UC Day05

预习课

非会员水印

预习内容

文件的读写位置

非会员水印

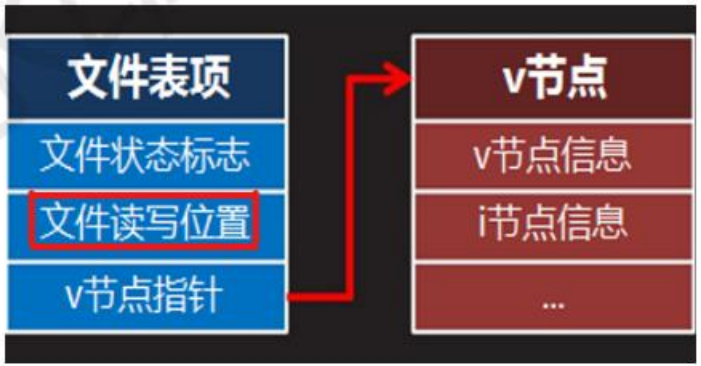
文件的读写位置

非会员水印

文件系统的物理结构

知识讲解

- 每个打开的文件都有一个与其相关的文件读写位置保存在文件表项中，用以记录从文件头开始计算的字节偏移
- 文件读写位置通常是一个非负的整数，用off_t类型表示，在32位系统上被定义为long int，而在64位系统上则被定义为long long int



文件系统的物理结构

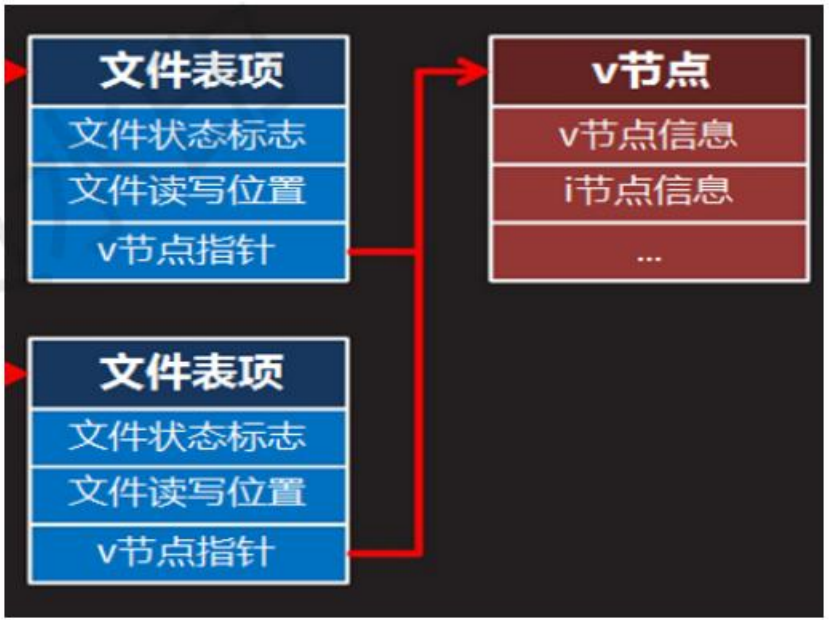
知识讲解

- 打开一个文件时，除非指定了O_APPEND标志，否则文件读写位置一律被设为0，即文件首字节的位置
- 每一次读写操作都从当前的文件读写位置开始，并根据所读写的字节数，同步增加文件读写位置，为下一次读写做好准备
- 因为文件读写位置是保存在文件表项而不是v节点中的，因此通过多次打开同一个文件得到多个文件描述符，各自拥有各自的文件读写位置

文件系统的物理结构

知识讲解

- 因为文件读写位置是保存在文件表项而不是v节点中的，因此通过多次打开同一个文件得到多个文件描述符，各自拥有各自的文件读写位置



直播课见

非会员水印

Tedu.cn
达内教育

UC

C/C++教学体系

非会员水印

目录

顺序读写与随机读写

文件描述符的复制

访问测试

修改文件大小

Tedu.cn
达内教育

顺序读写与随机读写

非会员水印

文件的打开与关闭

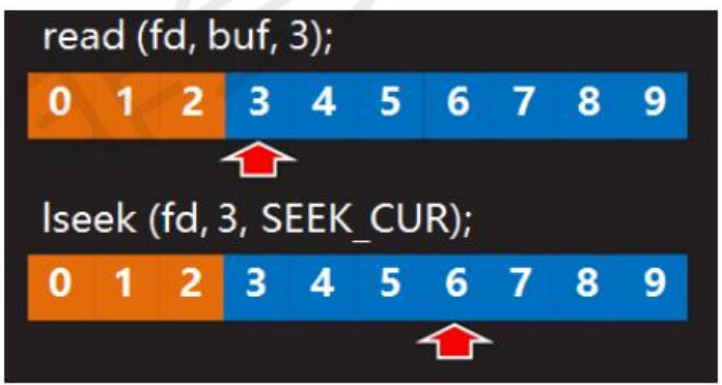
知识讲解

- #include <unistd.h>
- off_t lseek(int fd, off_t offset, int whence);
  - 功能：人为调整文件读写位置
  - 参数：fd：　　文件描述符。
    offset：　文件读写位置偏移字节数
    whence：　offset参数的偏移起点，可以如下取值：
    SEEK_SET　- 从文件头(首字节)开始
    SEEK_CUR - 从当前位置(最后被读写字节的下一个字节)开始
    SEEK_END - 从文件尾(最后一个字节的下一个字节)开始
  - 返回值：成功返回调整后的文件读写位置，失败返回-1

文件的打开与关闭

知识讲解

- lseek函数的功能仅仅是修改保存在文件表项中的文件读写位置，并不实际引发任何I/O动作
  - lseek (fd, -7, SEEK_CUR); // 从当前位置向文件头偏移7字节
  - lseek (fd, 0, SEEK_CUR); // 返回当前文件读写位置
  - lseek (fd, 0, SEEK_END); // 返回文件总字节数



Tedu.cn
达内教育

文件描述符的复制

非会员水印

文件描述符的复制

知识讲解

- #include <unistd.h>
- int dup(int oldfd);
  - 功能：复制文件描述符表的特定条目到最小可用项：
  - 参数：oldfd：源文件描述符
- 返回值：成功返回目标文件描述符，失败返回-1

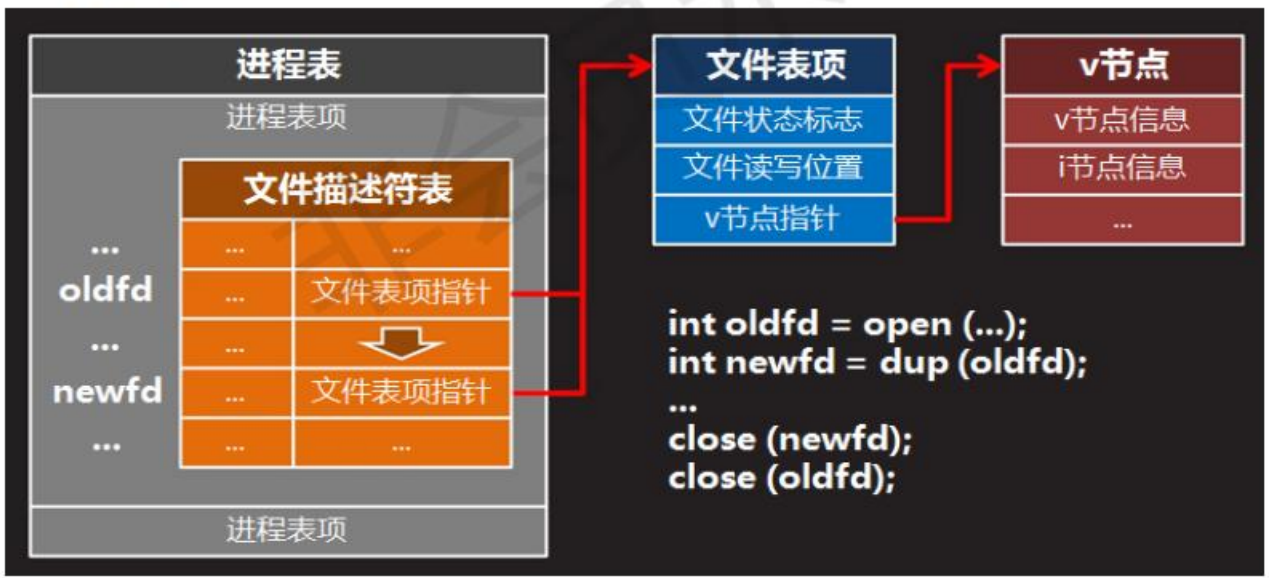
- dup函数将oldfd参数所对应的文件描述符表项复制到文件描述符表第一个空闲项中，同时返回该表项对应的文件描述符。dup函数返回的文件描述符一定是调用进程当前未使用的最小文件描述符

非会员水印

文件描述符的复制

- dup函数只复制文件描述符表项，不复制文件表项和v节点，因此该函数所返回的文件描述符可以看做是参数文件描述符oldfd的副本，它们标识同一个文件表项

知识讲解



文件描述符的复制

知识讲解

- 注意，当关闭文件时，即使是由dup函数产生的文件描述符副本，也应该通过close函数关闭，因为只有当关联于一个文件表项的所有文件描述符都被关闭了，该文件表项才会被销毁，类似地，也只有当关联于一个v节点的所有文件表项都被销毁了，v节点才会被从内存中删除，因此从资源合理利用的角度讲，凡是明确不再继续使用的文件描述符，都应该尽可能及时地用close函数关闭
- dup函数究竟会把oldfd参数所对应的文件描述符表项，复制到文件描述符表的什么位置，程序员是无法控制的，这完全由调用该函数时文件描述符表的使用情况决定，因此对该函数的返回值做任何约束性假设都是不严谨的

文件描述符的复制

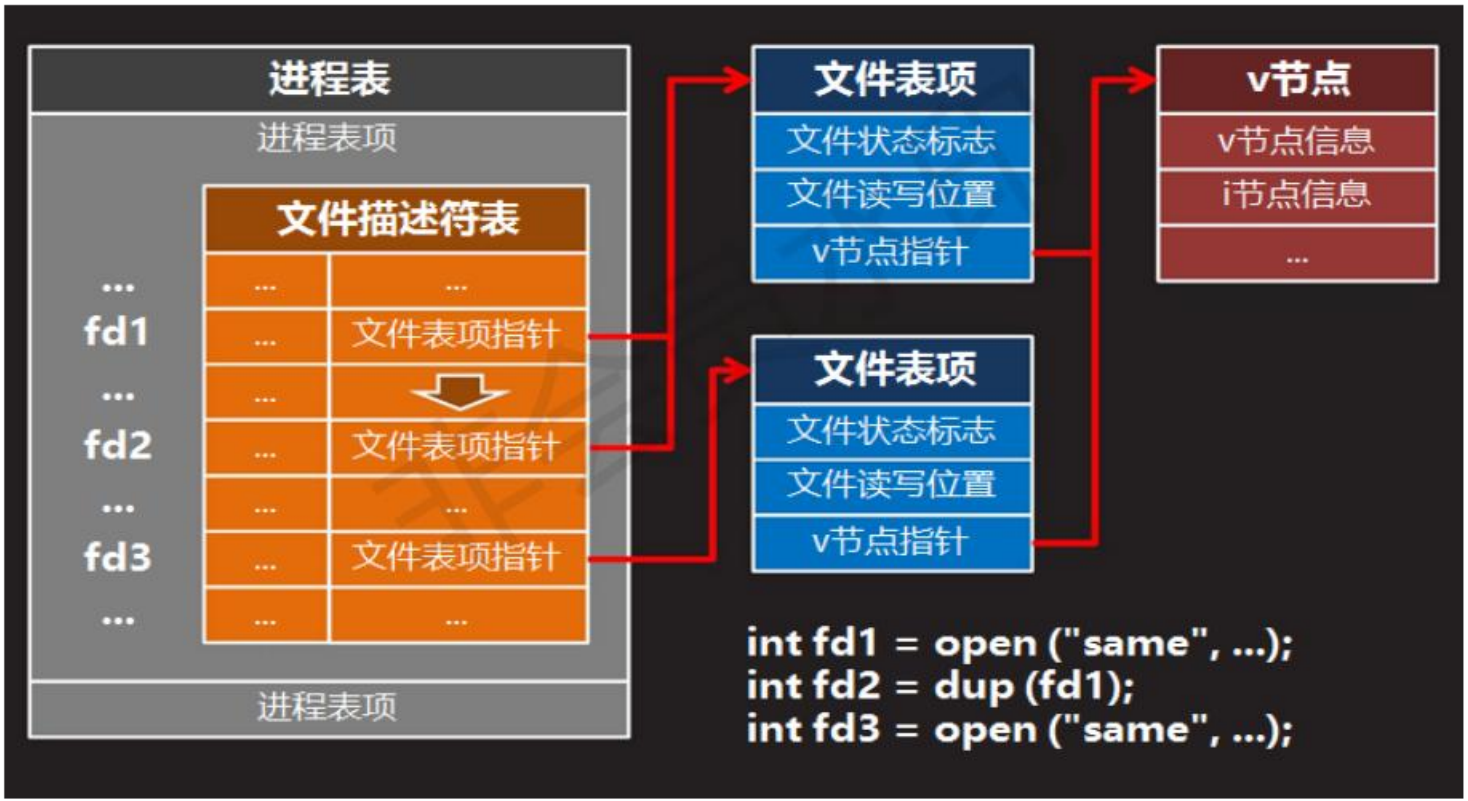
- 由dup函数返回的文件描述符与作为参数传递给该函数的文件描述符标识的是同一个文件表项，而文件读写位置是保存在文件表项而非文件描述符表项中的，因此通过这些文件描述符中的任何一个，对文件进行读写或随机访问，都会影响通过其它文件描述符操作的文件读写位置。这与多次通过open函数打开同一个文件不同

知识讲解

非会员水印

文件描述符的复制

知识讲解



文件描述符的复制

知识讲解

- #include <unistd.h>
- int dup2(int oldfd, int newfd);
  - 功能：复制文件描述符表的特定条目到指定项：
  - 参数：oldfd：源文件描述符
    newfd：目标文件描述符
  - 返回值：成功返回目标文件描述符(newfd)，失败返回-1。
- dup2函数在复制由oldfd参数所标识的源文件描述符表项时，会先检查由newfd参数所标识的目标文件描述符表项是否空闲，若空闲则直接将前者复制给后者，否则会先将目标文件描述符newfd关闭，使之成为空闲项，再行复制

北京尼尔水印

Tedu.cn
达内教育

访问测试

非会员水印

访问测试

知识讲解

- #include <unistd.h>
- int access(char const* pathname, int mode);
  - 功能：判断当前进程是否可以对某个给定的文件执行某种访问。
  - 参数：pathname　文件路径
    mode　被测试权限，可以以下取值
    R_OK - 可读否
    W_OK - 可写否
    X_OK - 可执行否
    F_OK - 存在否
  - 返回值：成功返回0，失败返回-1

Tedu.cn
达内教育

修改文件大小

非会员水印

修改文件大小

知识讲解

- #include <unistd.h>
- int truncate(char const* path, off_t length);
- int ftruncate(int fd, off_t length);
  - 功能：修改指定文件的大小
  - 参数：path　　文件路径
    length　文件大小
    fd　　文件描述符
  - 返回值：成功返回0，失败返回-1。
- 该函数既可以把文件截短，也可以把文件加长，所有的改变均发生在文件的尾部，新增加的部分用数字0填充

非会员水印

谢谢

非会员水印

UC Day05

复习课

非会员水印

系统I/O与标准I/O

非会员水印

标准I/O与系统I/O

知识讲解

- 标准库中所提供的文件操作函数，底层会调用系统调用函数
- 当系统调用函数被执行时，需要在用户态和内核态之间来回切换，因此频繁执行系统调用函数会严重影响性能
- 标准库做了必要的优化，内部维护一个缓冲区，只在满足特定条件时才将缓冲区与系统内核同步，借此降低执行系统调用的频率，减少进程在用户态和内核态之间来回切换的次数，提高运行性能

未经授权

下节课见

非会员水印