

文件操作的实现

创建文件:

建立系统与文件的联系, 实质是建立文件的FCB

- 在目录中为新文件建立一个目录项, 根据提供的参数及需要填写相关内容
- 分配必要的存储空间

打开文件:

根据文件名在文件目录中检索, 并将该文件的目录项读入内存, 建立相应的数据结构, 为后续的文件操作做好准备

文件描述符/文件句柄



文件操作—建立文件

create (文件名, 访问权限)

① 检查参数的合法性

例如: 文件名是否符合命名规则;
有无重名文件;

合法→②, 否则→报错、返回

② 申请空闲目录项, 并填写相关内容;

③ 为文件申请磁盘块; (?)

④ 返回



文件操作—打开文件

为文件读写做准备

给出文件路径名，获得文件句柄(file handle)或文件描述符(file descriptor)，需将该文件的目录项读到内存

fd=open (文件路径名, 打开方式)

- ① 根据文件路径名查目录，找到目录项(或I节点号)；
- ② 根据文件号查系统打开文件表，看文件是否已被打开；
是 → 共享计数加1
否则 → 将目录项(或I节点)等信息填入系统打开文件表空表项，共享计数置为1；
- ③ 根据打开方式、共享说明和用户身份检查访问合法性；
- ④ 在用户打开文件表中获取一空表项，填写打开方式等，并指向系统打开文件表对应表项

返回信息：fd：文件描述符，是一个非负整数，用于以后读写文件



文件操作——指针定位

seek (fd, 新指针的位置)

系统为每个进程打开的每个文件维护一个读写指针，即相对于文件开头的偏移地址（读写指针指向每次文件读写的开始位置，在每次读写完成后读写指针按照读写的数据量自动后移相应数值）

- ① 由fd查用户打开文件表，找到对应的表项；
- ② 将用户打开文件表中文件读写指针位置设为新指针的位置，供后继读写命令存取该指针处文件内容

文件表当中的这个读写指针的位置，对它进行相应的设置 就完成了指针定位的操作



文件操作—读文件

read (文件描述符, 读指针, 要读的长度, 内存目的地址)

- ① 根据打开文件时得到的文件描述符, 找到相应的文件控制块 (目录项)
确定读操作的合法性
读操作合法→②, 否则→出错处理
问题: 文件尚未打开?
- ② 将文件的逻辑块号转换为物理块号
根据参数中的读指针、长度与文件控制块中的信息
定块号、块数、块内位移
- ③ 申请缓冲区
- ④ 启动磁盘I/O操作, 把磁盘块中的信息读入缓冲区, 再
送到指定的内存区 (多次读盘)
- ⑤ 反复执行③、④直至读出所需数量的数据或读至文件



讨论

怎样实现系统调用
rename (给文件重命名) ?

一种呢是用基本的文件操作搭建，第二种呢是直接实现 性能更快



下面我们来介绍文件操作的实现 那么文件操作的实现呢是依赖于具体的文件系统的 我们这里介绍呢, 主要 unix 以为例 但是呢也是介绍了一般文件操作实现的主要步骤。好! 我们来看一下 在文件操作当中的一个重要的操作呢就是创建文件 所谓创建文件呢, 就是建立了这个文件和系统的一个联系 最重要的就是建立这个文件的 FCB 把 FCB 内容填写好。具体来说呢 包括了两个重要的啊步骤 第一个呢是在目录当中为这个新的文件建立一个目录项 那么当然如果是 Unix 文件系统呢, 实际上呢就是目录项再加上 I 节点 目录项建立在目录文件中, I 节点呢是放在了 I 节点区中 如果是 Fat 文件系统, 那么就是建一个目录项 然后是根据提供的参数 填写相应内容, 啊, 一些缺省的值 然后要给这个文件分配必要的存储空间 如果文件需要五块, 你就给它分配五块啊内容 这就是创建文件的一些主要的工作 第二个非常重要的文件操作呢, 是打开文件 在使用文件之前呢, 必须先把文件打开 打开文件呢主要工作呢 首先是根据用户提供的文件名 在文件目录中进行相应的检索 找到这个文件所对应的目录项或 FCB 然后再将对应的目录项或 FCB 把它读入到内存, 保存在内存的 相应的数据结构的地方, 那么这个数据结构 就是我们上一讲所介绍的系统打开文件表 和用户打开文件表。打开文件 这个操作呢, 一般通常系统会返回 一个值, 这个值呢通常称为文件描述符或者是文件句柄 也就是之后对文件的各种操作呢 是通过文件描述符或者是文件句柄来进行的 下面我们具体介绍一下 创建文件的主要工作 创建文件的时候呢, 第一步呢是要检查参数的合法性 啊, 包括了命名啊文件命名是否 符合规则, 有些特殊的字符是不能用于文件名的 另外呢, 有没有重名文件等等 如果合法性检查无误, 那么就进入下一个环节 否则就会报错、返回。第二步呢就是要 申请一个空闲的目录项, 并且把它填填入相关的信息 那么 第三步呢, 是为这个文件申请一些磁盘块 那么当然通常啊建立文件至少是一块分配给这个文件 啊, 当然也可能是一次分配给它几块, 最后这个 文件操作啊创建文件的操作结束, 然后返回 这就是创建文件的这样一个主要步骤 在使用文件之前呢, 我们首先要打开文件 所以打开文件呢, 就是为文件的读写呢做准备工作 用户给出了一个路径名之后, 系统呢会 返回给这个用户一个文件句柄或者是文件描述符 伴随着这个文件打开, 实际上呢这个文件所对应的目录项就要读入内存 下面我们来看一下具体的步骤 那么打开操作当中呢我们给出这样一个打开方式这样一个参数 那么确定这次打开是以读的方式打开这个文件, 还是以写的方式打开这个文件 具体步骤呢, 我们第一步呢是根据路径名去查目录 找到对应的目录项或者是 I 节点号, 那这个过程呢 是个反复的过程, 因为我们是树形

结构我们从根目录开始找，一直找到最后这个文件所对应的这个目录项或者是 I 节点号。根据的这个文件号去查系统打开文件表，看一看这个文件的之前是不是被别的进程打开过，如果是，那么共享计数就加 1 就可以了。所以在系统打开文件表当中的，共享计数加 1。如果这个文件之前没有其他进程打开，就要把这个文件所对应的 I 节点或者是目录项这些信息读入内存，保存在系统打开文件表中，并且把共享计数呢设置成 1。当然这个过程当中还会根据啊打开的方式，共享说明和用户身份的检查等，来检查这次访问的合法性。之后呢会在用户打开表当中呢获取一个空的表项来填写打开方式啊，系统打开表这个对应的指针啊。这样一些信息最后呢会返回给用户一个文件描述符或者文件句柄。这是打开文件操作的主要步骤。我们再介绍一个叫指针定位操作。指针定位操作呢，也就是说系统会为每个进程它打开的每个文件都维护一个读写指针。也就是说这个操作，啊，这个用户对这个文件的操作是从这个读写指针开始，它呢是相当于对文件头的一个偏移。每次做完相应的操作之后，都会去修改这个读写指针，把它移位到一个正确的一个偏移的位置。那么这个指针定位操作呢，其实主要是，因为这个文件已经打开了，所以主要是根据它的文件对应的描述符 fd 啊文件描述符根据这个描述符去查什么呢，用户打开文件表找到对应的表项，然后根据用户打开文件表当中的这个读写指针的位置，对它进行相应的设置就完成了指针定位的操作。[无声] 对文件来讲，最重要的呢是读或者是写文件。我们就以读文件为例来考虑啊这个主要的一些步骤。那么在读文件当中呢，读一个文件呢首先要给出文件描述符，要给出指针的位置从什么地方开始读。当然了缺省的话，应该是从文件头开始读。你读多长的字节以及你要把读入的内容啊保存在内存的什么位置，也就是它的目的，目的地好，那么我们看一下主要步骤。根据打开文件时得到的这个文件描述符我们会去找到对应的文件控制块或者是目录项啊，就是通过啊查系统打开文件表就找到相应内容了。确定读操作的合法性，因为有的时候你读一个这个文件，这里头有一些，比如说指针呐、位置啊等等可能出现了问题。所以呢要确定读操作的合法性，如果合法继续可以走，那么否则就出错啦，就报错、返回。这里头有的时候有这样一个问题，就是说如果文件没有打开怎么办？当然有的时候呢在一些系统当中呢它要求你必须先打开文件再做后续的操作。也有些系统呢，允许你在读的时候如果这文件没有打开，那么就要隐隐地把它打开，就隐式打开文件。这是不同的系统不同的做法。好，那么下面呢就是说要将文件的逻辑块号转换成物理块

号，因为我们已经有了文件控制块的相关信息，那么你要读的这个指针实际上对应了可以，对应的一个文件的逻辑块号。我们通过 FCB 当中的相关信息呢，把它转换成物理块号。所以这是根据参数中的读指针、长度与文件控制块当中的信息来进行块的位移和块内偏移这样一些数字的计算。你要读信息呢通常要申请一个缓冲区，把这个信息读入缓冲区。然后呢，启动磁盘 I/O 操作，我们前面也讲过了，文件系统当中后面接着就是 I/O，啊，I/O，然后启动 I/O 之后把相应内容去读入缓冲区，然后再送到指定的内存去。这里头会有多次的这个循环过程。直到把你所需要的要读多少长度，都读入到内存，这个读操作结束。那么介绍了这个几个基本的最常用的操作之外呢，我们现在讨论一下大家思考一下，怎么样来实现系统调用，rename 这个系统调用。当然我们知道这个系统调用是给文件重命名。请大家先回忆一下，在我们的上一讲当中，我们介绍如何用基本文件的操作来搭建一个 copy 操作。同样道理呢，我们也可以用文件的基本操作来搭建怎么样去实现系统调用。但是呢，如果我们直接就要实现系统调用的话呢，这样性能更快一点。所以希望大家考虑，怎么样直接实现这个系统调用。主要呢提示一下，就是根据文件名进行目录检索，找到对应的 FCB 或者目录项，其实直接把这个名字原来的一个名字，把它改成新的名字。直接在这个文件的 FCB 或者是目录项当中把它文件名改了就可以了，所以这样呢是最快的。不需要把这个文件再去读进内存啊，直接就把 FCB 内容直接就改了，保存回去就可以了。这就是系统调用 rename 的一个实现，它通常是两种方式。一种呢是用基本的文件操作搭建，第二种呢是直接实现性能更快。