

深圳大学 计算机与软件学院

College of Computer Science and Software Engineering of Shenzhen University



系统编程

基于TaiShan服务器/openEuler OS 的实践

第四讲：文件操作 – 内核表达

打开文件的内核表达

■ 内核用3个相关的数据结构来表达打开的文件

● 描述符表

- 每个进程
- 通过进程打开文件描述符索引
- 每项指向文件表中的一项

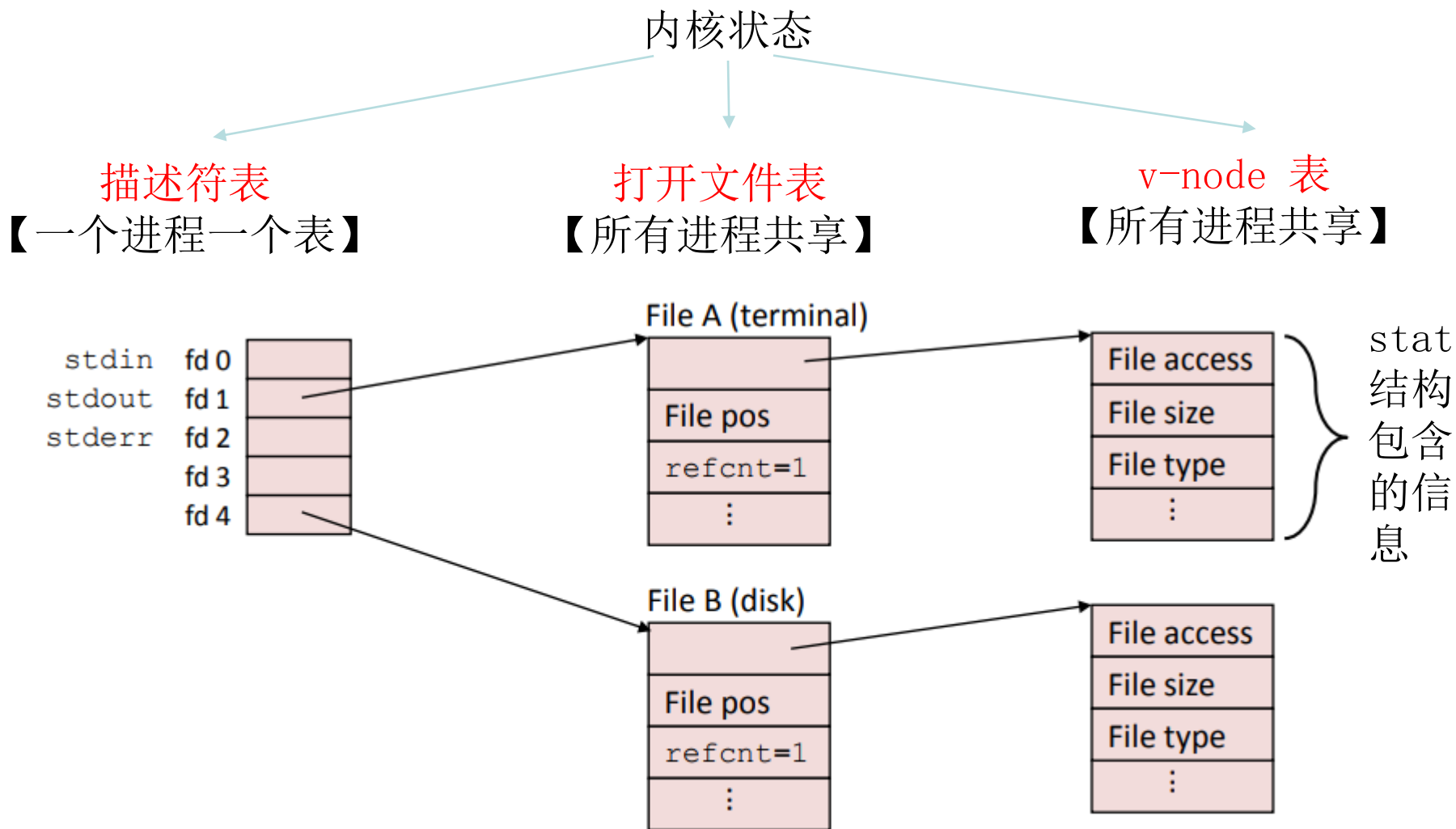
● 文件表

- 所有进程共享
- 每项包含以下信息：文件位置、引用数、…、指向v-node表的一项的指针等

● v-node表

- 所有进程共享
- 包含可被stat系统调用读取的信息

内核跟踪用户进程打开的文件



内核跟踪用户进程打开的文件

■ 用同一文件名两次调用open

描述符表

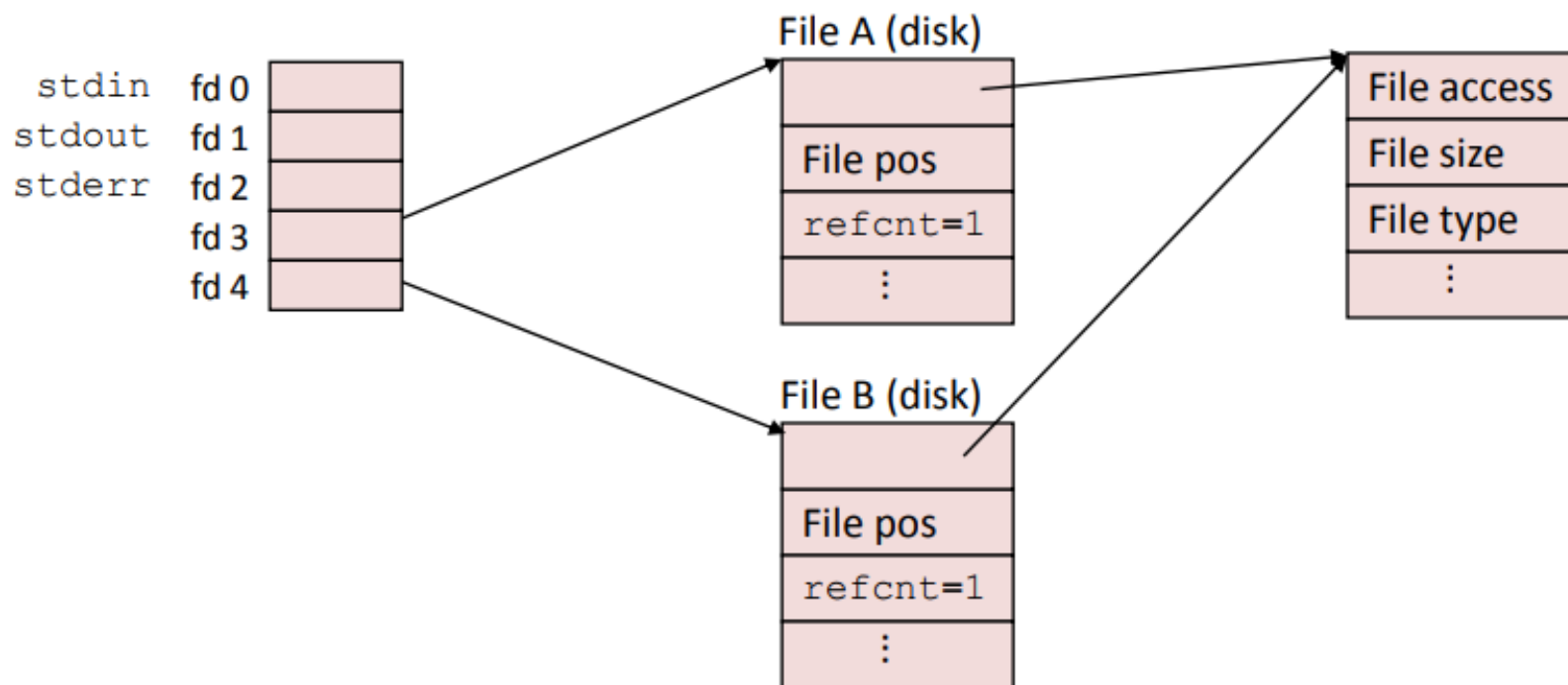
【一个进程一个表】

打开文件表

【所有进程共享】

v-node 表

【所有进程共享】



子进程继承其父进程打开的文件

■ 调用fork()之前

描述符表

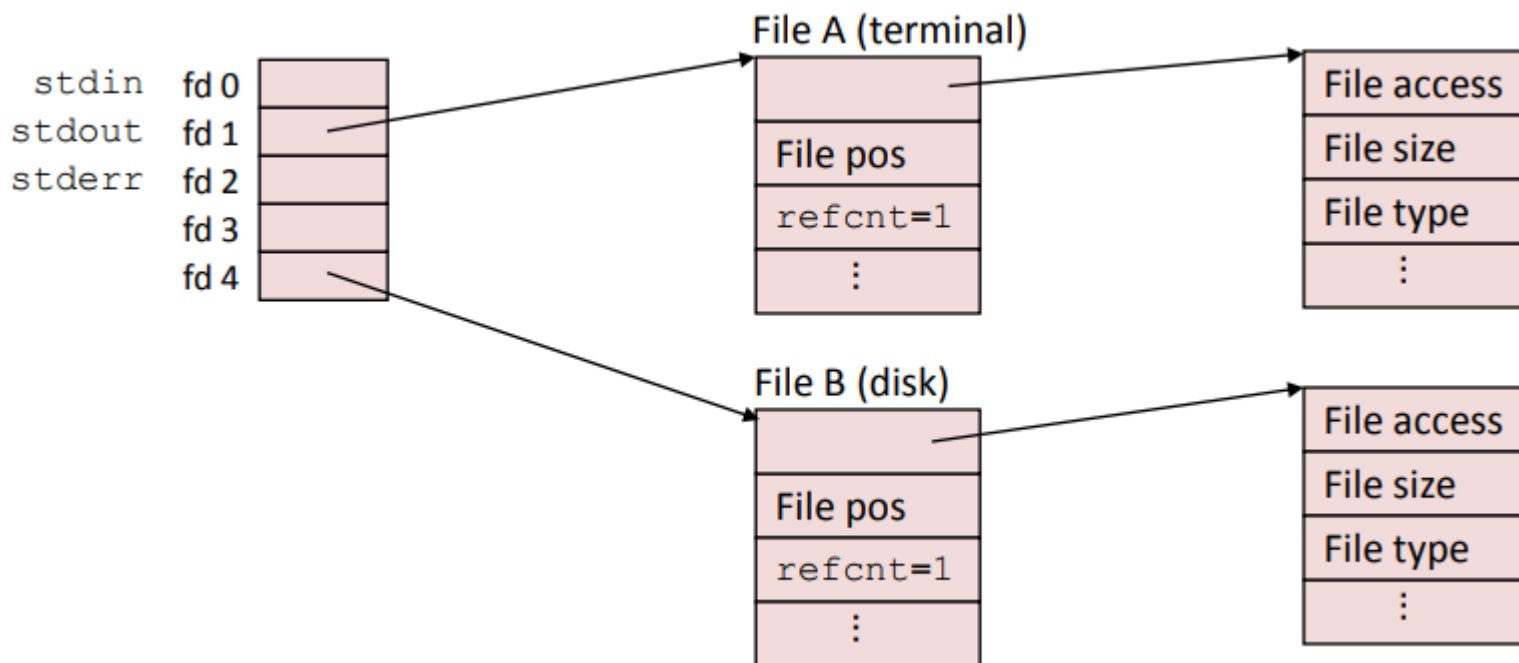
【一个进程一个表】

打开文件表

【所有进程共享】

v-node 表

【所有进程共享】



子进程继承其父进程打开的文件

■ 调用fork()之后

- 子进程的描述符表与父进程的一样
- 打开文件表的引用数+1

描述符表

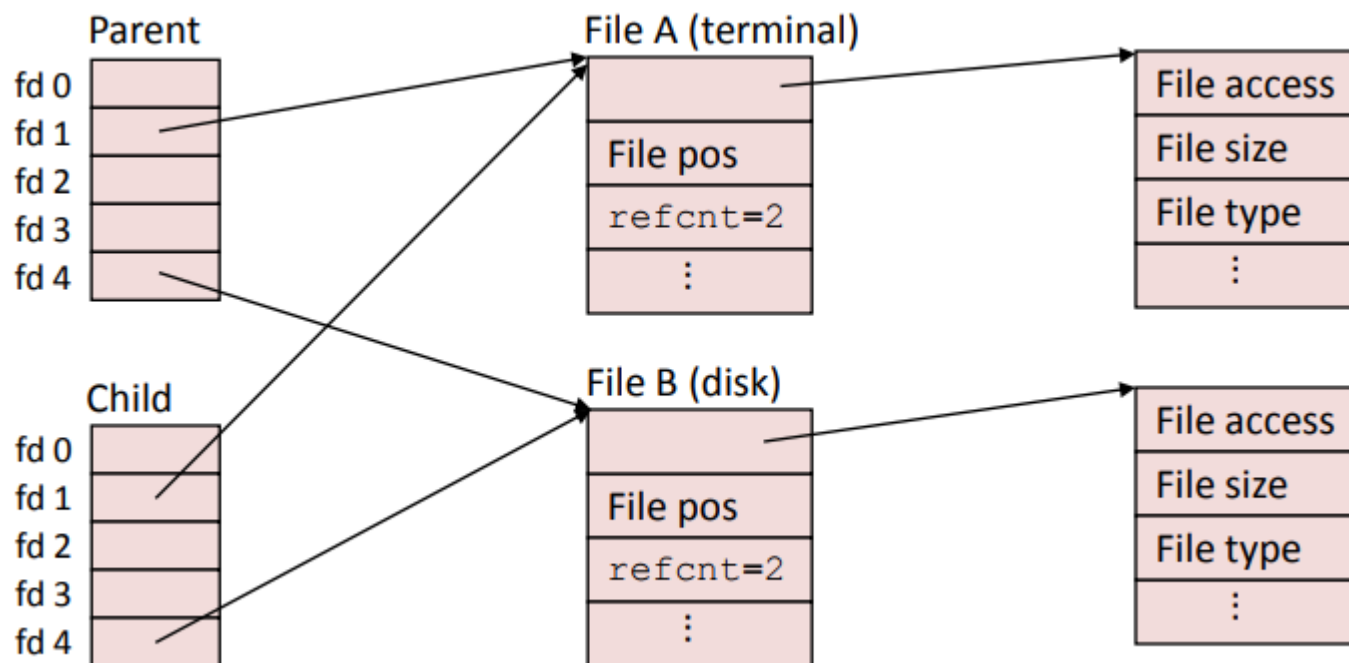
【一个进程一个表】

打开文件表

【所有进程共享】

v-node 表

【所有进程共享】



文件描述符 - fork

■ 如果文件仅包含串” abcde”, 程序将打印什么?

```
#include <stdio.h>
#include <fcntl.h>
int main(int argc, char *argv[])
{
    int fd1;
    char c1, c2;
    char *fname = argv[1];
    fd1 = open(fname, O_RDONLY, 0);
    read(fd1, &c1, 1);
    if (fork()) { /* Parent */
        read(fd1, &c2, 1);
        printf("Parent: c1 = %c, c2 = %c\n", c1, c2);
    } else { /* Child */
        sleep(5);
        read(fd1, &c2, 1);
        printf("Child: c1 = %c, c2 = %c\n", c1, c2);
    }
    return 0;
}
```

文件描述符 - dup2

- 如果文件仅包含串“**abcde**”,程序将打印什么?

```
#include <stdio.h>
#include <fcntl.h>
int main(int argc, char *argv[])
{
    int fd1, fd2, fd3;
    char c1, c2, c3;
    char *fname = argv[1];
    fd1 = open(fname, O_RDONLY, 0);
    fd2 = open(fname, O_RDONLY, 0);
    fd3 = open(fname, O_RDONLY, 0);
    dup2(fd2, fd3);
    read(fd1, &c1, 1);
    read(fd2, &c2, 1);
    read(fd3, &c3, 1);
    printf("c1 = %c, c2 = %c, c3 = %c\n", c1, c2, c3);
    return 0;
}
ffiles1.c
```


I/O重定向

■ Shell是如何重定向I/O的？

`$ls > foo.txt`

■ 系统调用dup2(olddfd, newfd)

■ 拷贝描述符表的olddfd项到newfd项

描述符表
调用dup2(4,1)之前

fd 0	
fd 1	a
fd 2	
fd 3	
fd 4	b



描述符表
调用dup2(4,1)之后

fd 0	
fd 1	b
fd 2	
fd 3	
fd 4	b

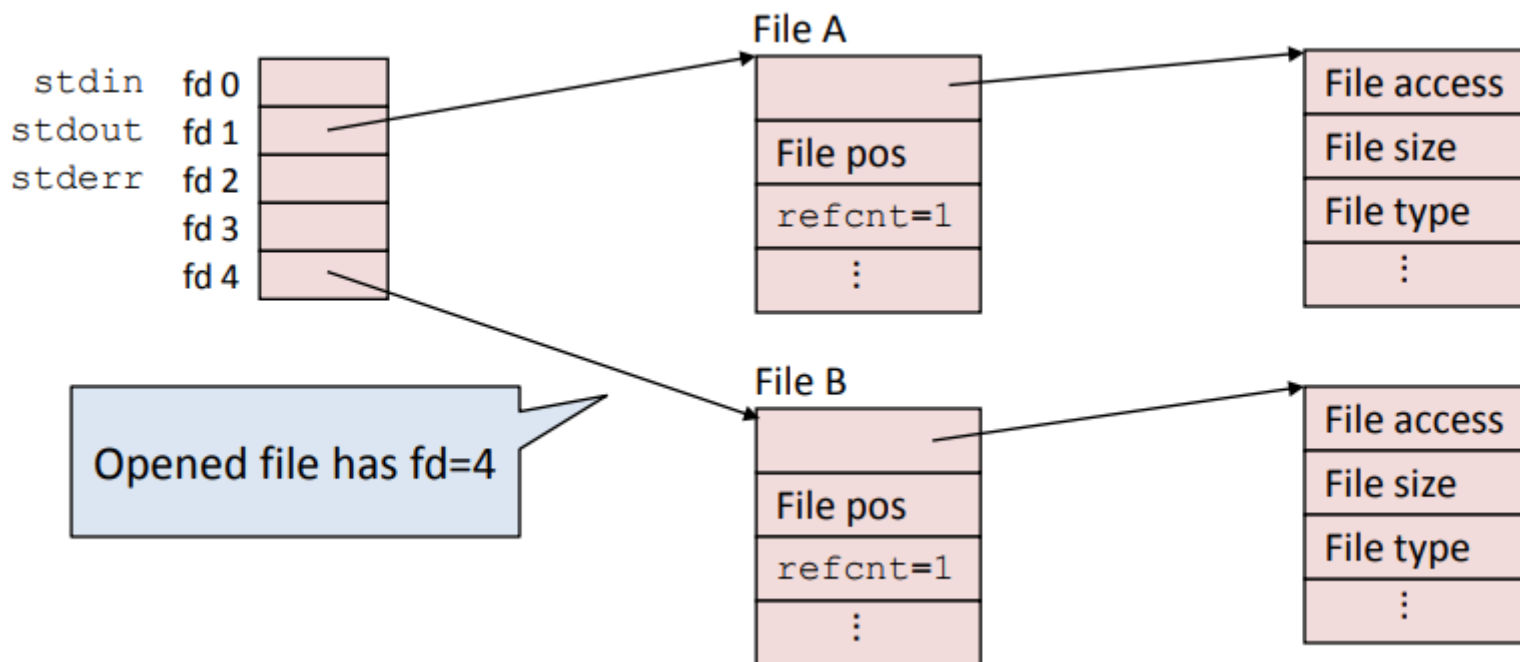
I/O重定向 - 例子

■ 第一步：打开标准输出要重定向的目标输出文件

描述符表
【一个进程一个表】

打开文件表
【所有进程共享】

v-node 表
【所有进程共享】



I/O重定向 - 例子

■ 第二步：调用dup2(4, 1)

■ 导致fd =1（标准输出）指向磁盘文件fd=4

描述符表

【一个进程一个表】

stdin	fd 0	
stdout	fd 1	
stderr	fd 2	
	fd 3	
	fd 4	

打开文件表

【所有进程共享】

File A

File pos
refcnt=0
⋮

File B

File pos
refcnt=2
⋮

v-node 表

【所有进程共享】

File access
File size
File type
⋮

File access
File size
File type
⋮