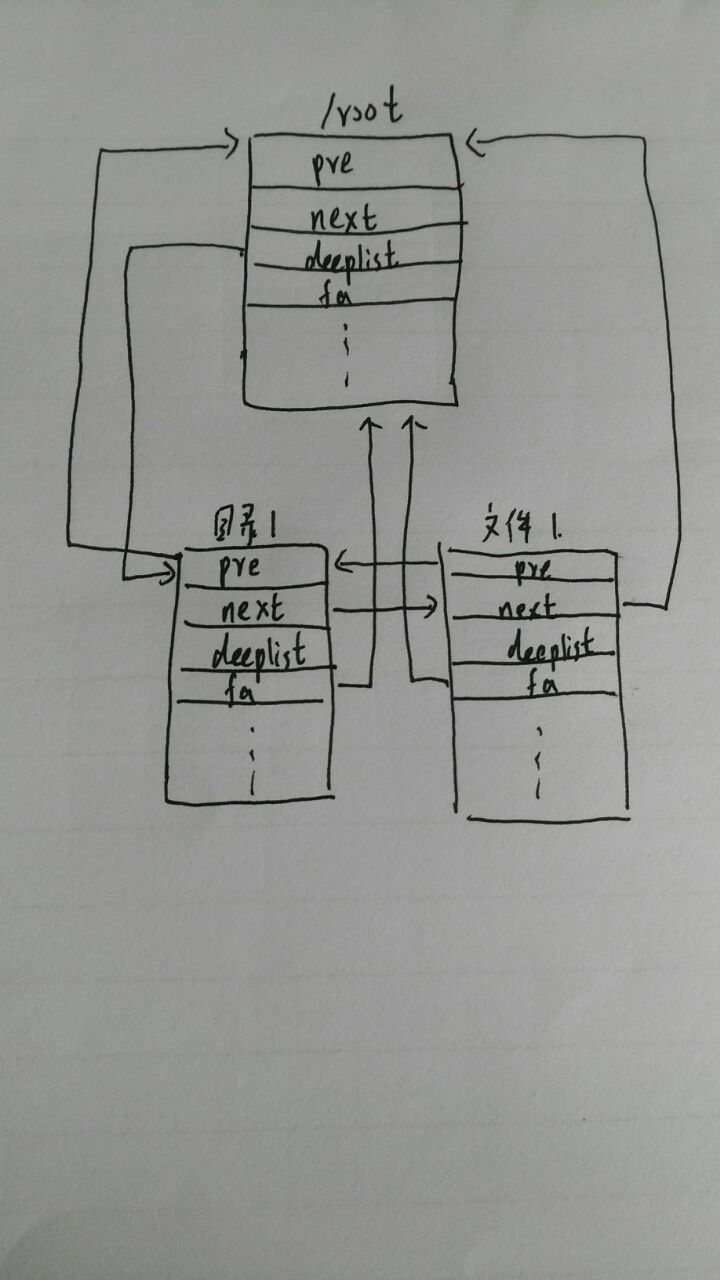
首先，第一次进入程序，程序初始化时，自动创建一个root目录（FCB），以后所有操作都在该目录下进行，

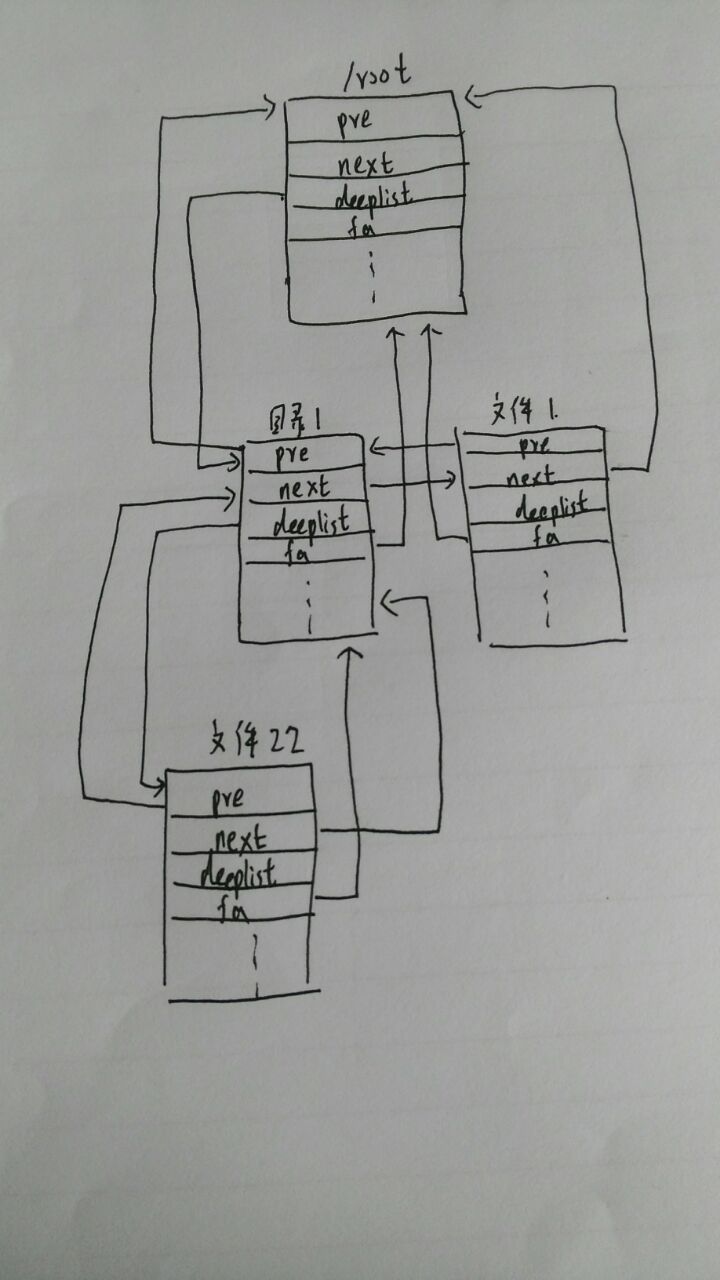
此时再root目录下创建一个1目录，调用创建目录函数mkdir，结构如下，同时，界面创建一个button,与改目录关联（button有个指向该目录的成员变量指针）

如果再在root目录下创建文件11，结构变为如下：

再在目录1下创建文件22

后面的删除，改名，等操作，都是对这颗树的操作。如改名，先遍历这棵树，找到要改名的FCB，然后修改他的成员变量name。

FCB只是记录了文件和目录的一些基本信息。

FAT表可以理解为int型数组，值为0代表未使用，为1代表已使用

每创建一个目录或文件，程序先扫描FAT表，看哪一块磁盘块为空，找到第一块为空的，更改为已使用。并在FCB中记录使用的是那一个磁盘块