

Linux操作系统编程

Linux文件共享

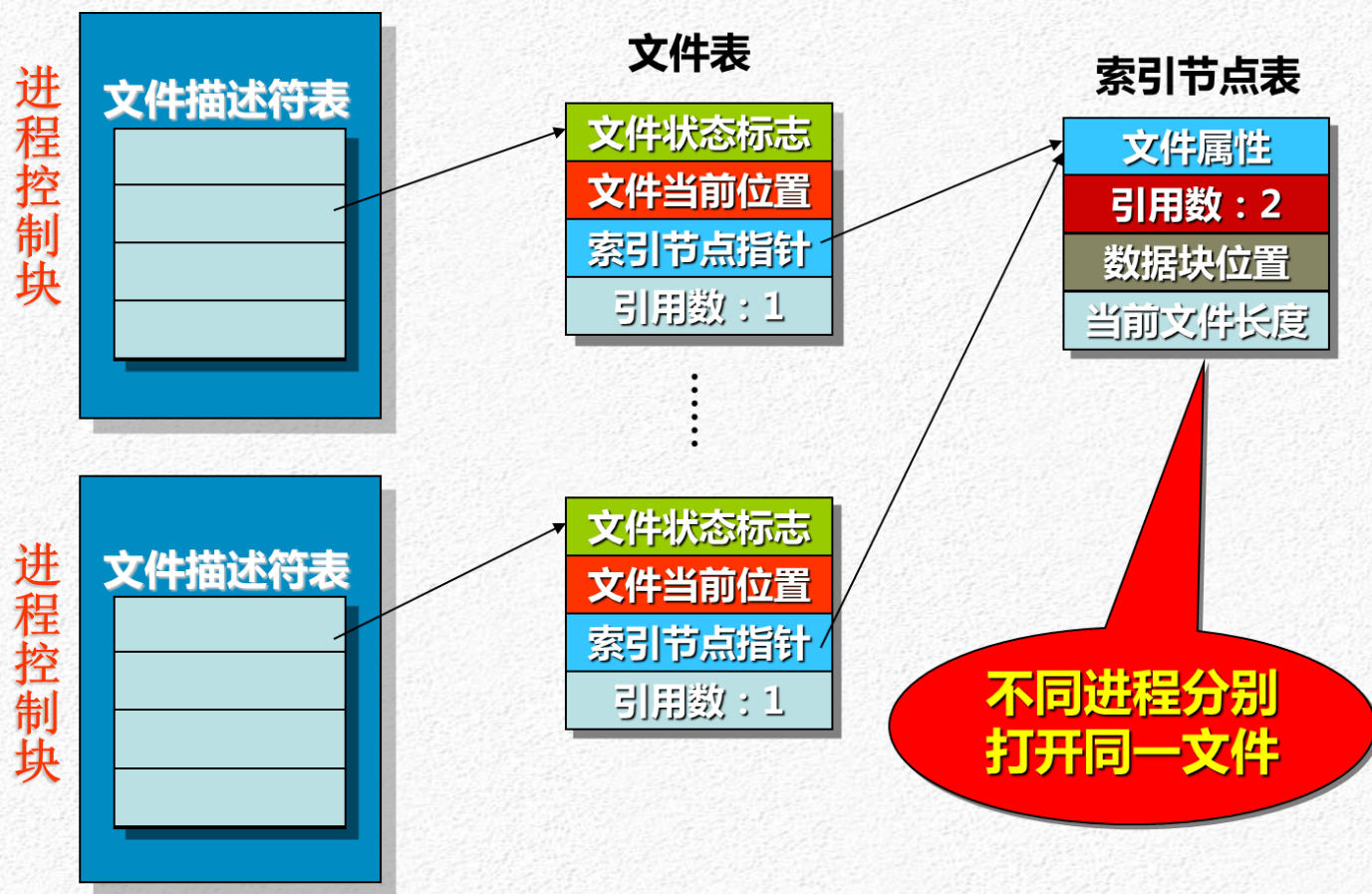
- **UNIX/Linux支持不同进程间共享文件。内核使用的三种表（文件描述符表、文件表、索引结点表）之间的关系决定了在文件共享方面一个进程对另一个进程可能产生的影响**
  - 每个进程在进程表中有一个文件描述符表，每个描述符表项指向一个文件表
  - 内核为每一个被打开的文件维护一张文件表，文件表项包含
    - **文件的状态标志（读、写、同步、非阻塞）**
    - **文件当前位置**
    - **指向该文件索引节点表的指针**
  - 每个文件（或设备）都有一个索引节点，它包含了文件类型属性及文件数据



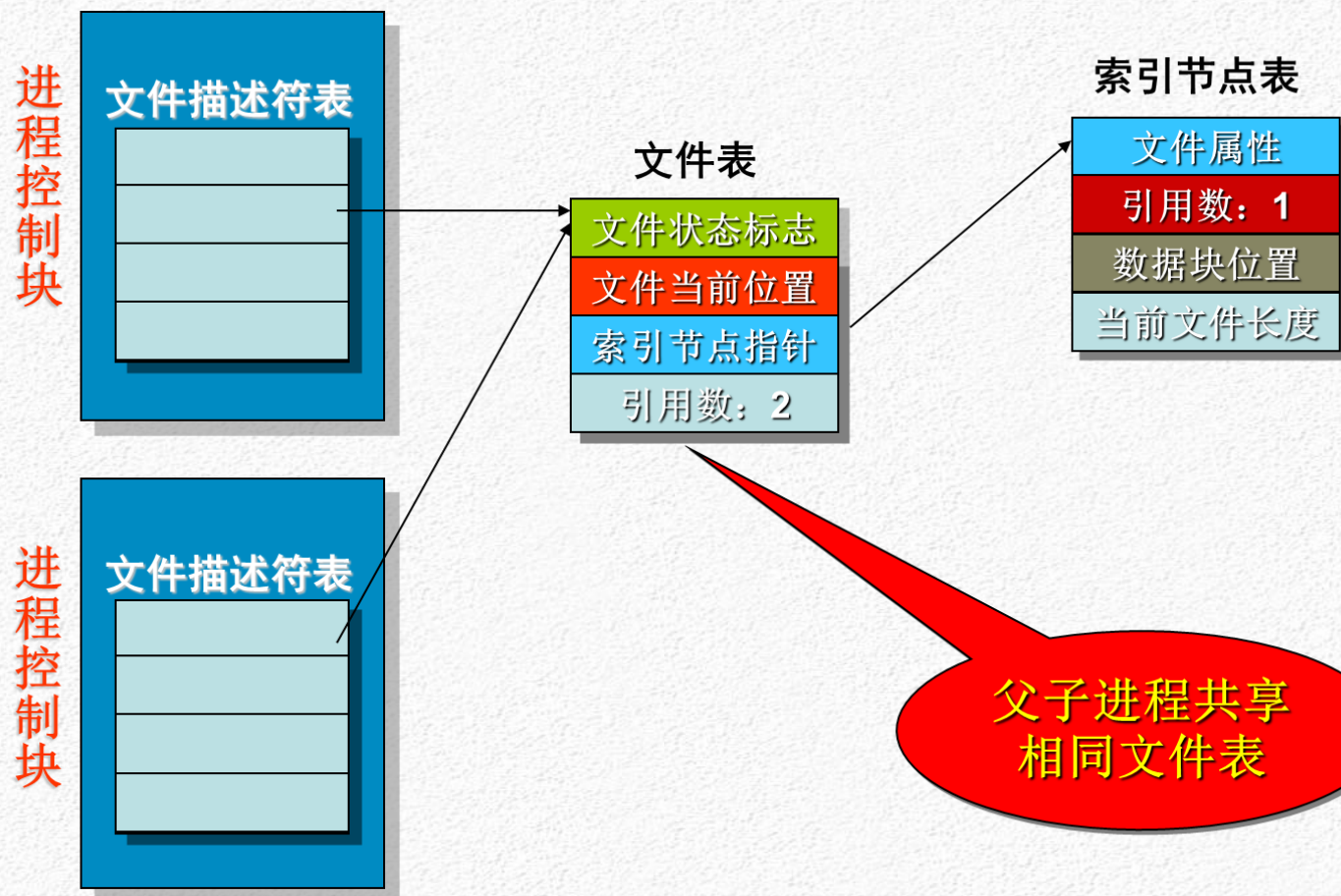
- 如果两个进程**分别打开同一个**的文件（物理文件），则它们有不同的文件表，因此每个进程有自己的文件当前位置，因此其读写操作互不影响。
- 也存在**不同进程共享**同一个文件表（父子进程），或**同一进程共享**同一个文件表（dup操作）。此时，两个进程对该文件的读写操作将基于同一个文件当前位置。



# 不同进程共享相同文件（一）

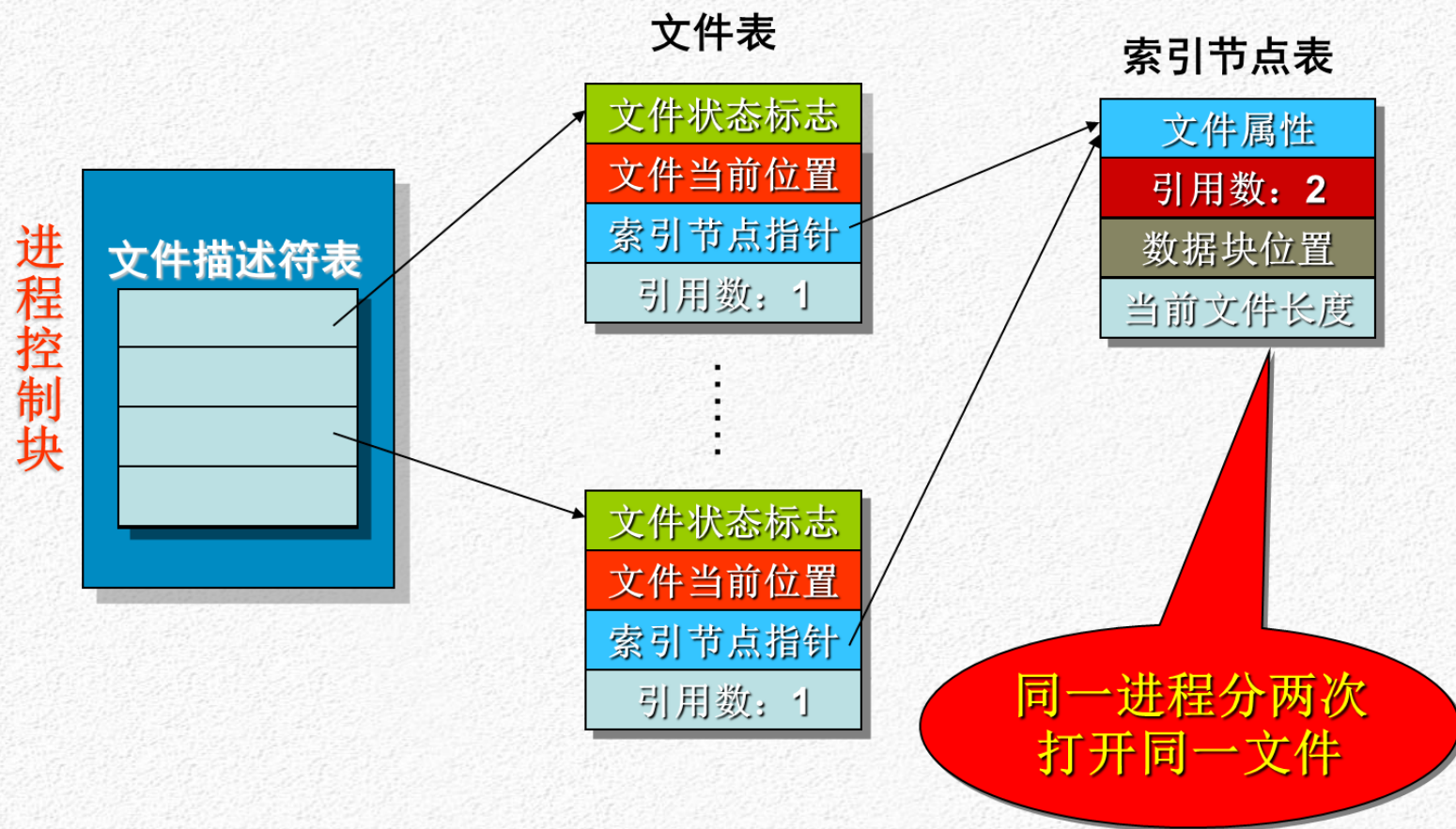


# 不同进程共享相同文件（二）





# 同一进程共享相同文件（一）





- **头文件：**unistd.h
- **函数原型**  
`int dup( int oldfd );`  
`int dup2( int oldfd, int targetfd );`
- **函数说明：**复制一个文件的描述符，dup2函数跟dup函数相似，但dup2函数允许调用者规定一个有效描述符和目标描述符的
- **返回值：**dup2函数成功返回时，目标描述符（dup2函数的第二个参数）将变成源描述符（dup2函数的第一个参数）的复制品
- **示例：**

```
int fd1, fd2;      int oldfd;  
...               oldfd = open("app_log", (O_RDWR | O_CREATE), 0644 );  
fd2 = dup( fd1 );  dup2( oldfd, 1 );  
                  close( oldfd );
```



# 同一进程共享相同文件（二）

