

第六章

文件管理

方 钰



主要内容

- 6.1 文件系统概述
- 6.2 UNIX文件系统接口
- 6.3 UNIX文件系统的物理结构
- 6.4 UNIX文件系统的打开结构**
- 6.5 UNIX文件系统的目录管理
- 6.6 UNIX文件系统的读写操作



每个文件还有一个内存文件控制块Inode（内存索引节点）

```
class Inode
{
    ...;
    /* Functions */
public:
    void ReadI();    /* 根据Inode对象中的物理磁盘块索引表，读取文件数据 */
    void WriteI();   /* 根据Inode对象中的物理磁盘块索引表，将数据写入文件 */
    int  Bmap(int lbn);    /* 将文件的逻辑块号转换成对应的物理盘块号 */
    void OpenI(int mode);  /* 打开文件 */
    void CloseI(int mode); /* 关闭文件 */
    void IUpdate(int time); /* 更新外存Inode的最后的访问时间、修改时间 */
    void ITrunc();         /* 释放Inode对应文件占用的磁盘块 */
    void Clean();          /* 清空Inode对象中的数据 */
    void ICopy(Buf* bp, int inumber); /* 将外存Inode信息拷贝到内存Inode中 */
}
```




每个文件还有一个内存文件控制块Inode（内存索引节点）

```

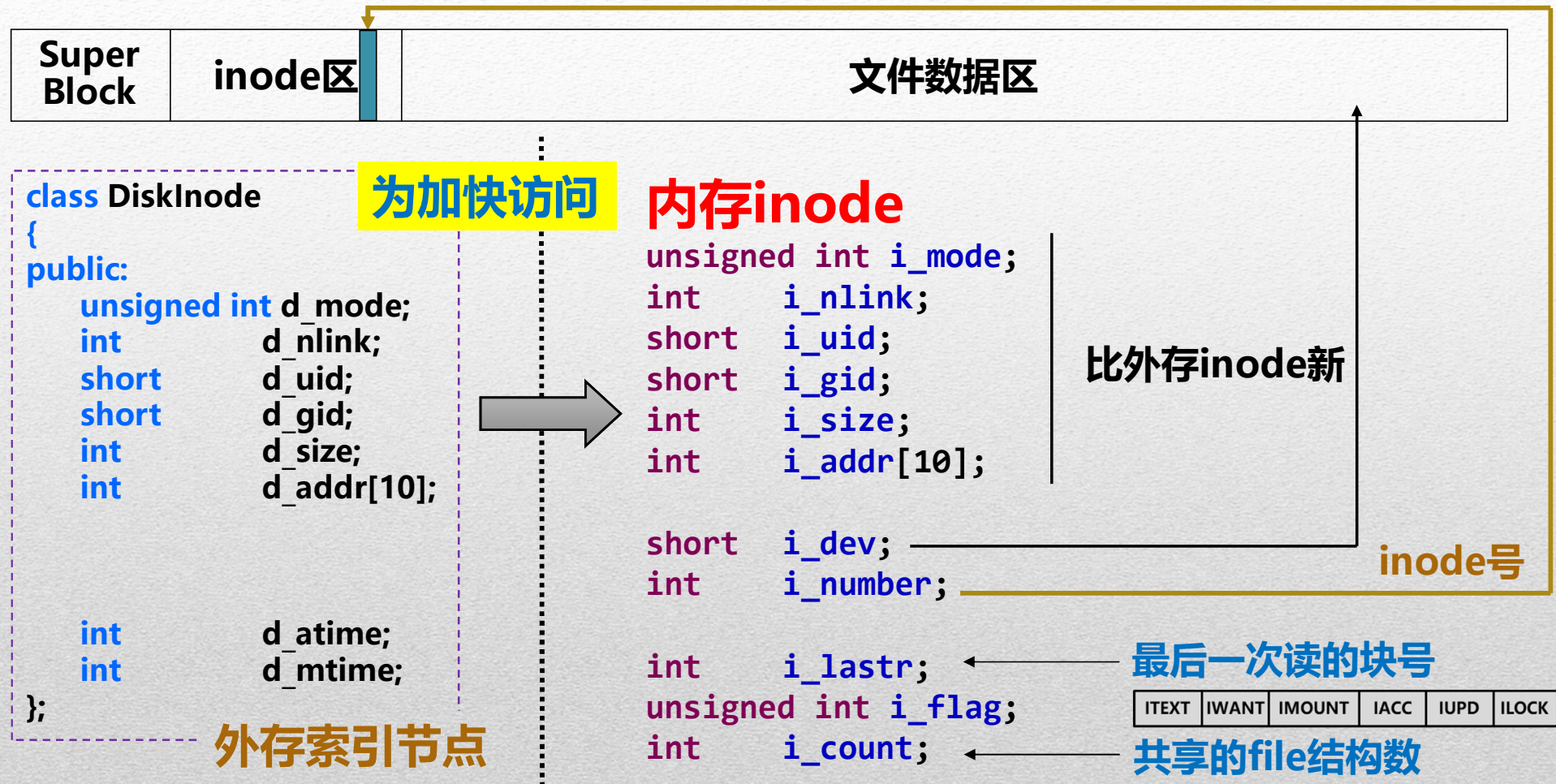
/* Members */
public:
    unsigned int i_flag;      /* 状态的标志位, 定义见enum INodeFlag */
    unsigned int i_mode;     /* 文件工作方式信息 */
    int i_count;             /* 引用计数 */
    int i_nlink;             /* 该文件在目录树中不同路径名的数量 */
    short i_dev;             /* 外存inode所在存储设备的设备号 */
    int i_number;            /* 外存inode区中的编号 */
    short i_uid;             /* 文件所有者的用户标识数 */
    short i_gid;             /* 文件所有者的组标识数 */
    int i_size;              /* 文件大小, 字节为单位 */
    int i_addr[10];          /* 文件逻辑块号和物理块号转换的基本索引表 */
    int i_lastr;             /* 存放最近一次读取文件的逻辑块号 */
};

```



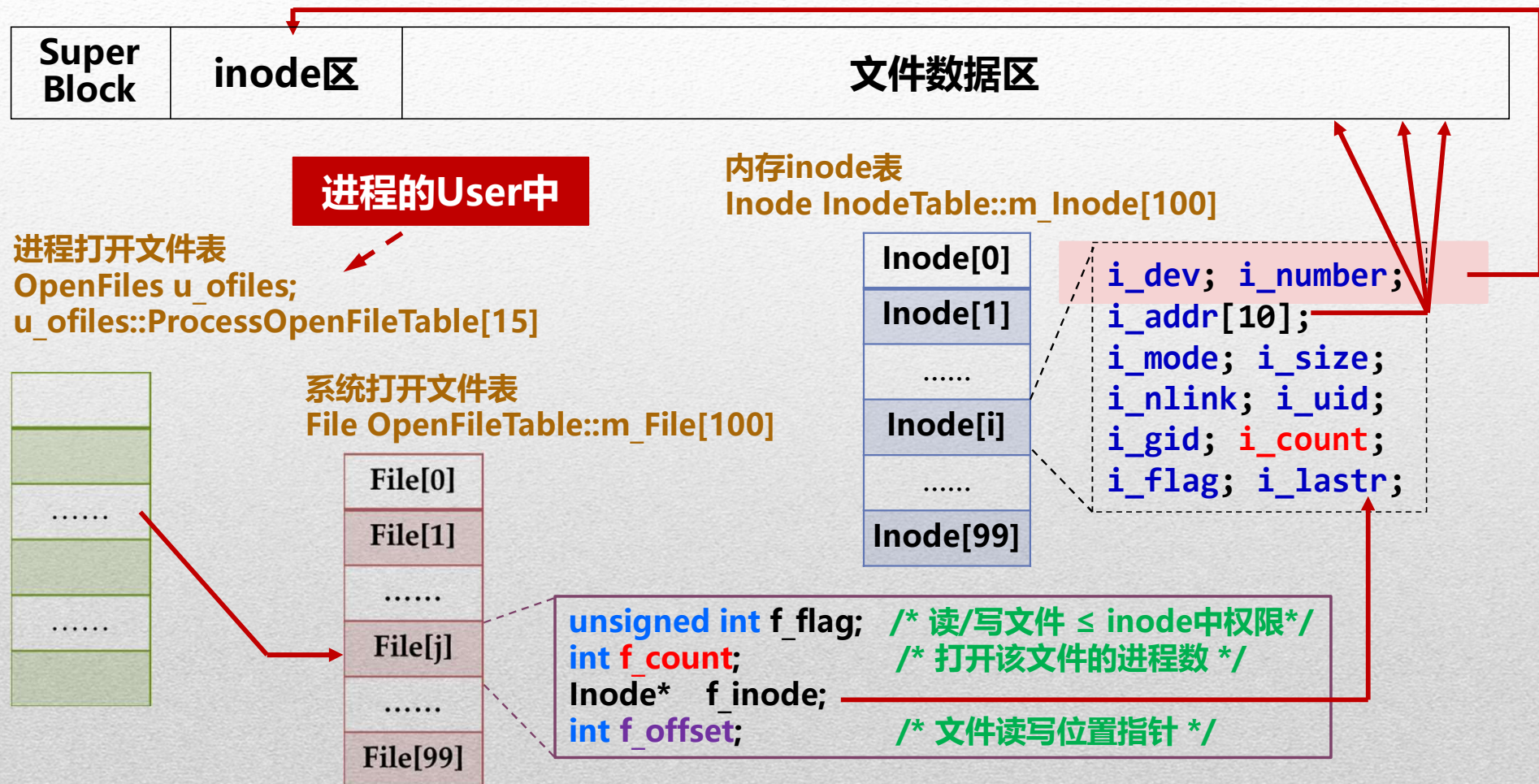


内存文件打开结构



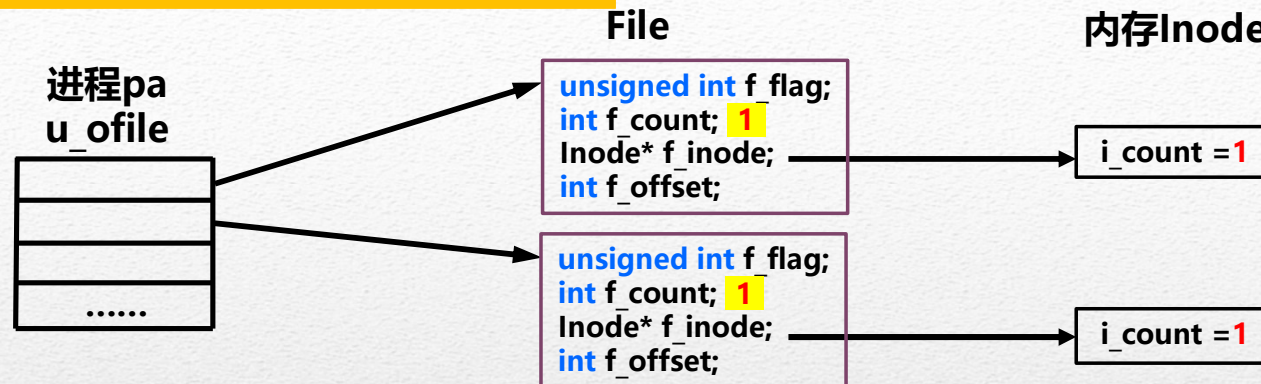


内存文件打开结构

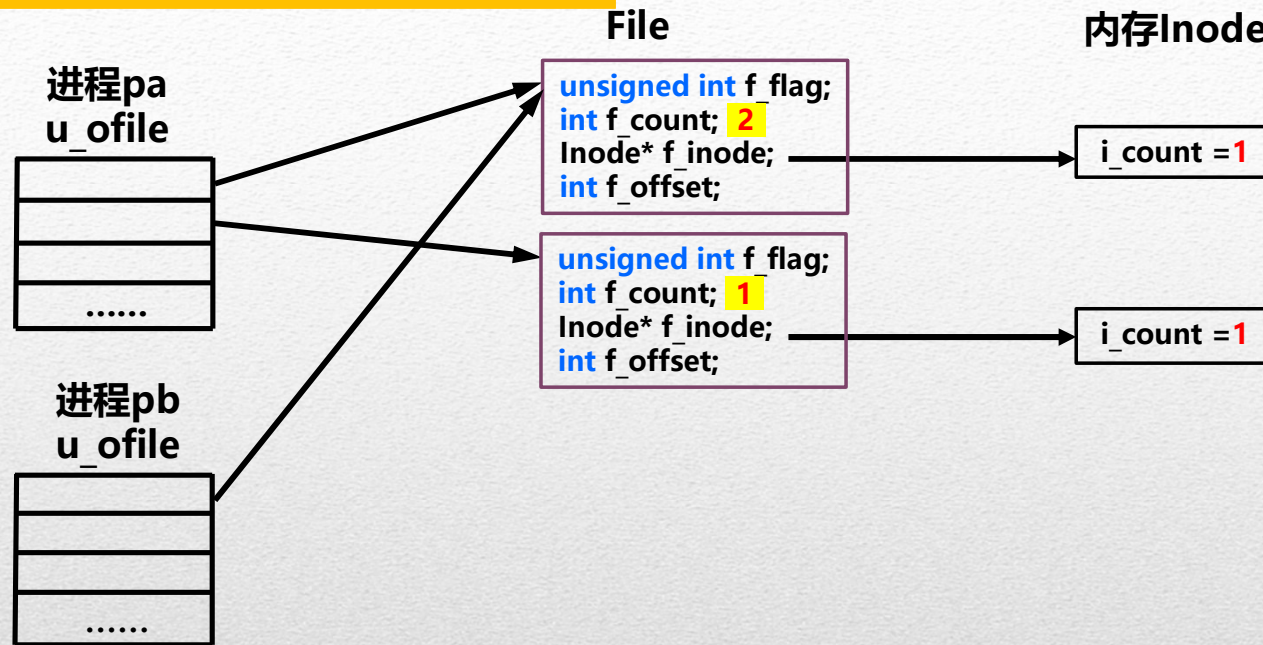




内存文件打开结构



内存文件打开结构

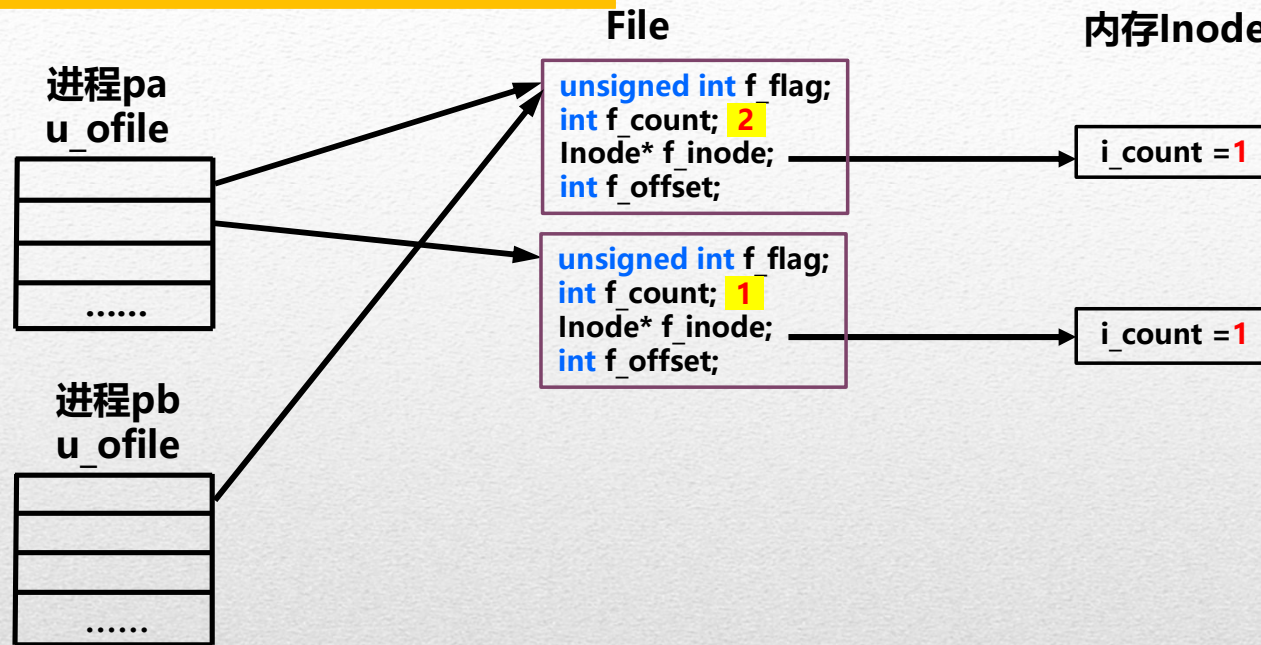


父子进程之间可共享一个打开文件及其读写指针。



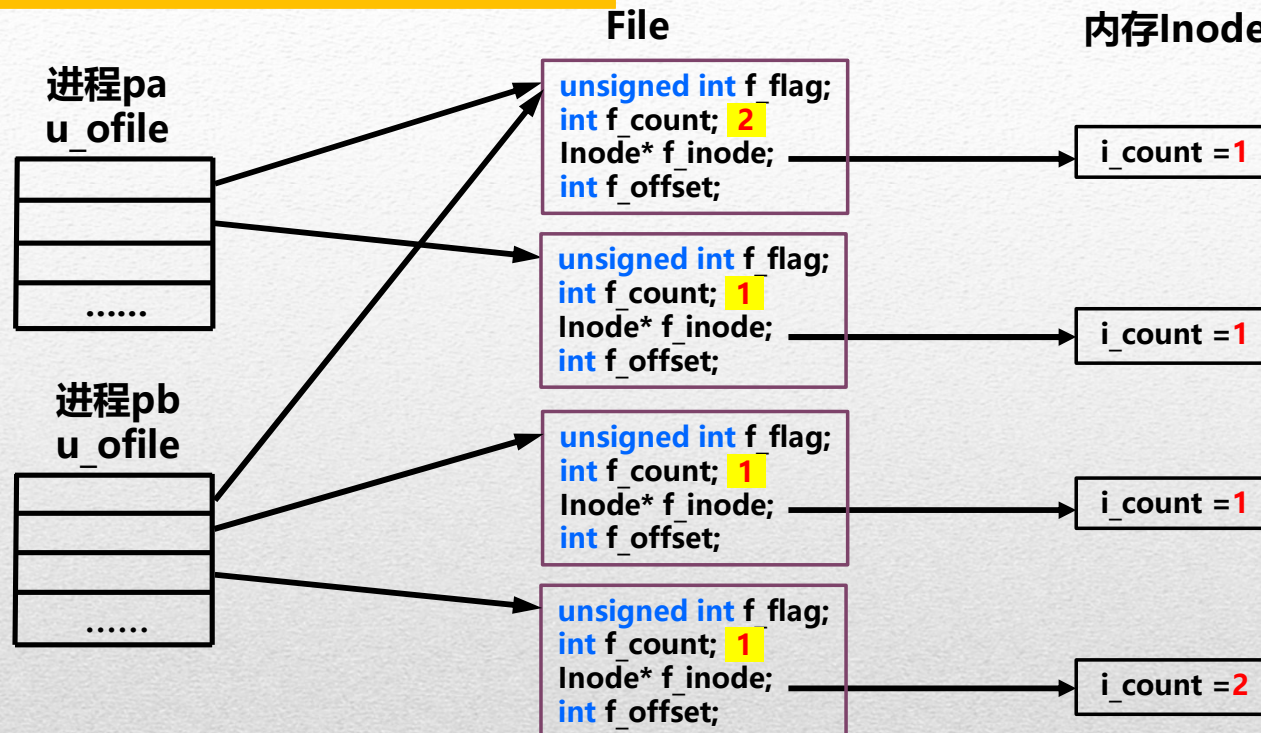


内存文件打开结构



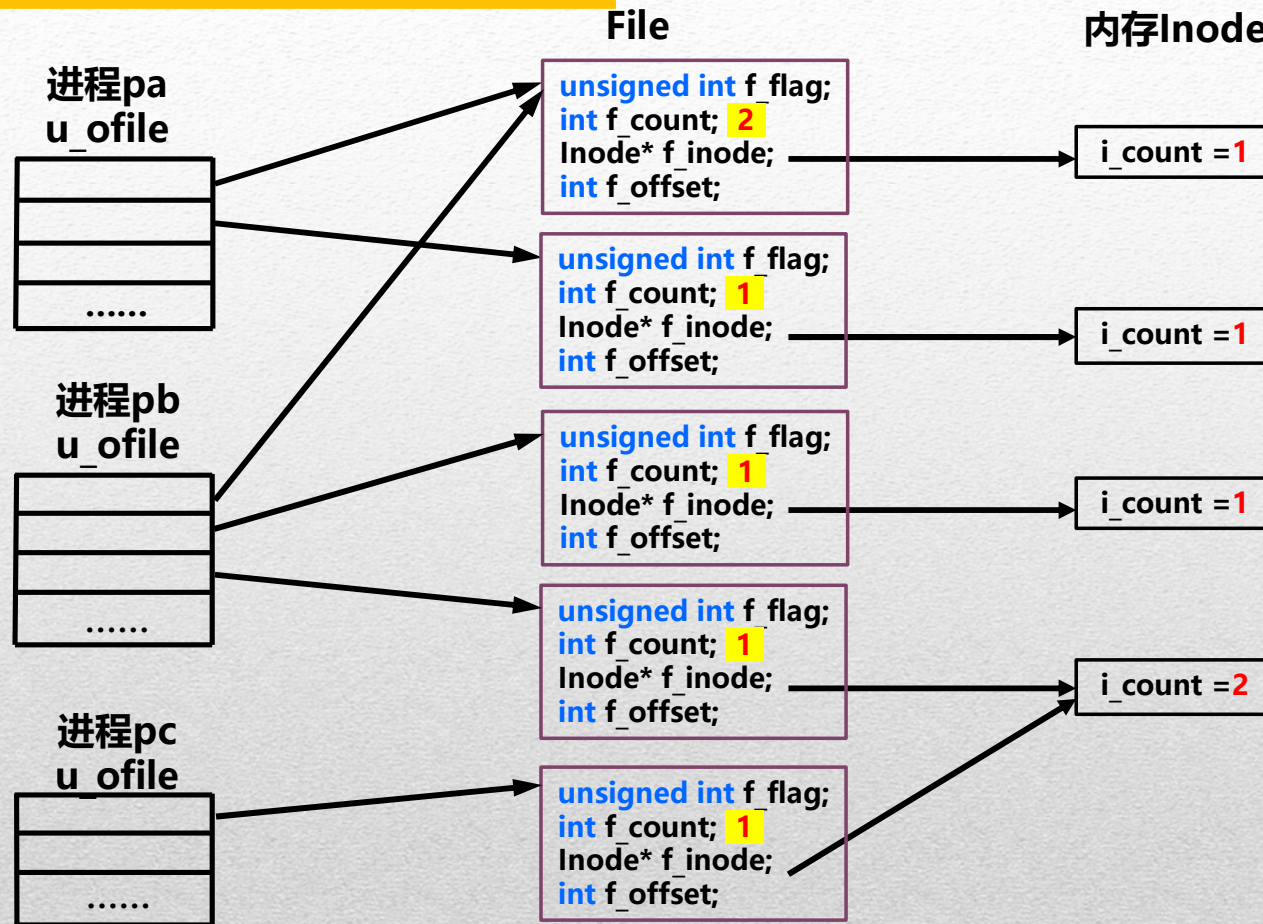


内存文件打开结构





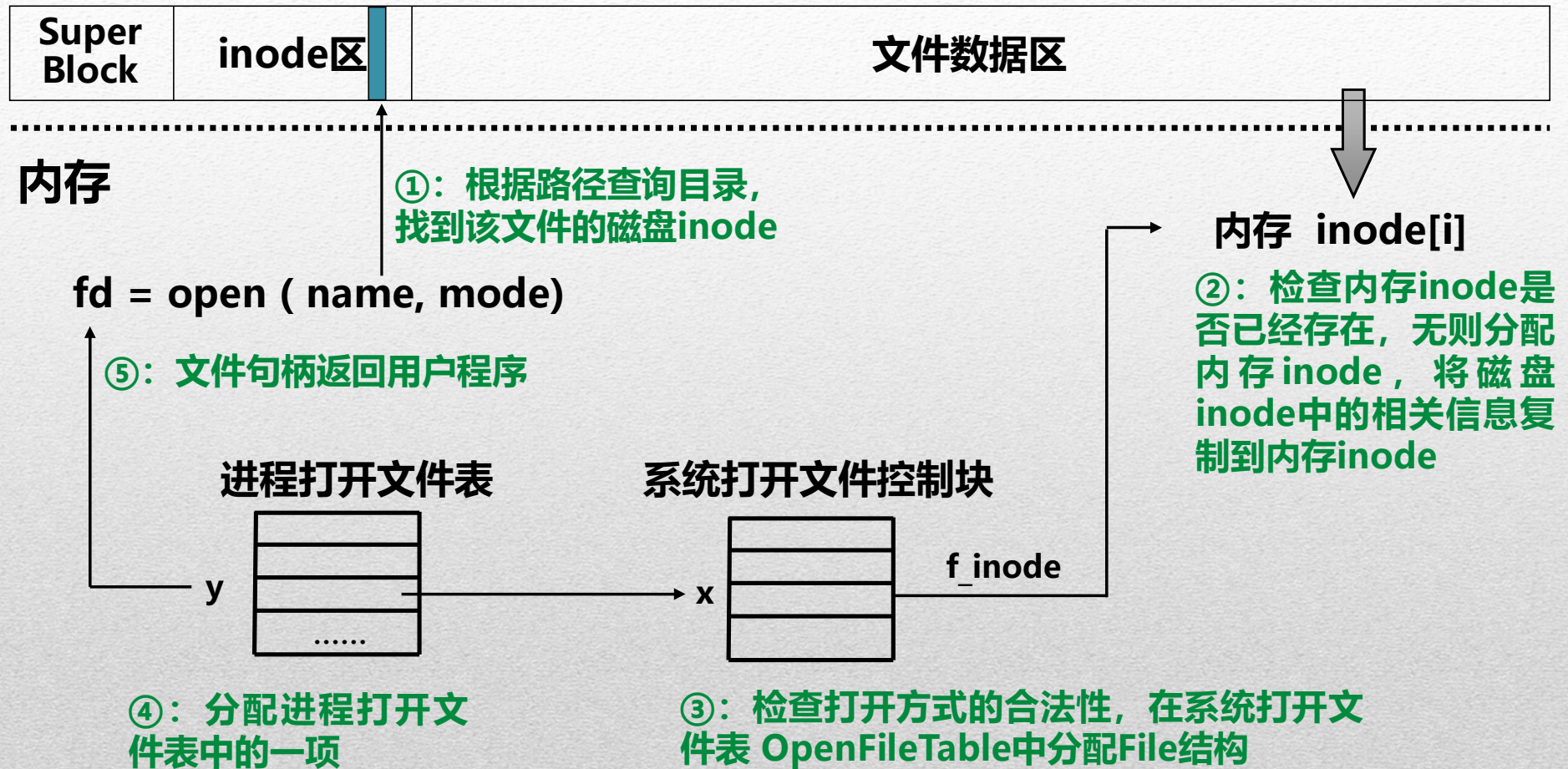
内存文件打开结构



多个进程以不同的读写权限和指针读写同一个磁盘文件

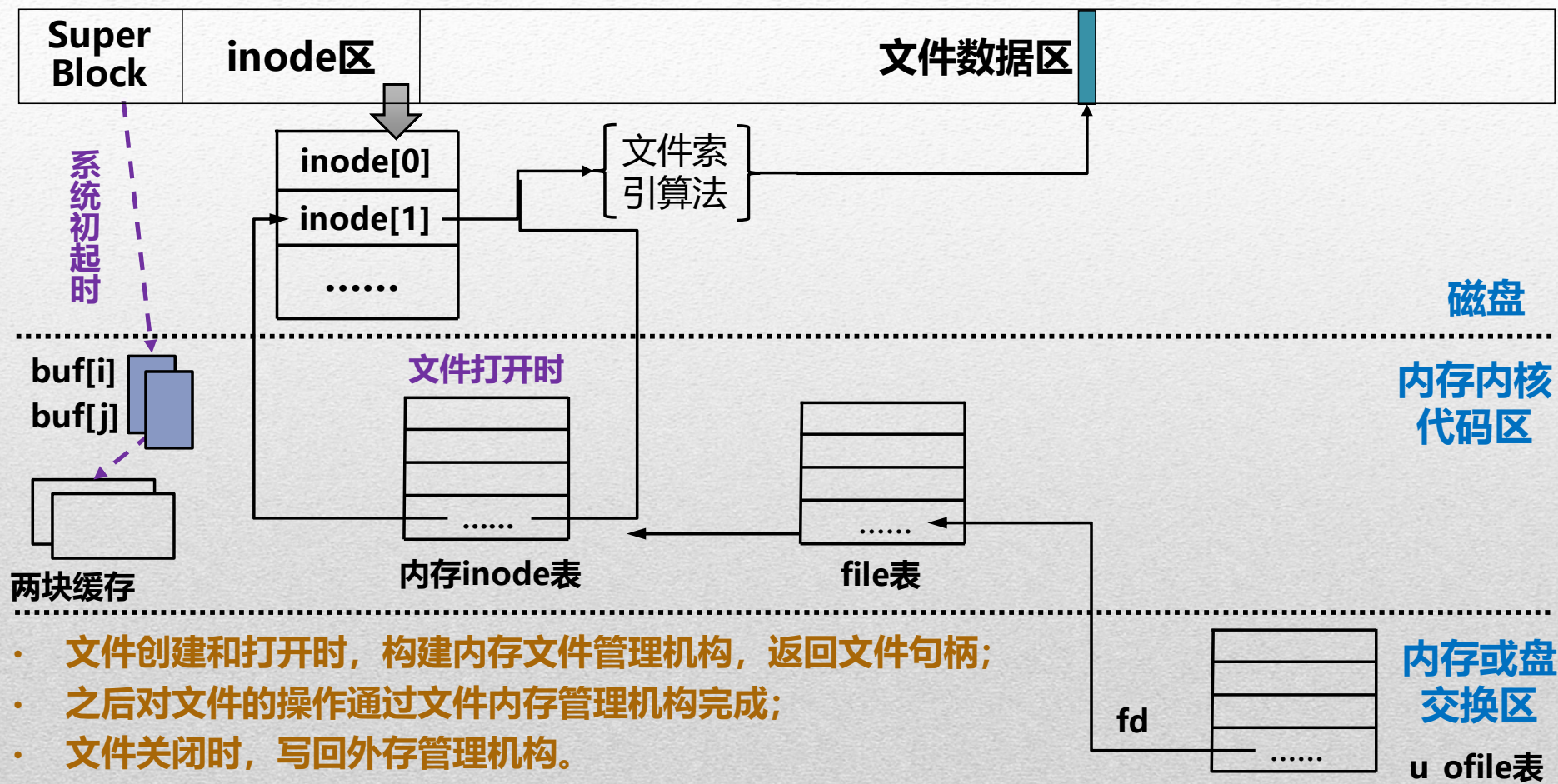


文件的打开过程





文件系统相关数据结构



- 文件创建和打开时，构建内存文件管理机构，返回文件句柄；
- 之后对文件的操作通过文件内存管理机构完成；
- 文件关闭时，写回外存管理机构。



主要内容

- 6.1 文件系统概述
- 6.2 UNIX文件系统接口
- 6.3 UNIX文件系统的物理结构
- 6.4 UNIX文件系统的打开结构
- 6.5 UNIX文件系统的目录管理**
- 6.6 UNIX文件系统的读写操作

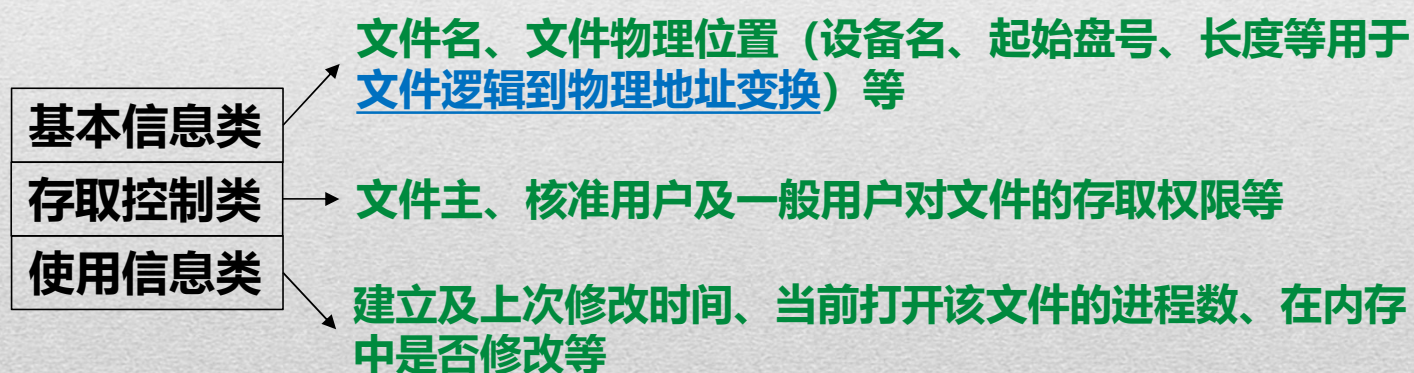


文件目录

文件目录 对大量的文件实施有效的管理。按文件名对所有文件进行管理，实现从文件符号名到文件实体的映射

目录管理要求 实现“**按名存取**”；提高目录的检索速度；文件共享；允许文件重名

文件控制块 (FCB) (文件与FCB一一对应)

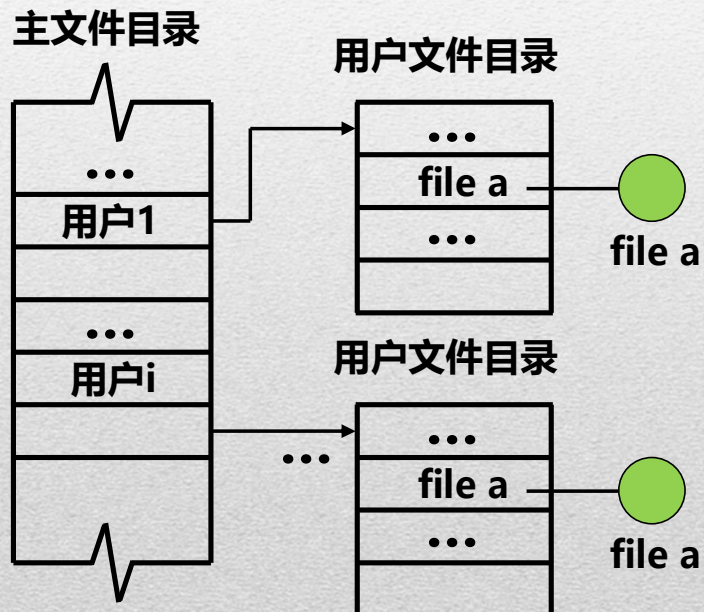


FCB的集合构成文件目录
(一个FCB就是一条目录项)，同样保存在外存，称为**目录文件**。

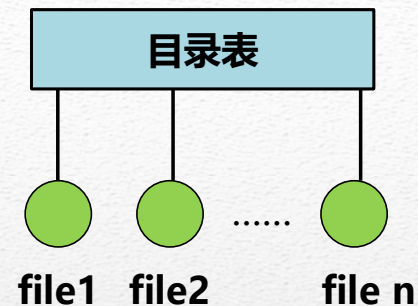
文件目录

目录结构

一级目录结构
二级目录结构
多级目录结构



- ① 检索速度提高
- ② 不同用户可以使用相同的文件名
- ③ 缺乏灵活性

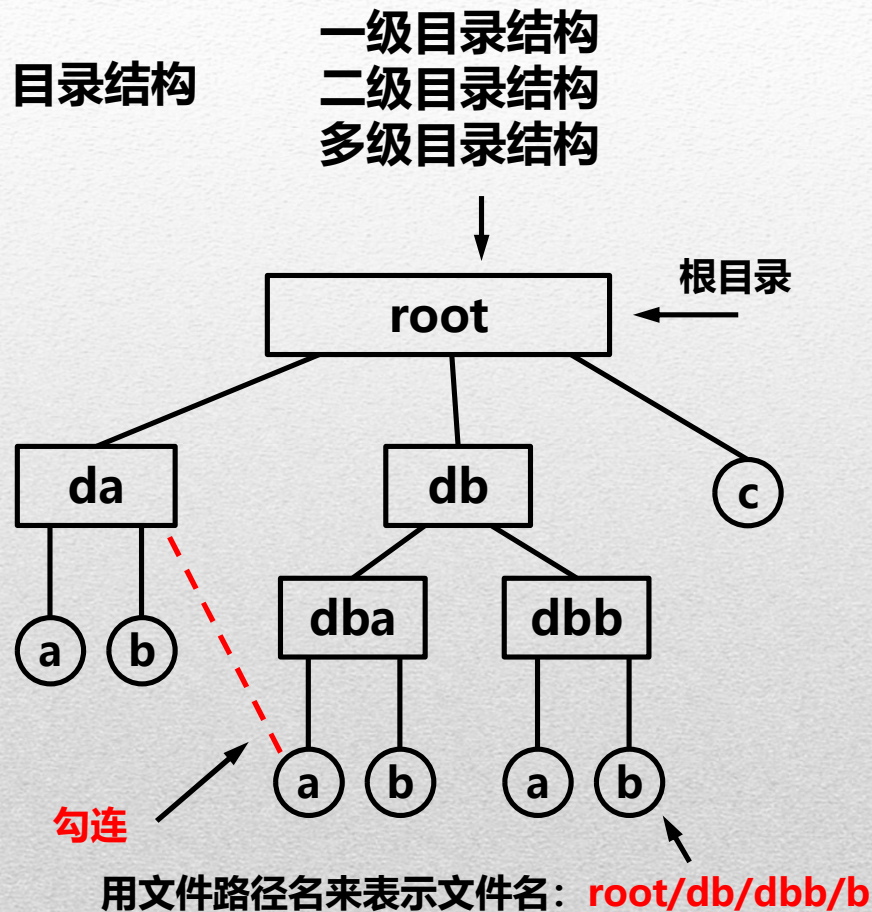


文件名	物理地址	文件说明	状态位
文件名1			
文件名2			
.....			

- ① 简单，可实现按名存取
- ② 文件不允许重名
- ③ 不便于对文件进行分组管理
- ④ 寻找目录项的过程比较冗长



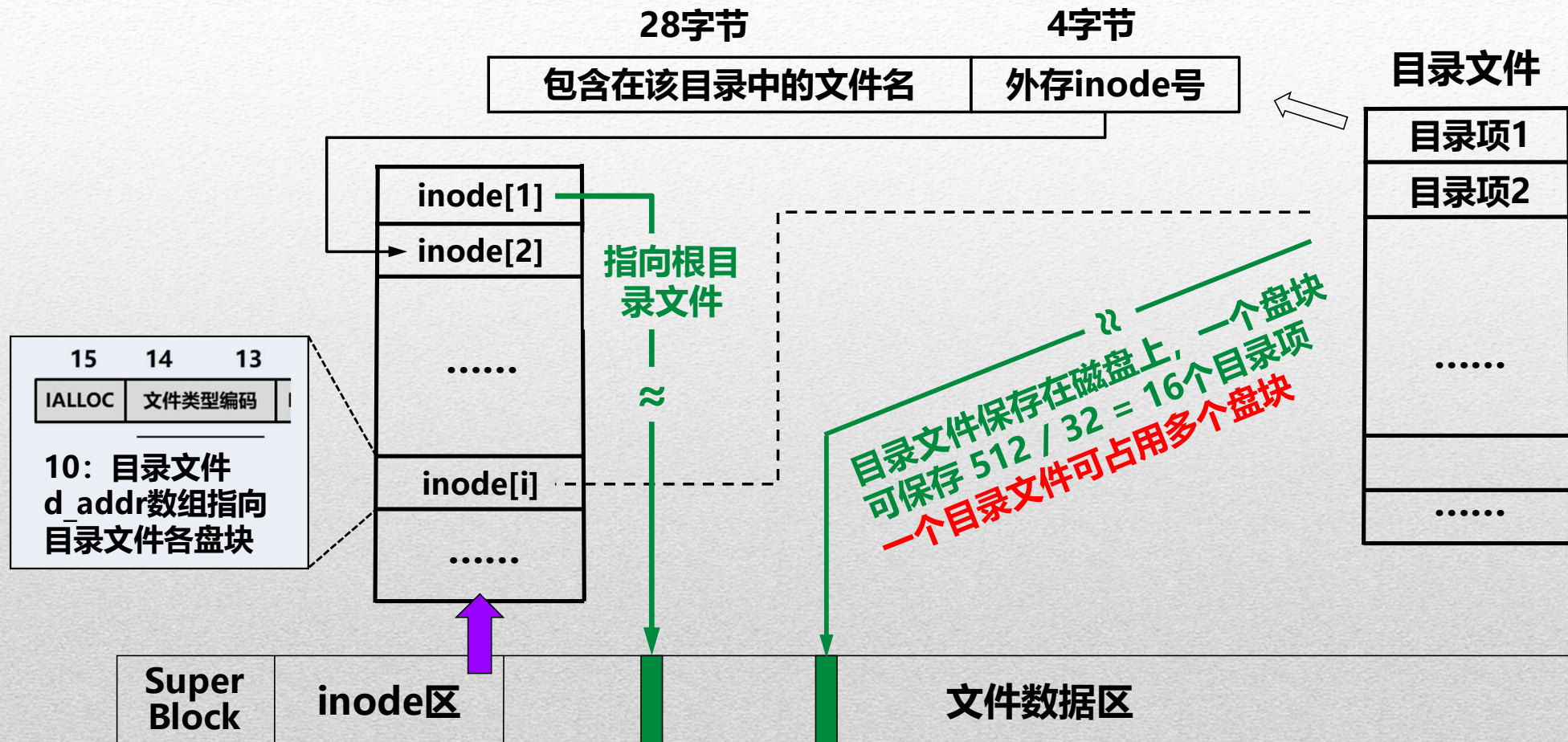
文件目录



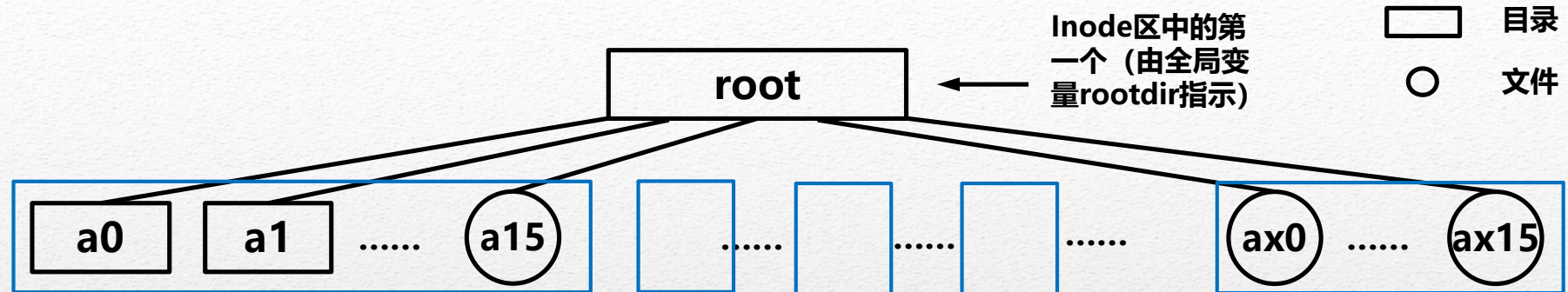
- **路径名**: 从根目录开始到任何数据文件存在的一条通路。在该路径上经过的全部目录文件名和数据文件名, 用“/”连接起来。
- **当前目录**: 可为每个进程设置一个“当前目录”, 又称为“工作目录”。进程对各文件的访问都相对与“当前目录”。
- **相对路径**: 从当前目录开始到数据文件构成的路径。
- **绝对路径**: 从根目录到数据文件构成的路径。



UNIX文件目录结构



UNIX文件目录结构

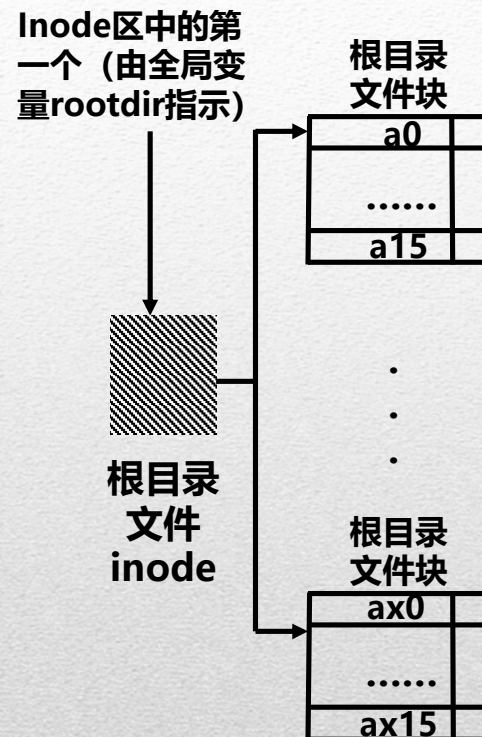


根目录文件

a0	
.....	
a15	
.....	
ax0	
.....	
ax15	



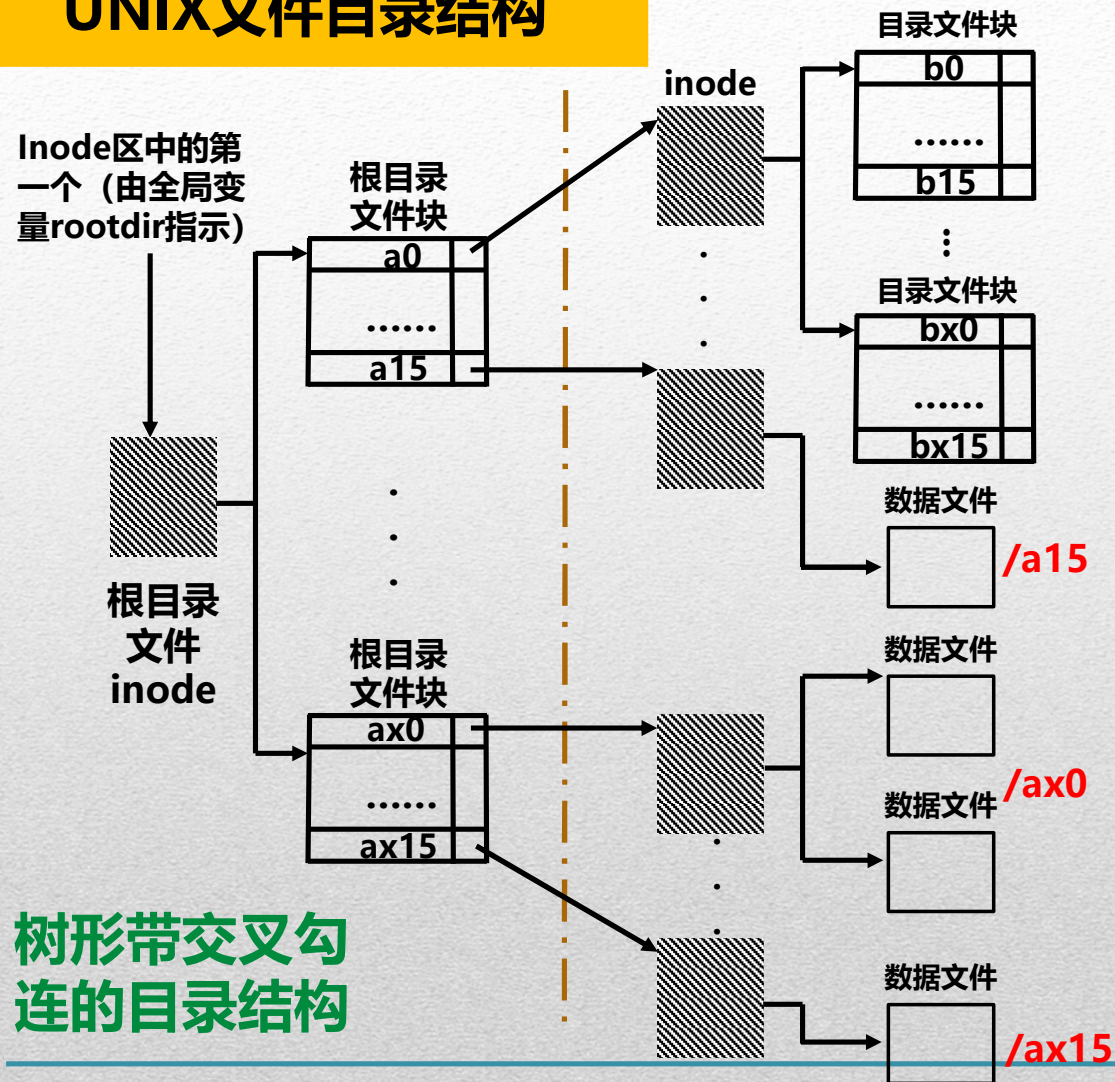
UNIX文件目录结构



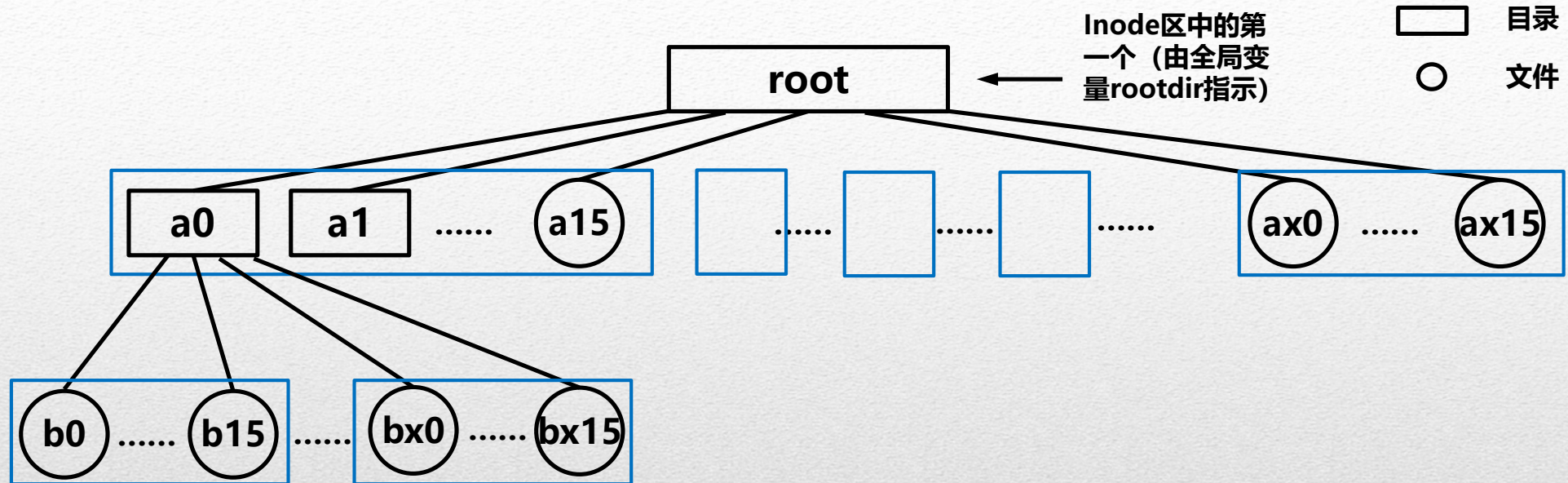
树形带交叉勾连的目录结构



UNIX文件目录结构



UNIX文件目录结构



UNIX文件目录结构



UNIX文件目录结构

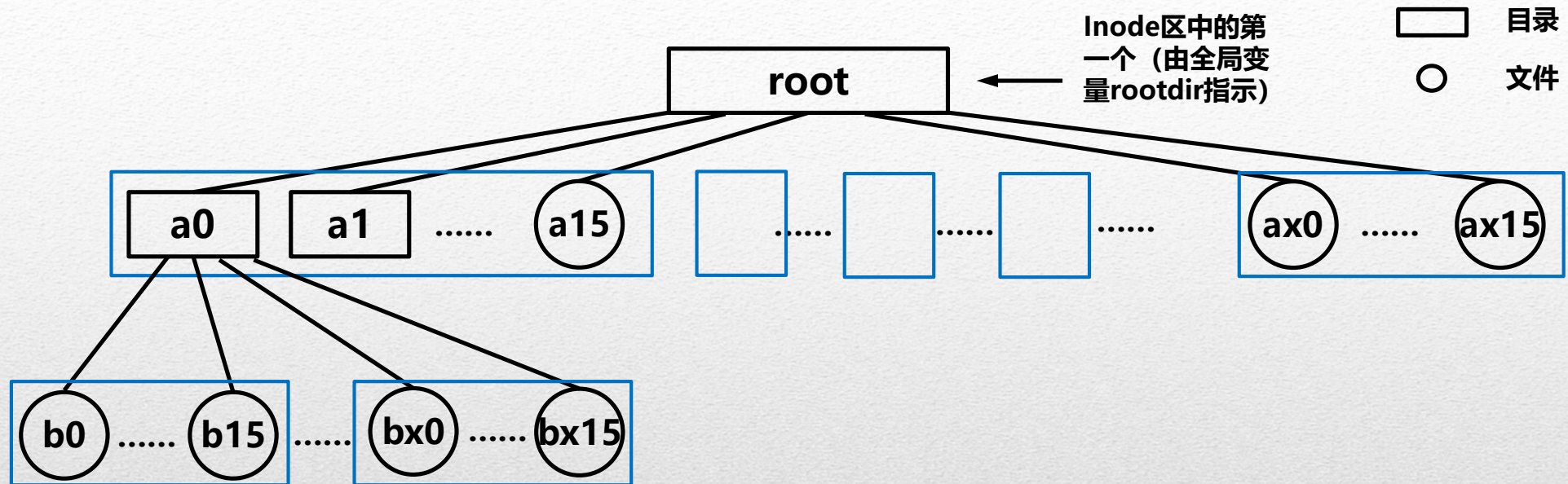


Inode区中的第一个 (由全局变量rootdir指示)



从逻辑结构上看，不同的路径对inode的地位是相同的。单独取消一个路径不能取消inode及其代表的文件实体。

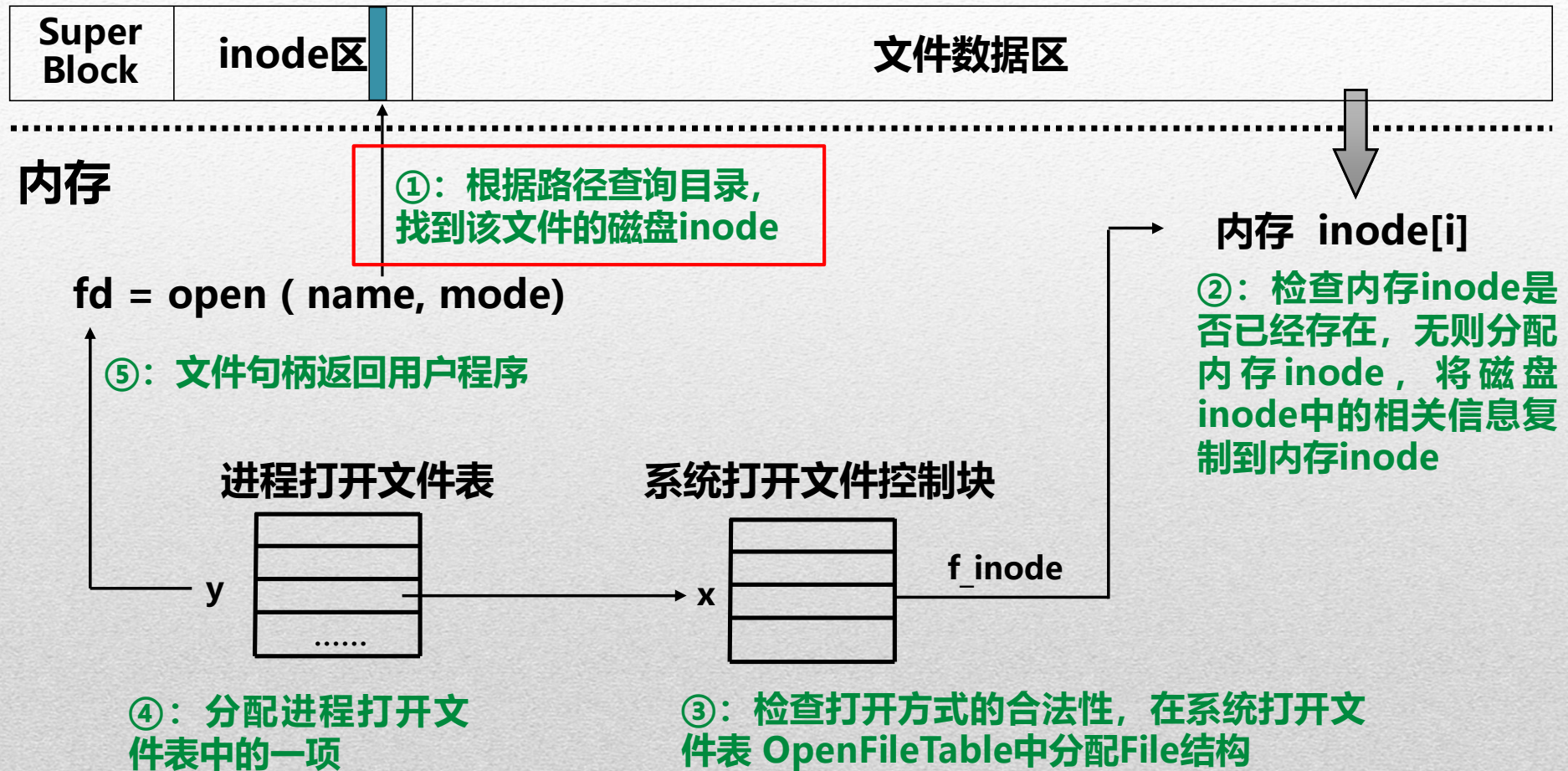
UNIX文件目录结构



1. 目录树中的**每一个节点对应一个文件**，具有**唯一的Inode**。
2. 如果是**目录节点**，Inode中标注为**目录文件**；如果是**文件节点**，Inode中标注为**数据文件或设备文件**。
3. 同一级的所有节点，在上一级目录文件中登记。

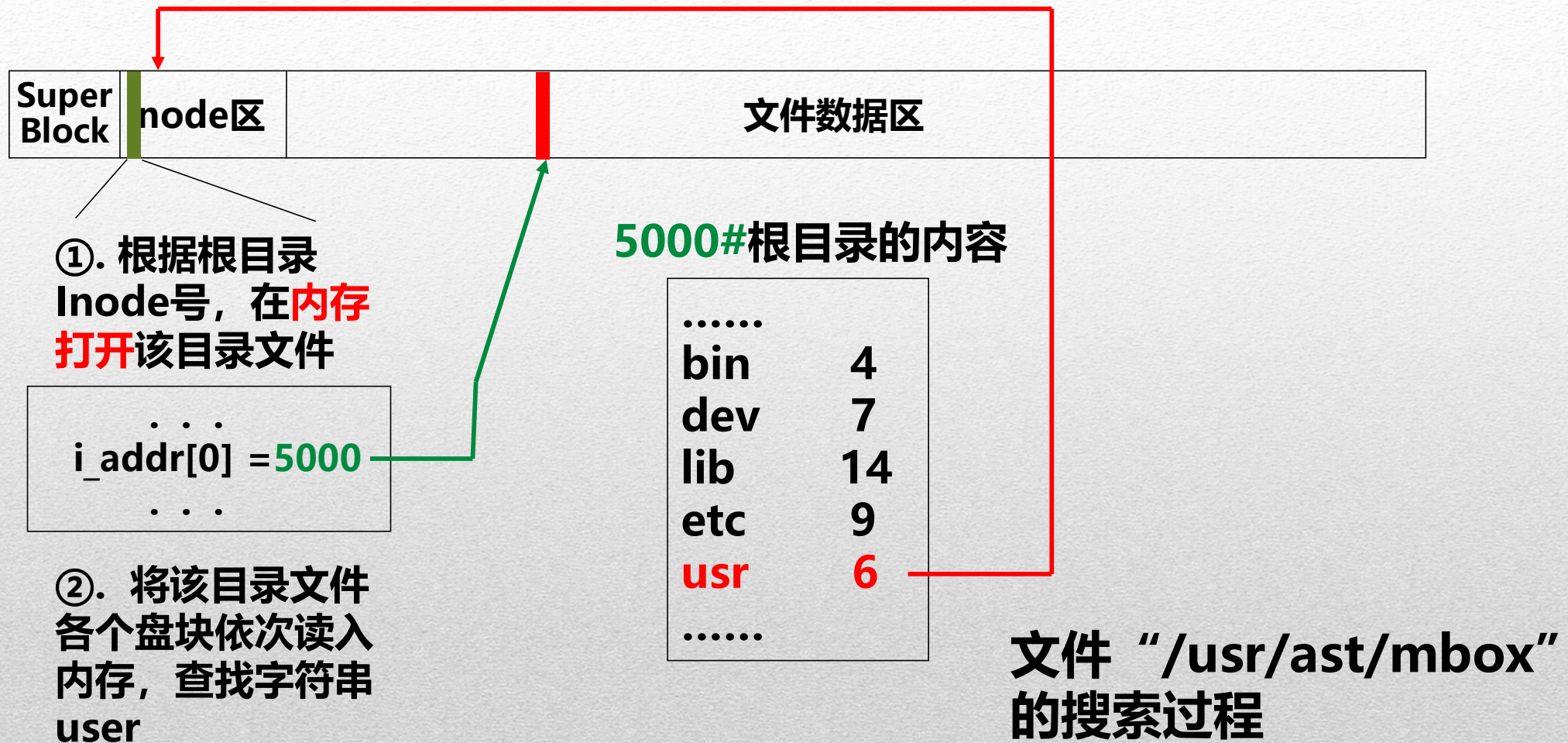


再看文件的打开过程



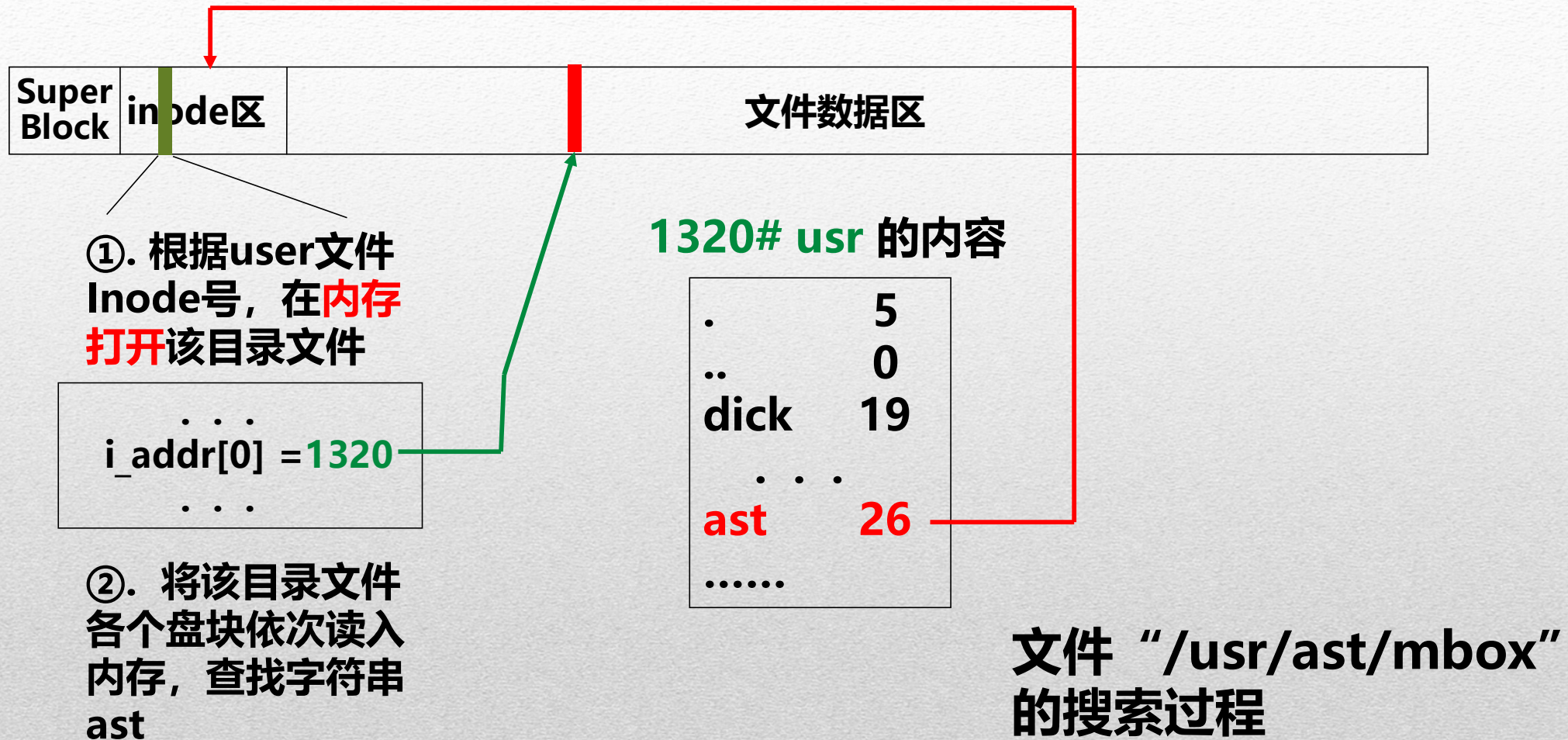


UNIX文件目录检索过程



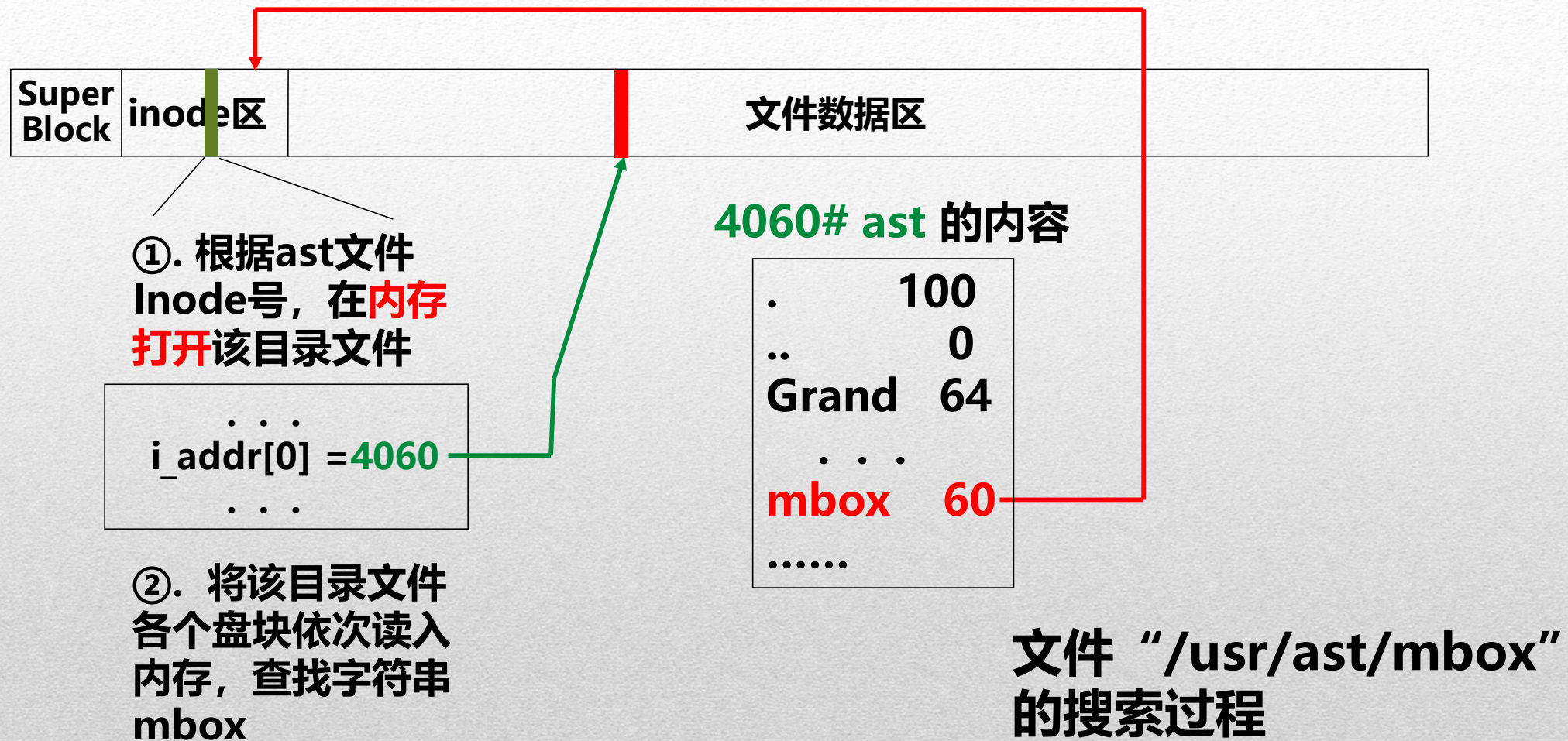


UNIX文件目录检索过程





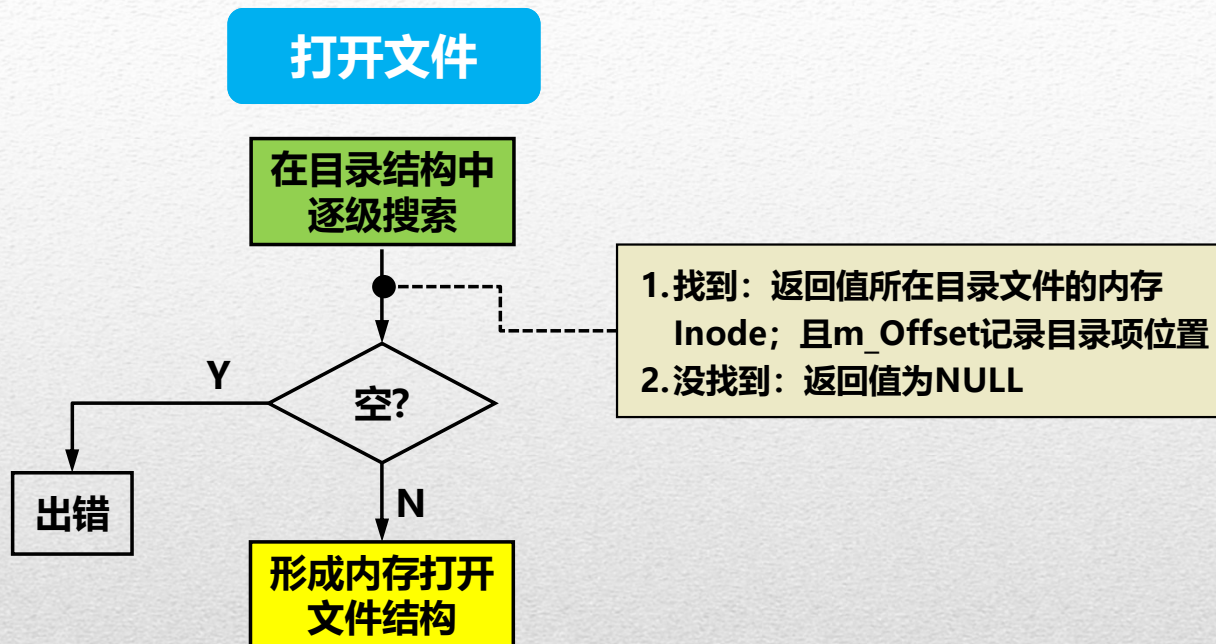
UNIX文件目录检索过程





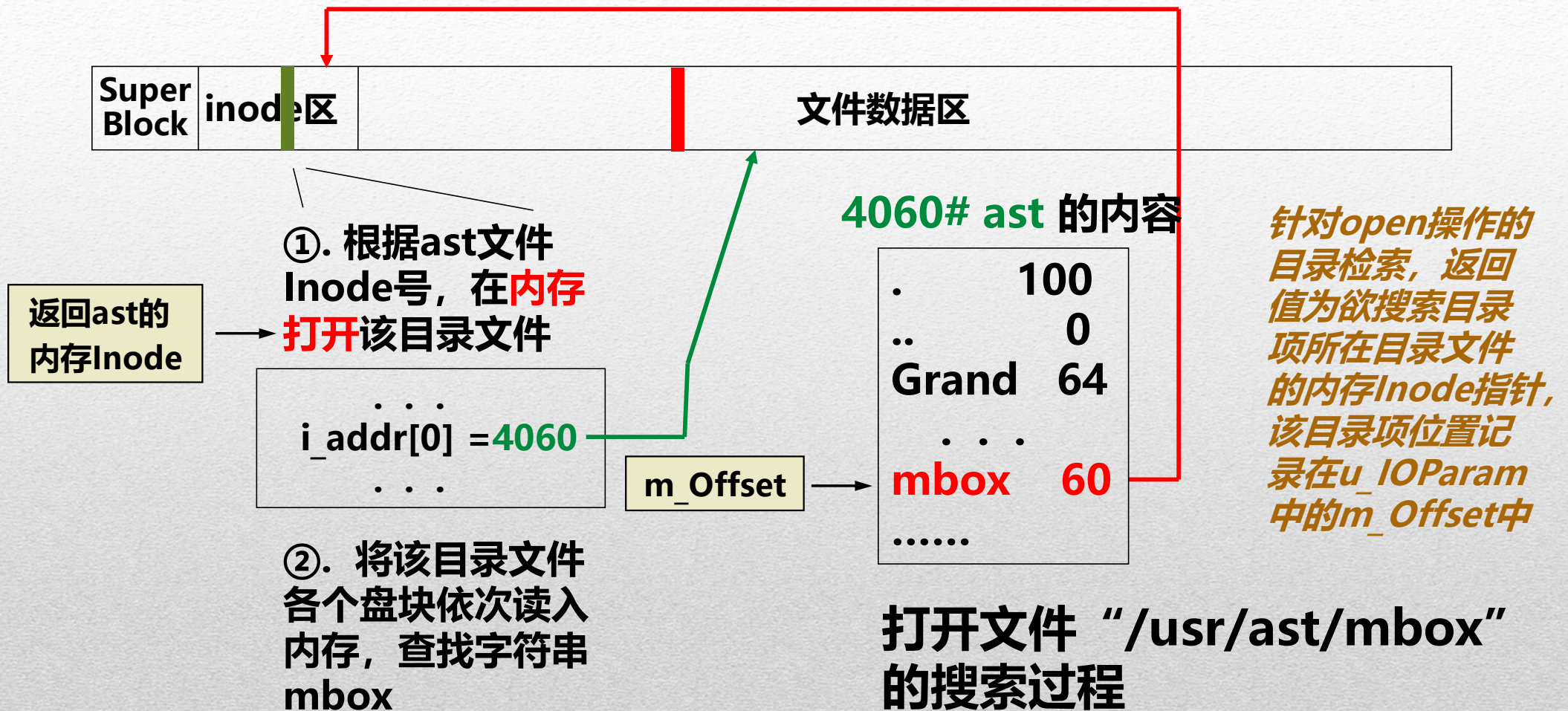
目录搜索过程：FileManager::NameI(char (*func)(), enum DirectorySearchMode mode)

```
enum DirectorySearchMode
{
    OPEN = 0, /* 以打开文件方式搜索目录 */
    CREATE = 1, /* 以新建文件方式搜索目录 */
    DELETE = 2 /* 以删除文件方式搜索目录 */
};
```



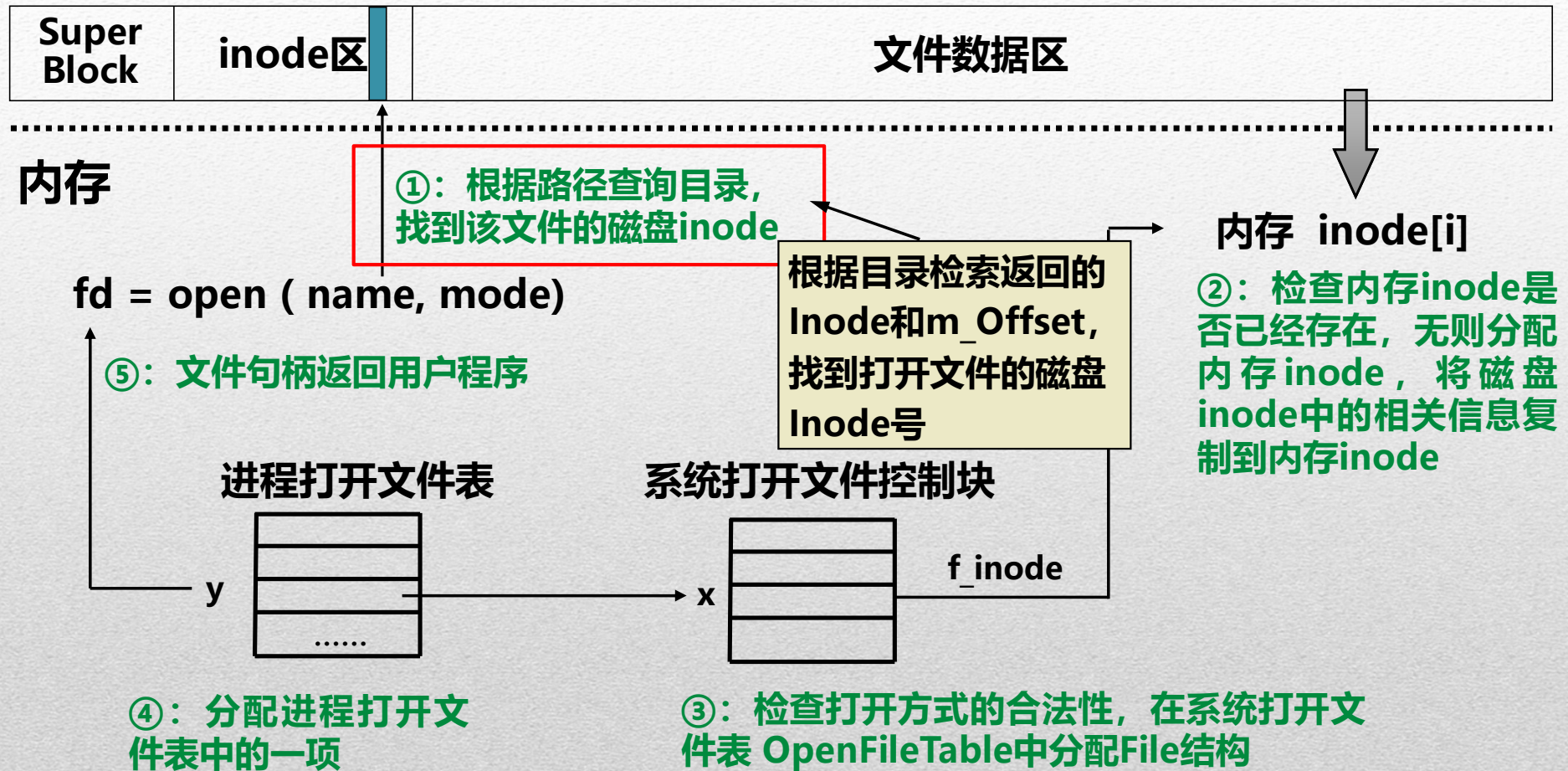


UNIX文件目录检索过程

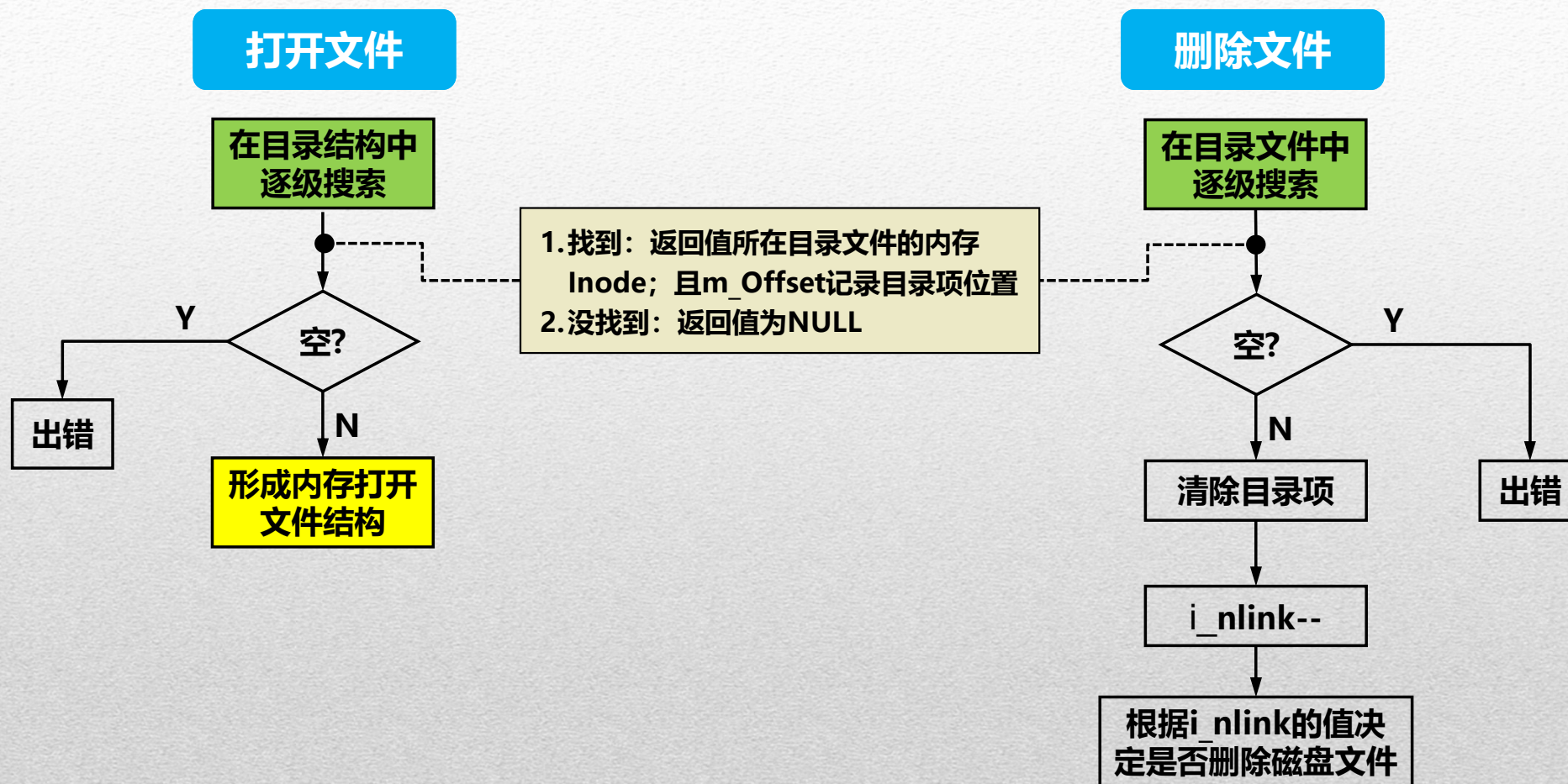




再看文件的打开过程

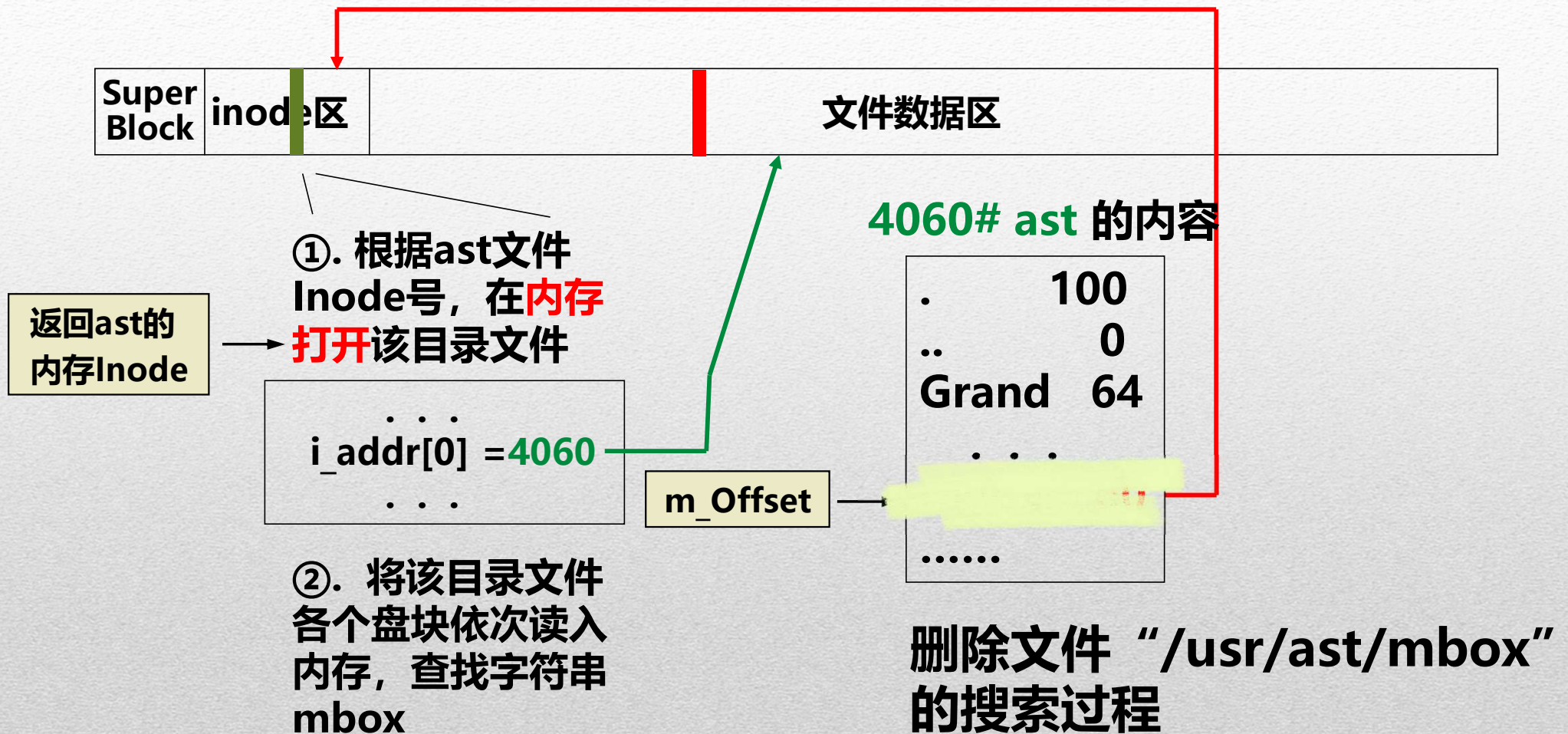


目录搜索过程: FileManager::NameI(char (*func)(), enum DirectorySearchMode mode)

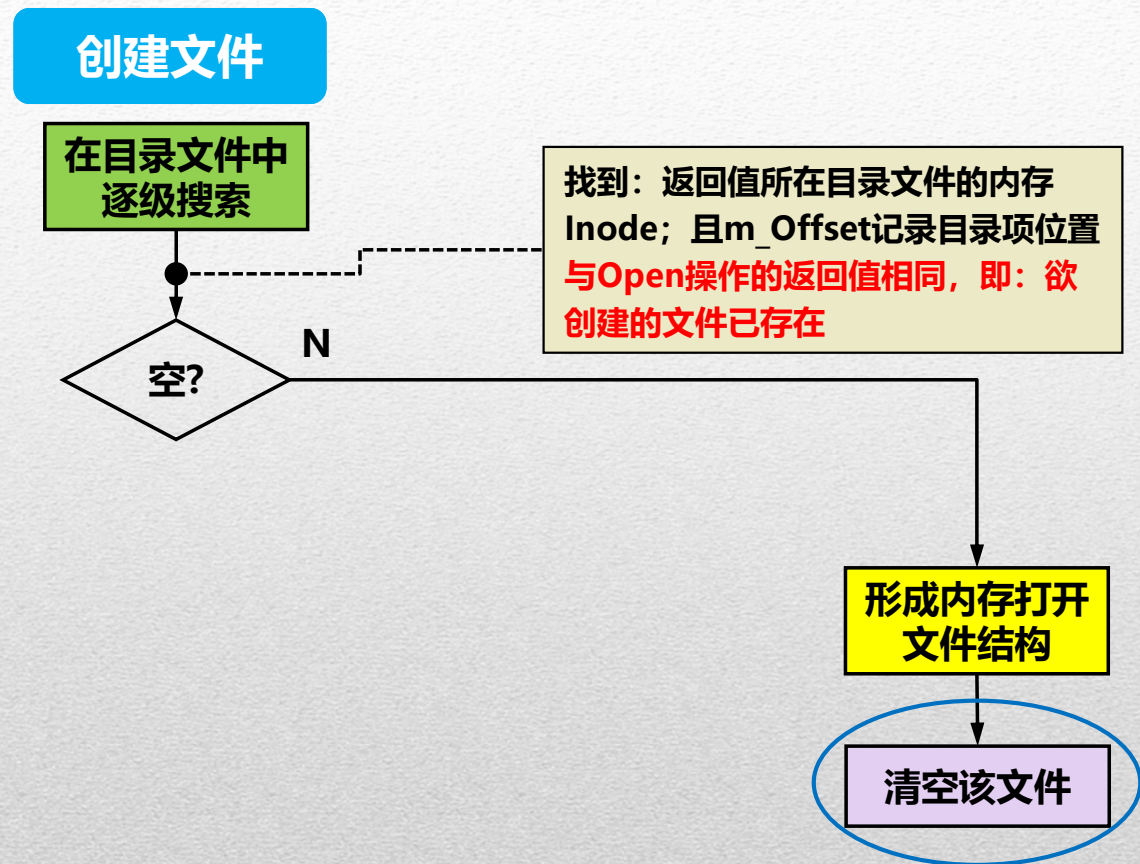




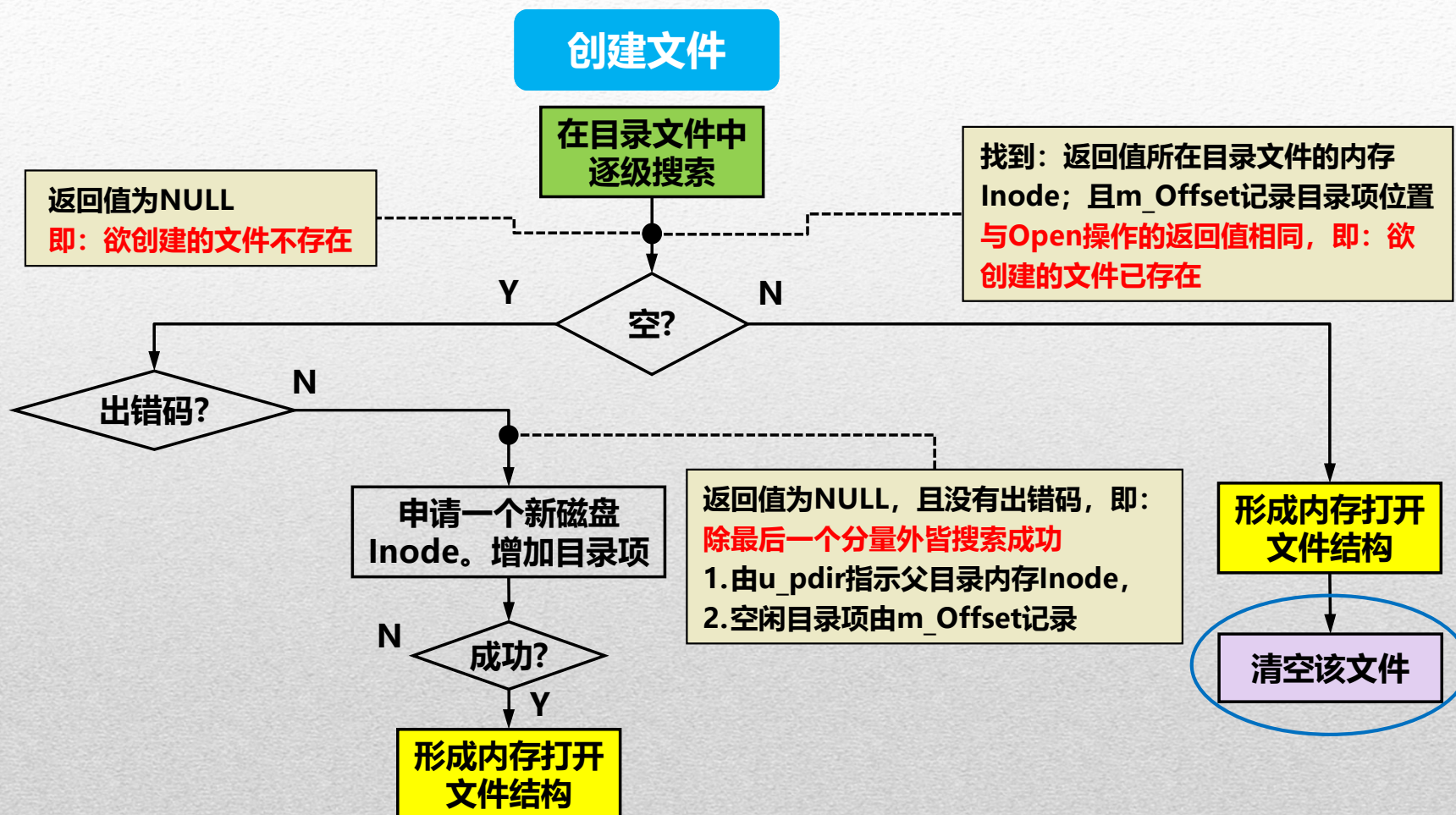
UNIX文件目录检索过程



目录搜索过程：FileManager::Name(char (*func)(), enum DirectorySearchMode mode)

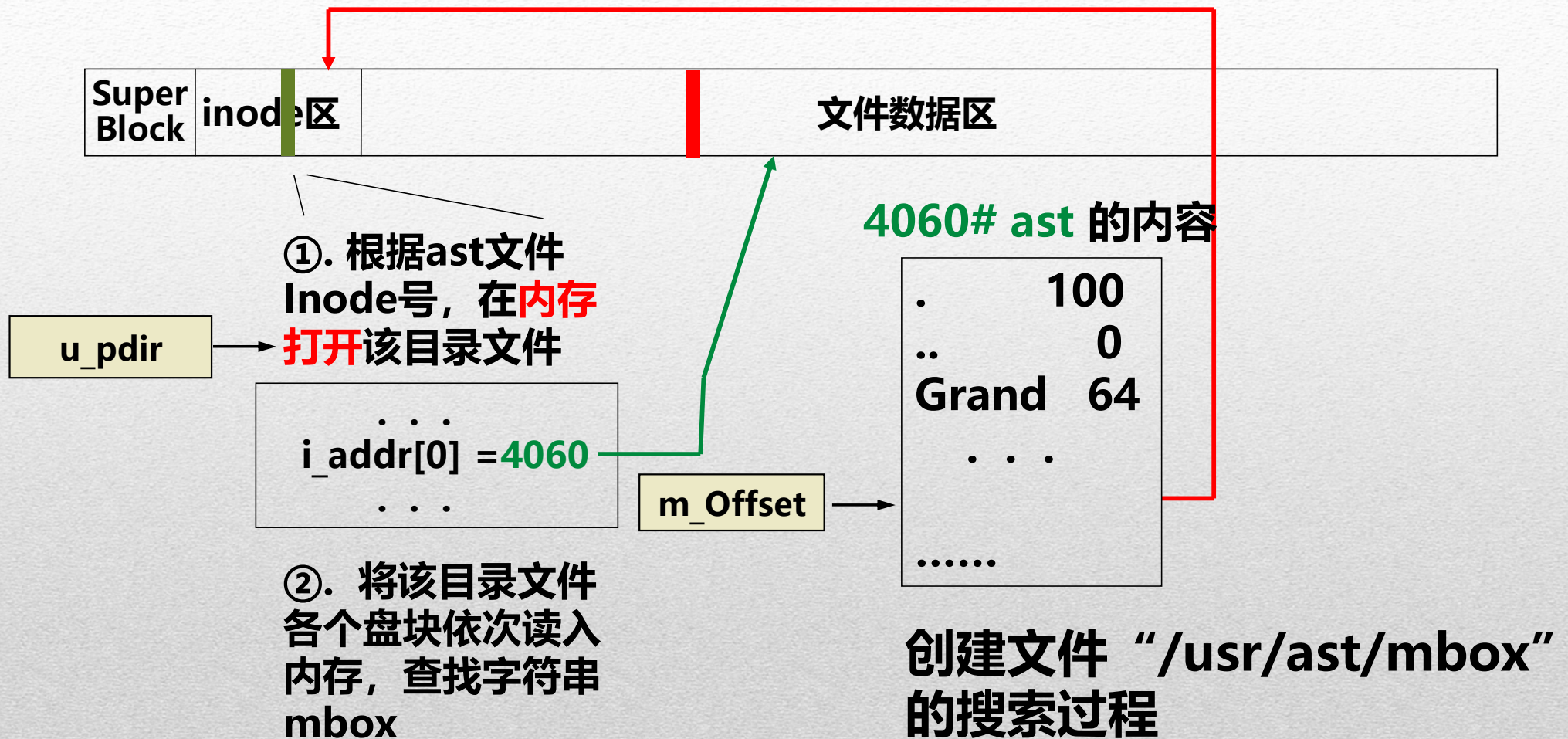


目录搜索过程: FileManager::NameI(char (*func)(), enum DirectorySearchMode mode)

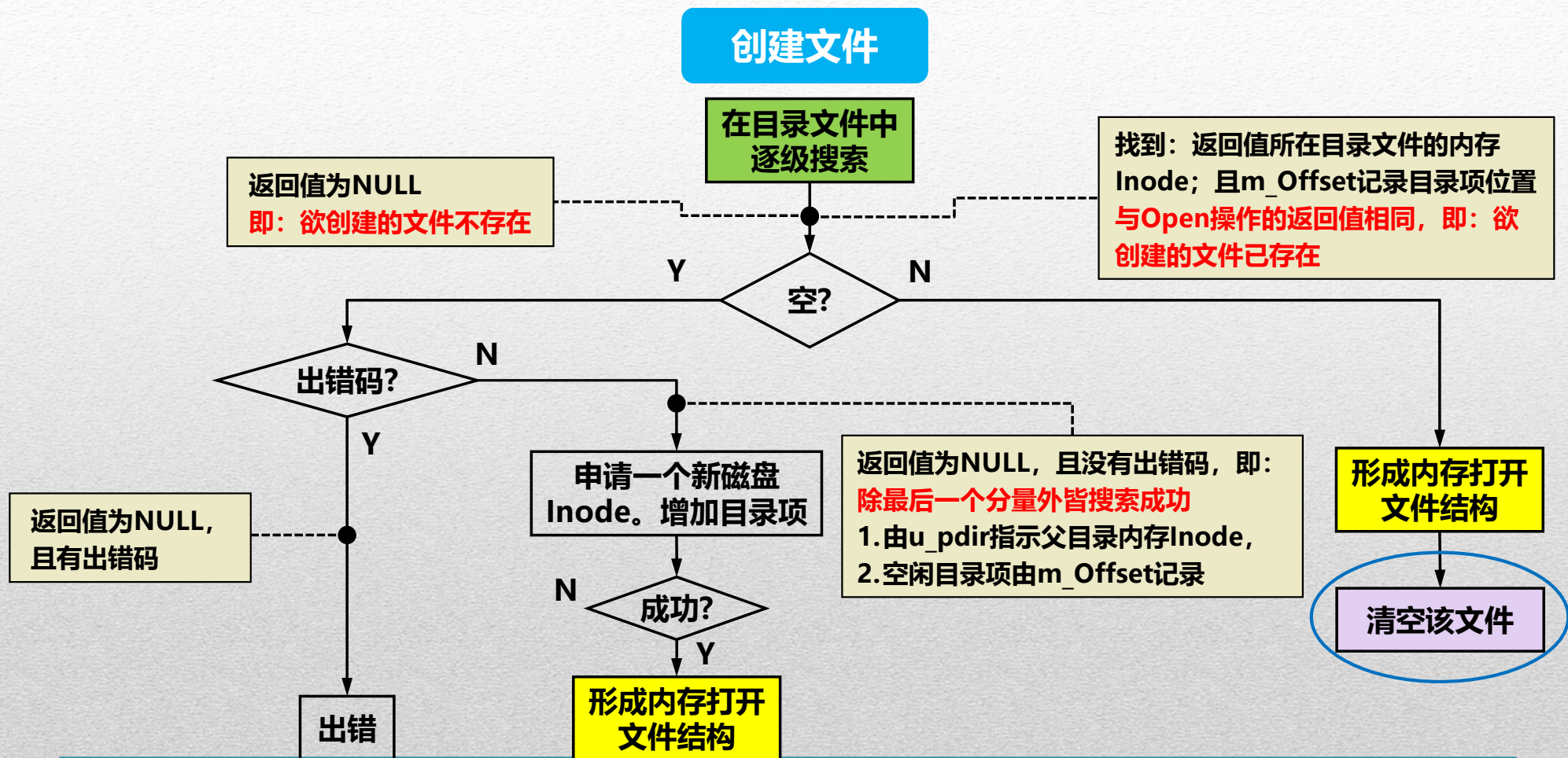




UNIX文件目录检索过程

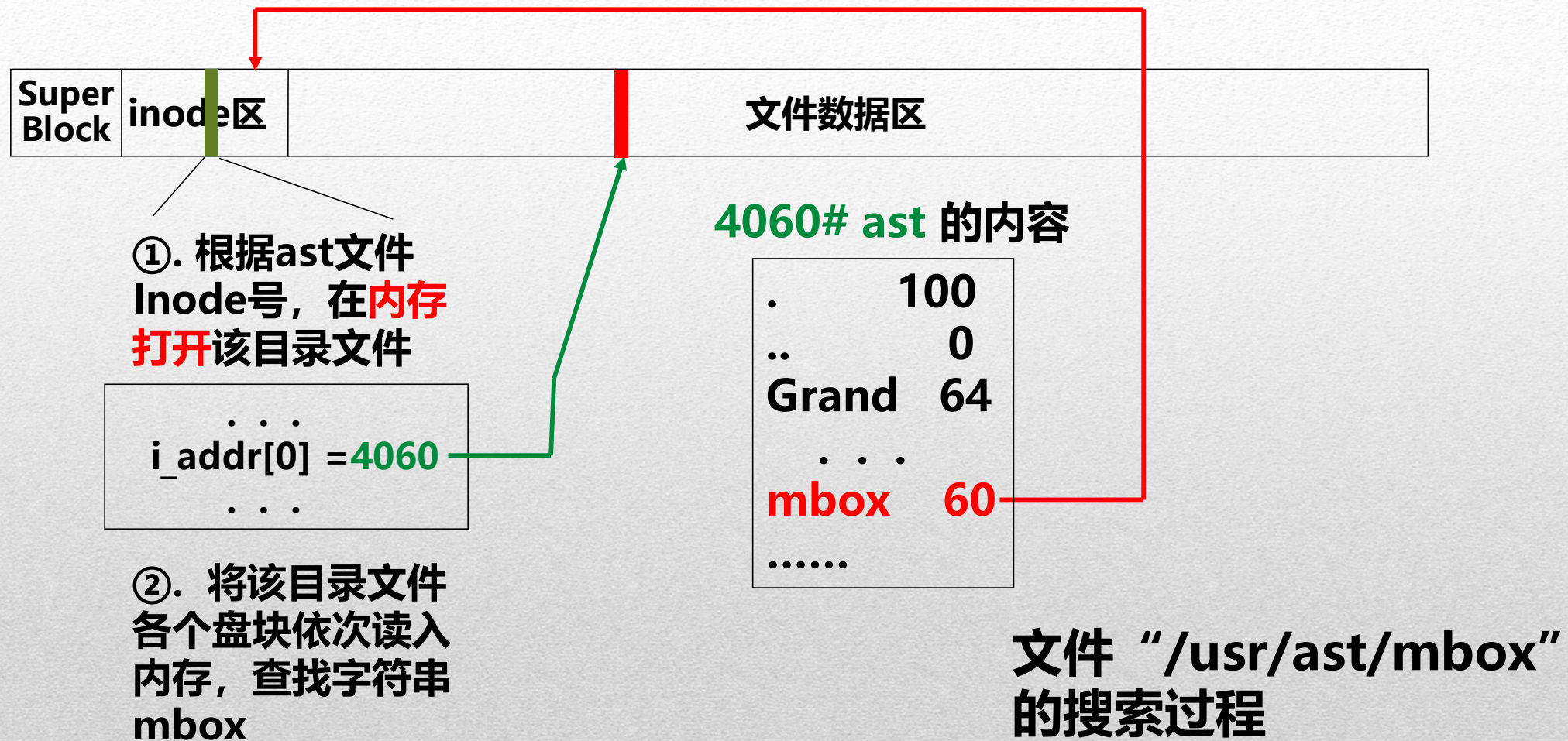


目录搜索过程: FileManager::NameI(char (*func)(), enum DirectorySearchMode mode)





UNIX文件目录检索过程





UNIX文件目录检索过程

如果设置 “/usr/ast/mbox” 为当前工作目录

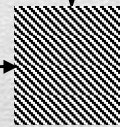
根目录的内容

.....	
bin	4
dev	7
lib	14
etc	9
usr	6
.....	

usr 的内容

.	5
..	0
dick	19
. . .	
ast	26
.....	

u_pdir



u_dent

ast 的内容

.	100
..	0
Grand	64
. . .	
mbox	60
.....	

u_cdir



UNIX文件系统

物理结构：混合索引结构（三级索引结构）

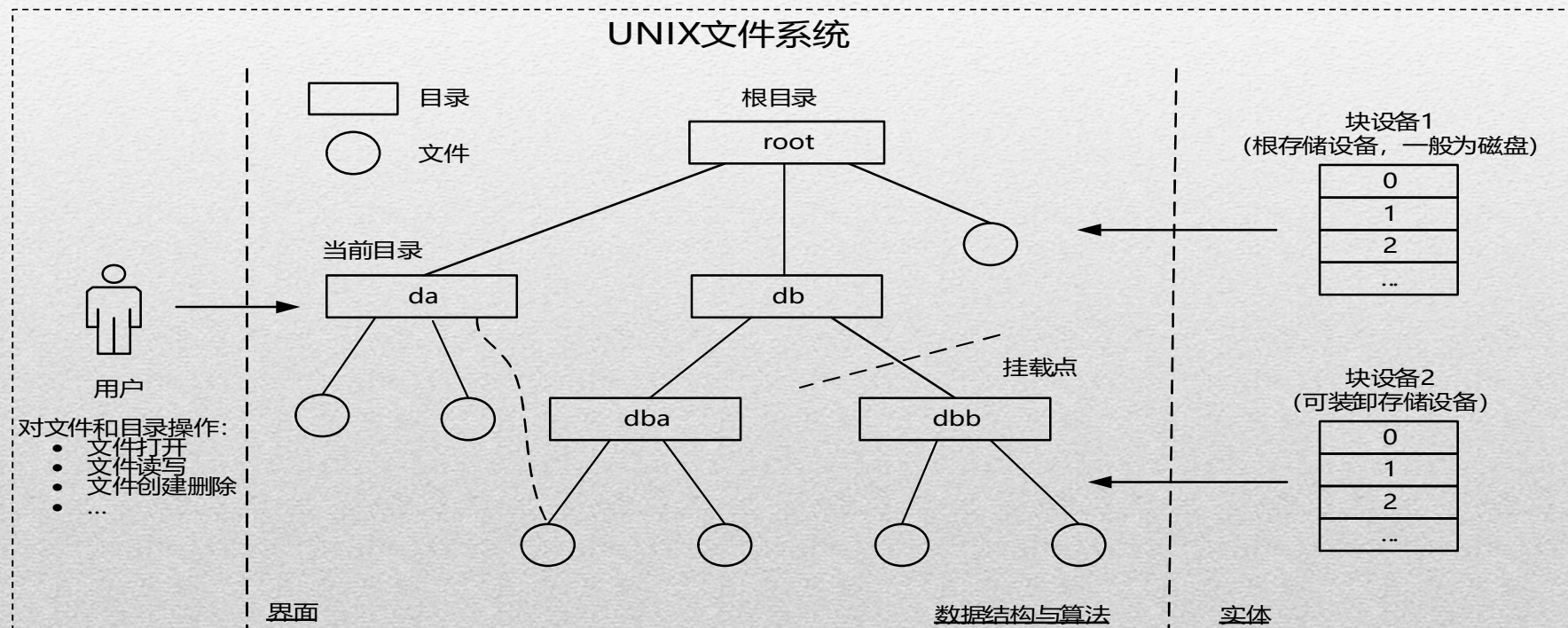
主要数据结构：

空闲盘块的管理：成组链接法

外存：i node、superblock、文件目录

目录结构：带勾连的树形结构

内存：内存文件打开结构





本节小结:

- 1 UNIX文件系统的目录管理
 - 2 UNIX文件系统的内存打开结构
-