说明：本视频对应王道书 4.1.3

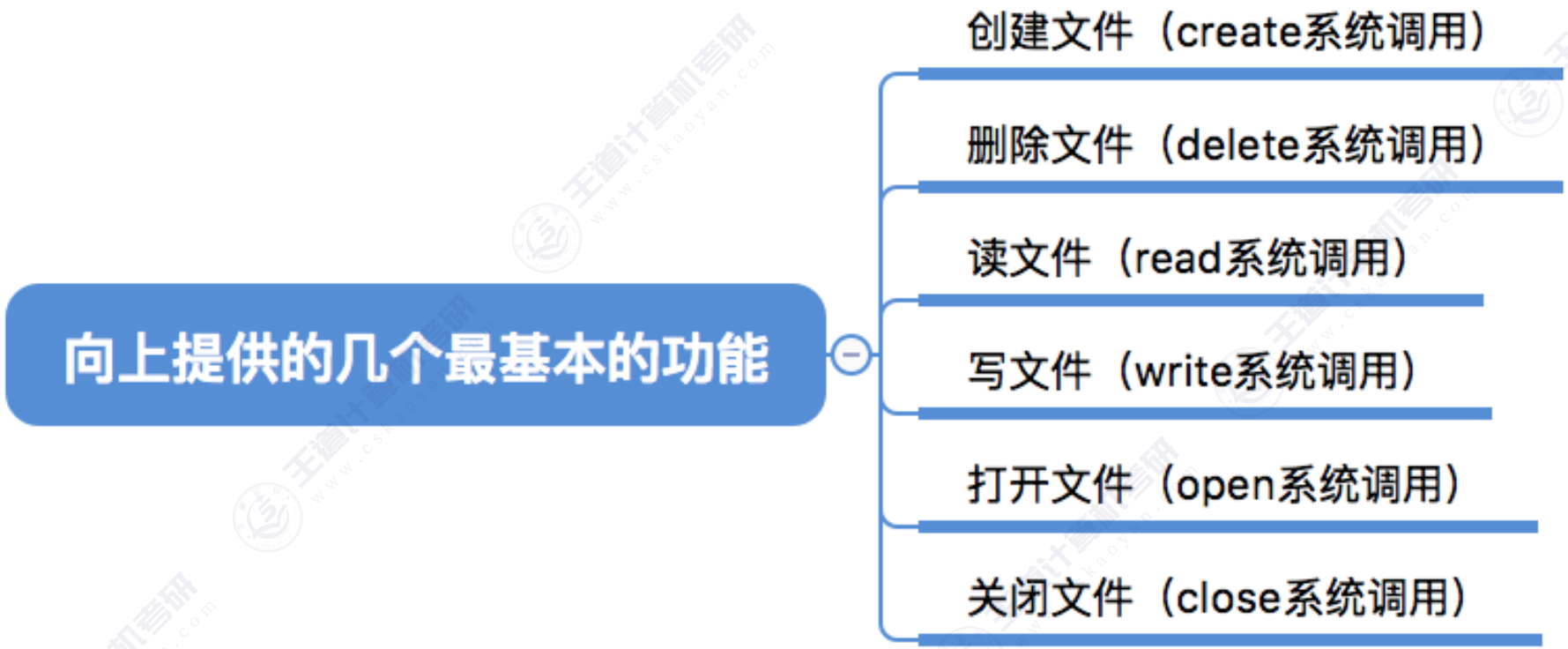
在前面几个视频中，我们先学习了文件的逻辑结构、文件的物理结构、文件目录相关的知识。在学完前面这些知识后，再学习“4.1.3 文件的操作”会更好理解。

建议：学完本视频后，可以接着阅读王道书 4.1.3

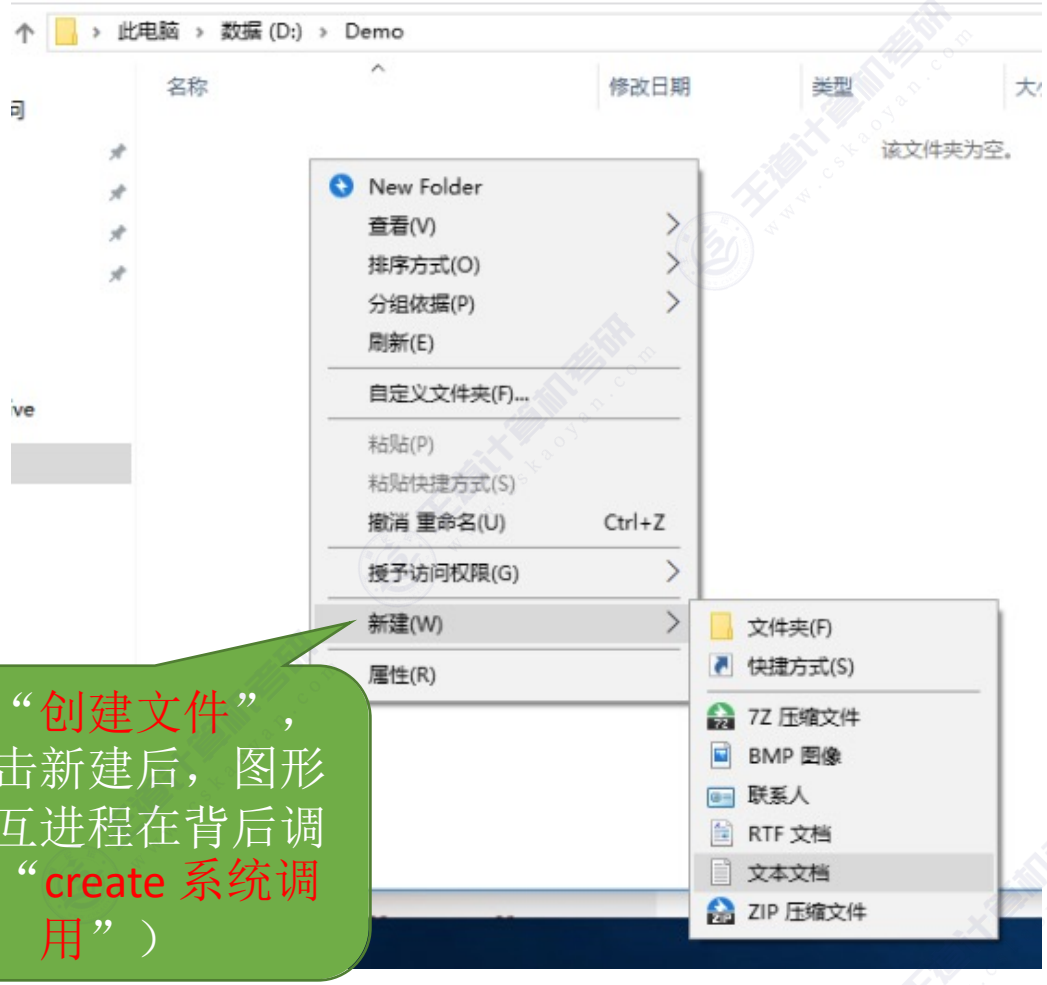
本节内容

文件的基本操作

知识总览



创建文件



可以“创建文件”，（点击新建后，图形化交互进程在背后调用了“create 系统调用”）

进行 Create 系统调用时，需要提供的几个主要参数：

1. 所需的外存空间大小（如：一个盘块，即1KB）
2. 文件存放路径（“D:/Demo”）
3. 文件名（这个地方默认为“新建文本文档.txt”）

操作系统在处理 Create 系统调用时，主要做了两件事：

1. 在外存中找到文件所需的空间（结合上小节学习的空闲链表法、位示图、成组链接法等管理策略，找到空闲空间）
2. 根据文件存放路径的信息找到该目录对应的目录文件（此处就是 D:/Demo 目录），在目录中创建该文件对应的目录项。目录项中包含了文件名、文件在外存中的存放位置等信息。

删除文件

可以“删除文件”（点了“删除”之后，图形化交互进程通过操作系统提供的“删除文件”功能，即 delete 系统调用，将文件数据从外存中删除）

进行 Delete 系统调用时，需要提供的几个主要参数：

1. 文件存放路径（“D:/Demo”）
2. 文件名（“test.txt”）

操作系统在处理 Delete 系统调用时，主要做了几件事：

1. 根据文件存放路径找到相应的目录文件，从目录中找到文件名对应的目录项。
2. 根据该目录项记录的文件在外存的存放位置、文件大小等信息，回收文件占用的磁盘块。（回收磁盘块时，根据空闲表法、空闲链表法、位图法等管理策略的不同，需要做不同的处理）
3. 从目录表中删除文件对应的目录项。

打开文件

用户进程A的“打开文件表”

编号	文件名	…
1	…	…
2	test.txt	…
3		

复制

文件名	…	外存中的位置
test.txt		…
…	…	…

之后用户进程A再操作文件就不需要每次都重新查目录了，这样可以加快文件的访问速度

内存

外存

在很多操作系统中，在对文件进行操作之前，要求用户先使用 open 系统调用“打开文件”，需要提供的几个主要参数：

1. 文件存放路径（“D:/Demo”）
2. 文件名（“test.txt”）
3. 要对文件的操作类型（如：r 只读；rw 读写等）

操作系统在处理 open 系统调用时，主要做了几件事：

1. 根据文件存放路径找到相应的目录文件，从目录中找到文件名对应的的目录项，并检查该用户是否有指定的操作权限。
2. 将目录项复制到内存中的“打开文件表”中。并将对应表目的编号返回给用户。之后用户使用打开文件表的编号来指明要操作的文件。

开文件

读/写指针记录了该进程对文件的读/写操作进行到的位置

如果打开文件时声明的是“只读”，则该进程不能对文件进行写操作

记录此时有多少个进程打开了此文件

编号	文件名	读写指针	访问权限	系统表索引号
1	…	…	…	
2	test.txt	…	只读	k
3				

用户进程A的“打开文件表”

编号	文件名	读写指针	访问权限	系统表索引号
1	test.txt	…	读和写	k

用户进程B的“打开文件表”

编号	文件名	…	外存地址	打开计数器
1	…		…	…
…	…		…	…
k	test.txt		…	2
…	…			

系统的打开文件表（整个系统只有一张）

可以方便实现某些文件管理的功能。例如：在Windows系统中，我们尝试删除某个txt文件，如果此时该文件已被某个“记事本”进程打开，则系统会提示我们“暂时无法删除该文件”。其实系统在背后做的事就是先检查了系统打开文件表，确认此时是否有进程正在使用该文件。

关闭文件

记录此时有多少个进程打开了此文件

编号	文件名	读写指针	访问权限	系统表索引号
1	…	…	…	
2	test.txt	…	只读	k
3				

用户进程A的“打开文件表”

编号	文件名	…	外存地址	打开计数器
1	…		…	…
…	…		…	…
k	test.txt		…	2 1
…	…			

系统的打开文件表（整个系统只有一张）

编号	文件名	读写指针	访问权限	系统表索
	xt		写	

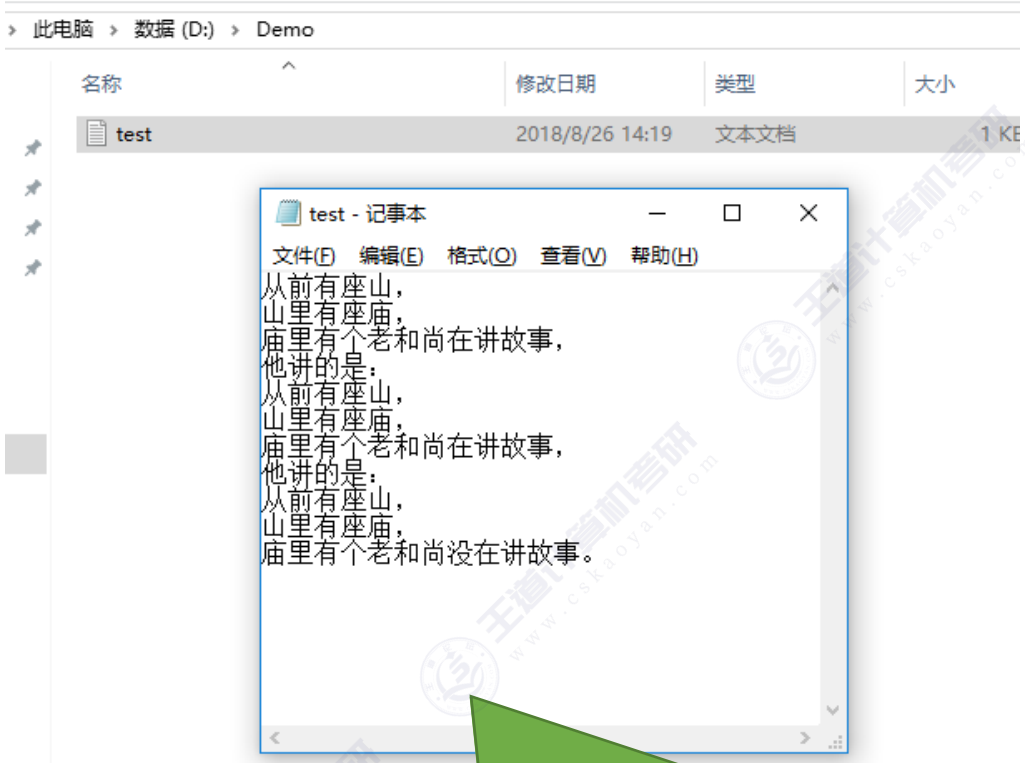
用户进程B的“打开文件表”

进程使用完文件后，要“关闭文件”

操作系统在处理 Close 系统调用时，主要做了几件事：

1. 将进程的打开文件表相应表项删除
2. 回收分配给该文件的内存空间等资源
3. 系统打开文件表的打开计数器count 减1，若 count = 0，则删除对应表项。

读文件



可以“读文件”，将文件数据读入内存，才能让CPU处理（双击后，“记事本”应用程序通过操作系统提供的“读文件”功能，即 read 系统调用，将文件数据从外存读入内存，并显示在屏幕上）

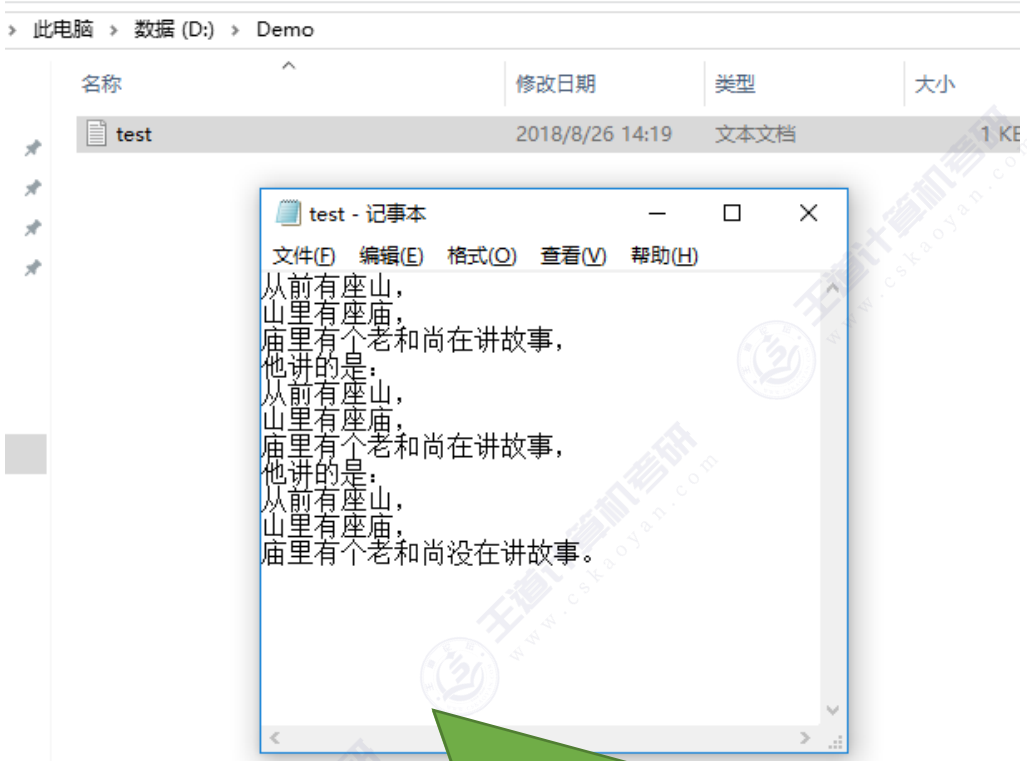
编号	文件名	读写指针	访问权限	…
1	test.txt	…	读和写	…

“记事本”进程的“打开文件表”

进程使用 read系统调用完成写操作。需要指明是哪个文件（在支持“打开文件”操作的系统中，只需要提供文件在打开文件表中的索引号即可），还需要指明要读入多少数据（如：读入 1KB）、指明读入的数据要放在内存中的什么位置。

操作系统在处理 read 系统调用时，会从读指针指向的外存中，将用户指定大小的数据读入用户指定的内存区域中。

写文件



可以“写文件”，将更改过的文件数据写回外存（我们在“记事本”应用程序中编辑文件内容，点击“保存”后，“记事本”应用程序通过操作系统提供的“写文件”功能，即 write 系统调用，将文件数据从内存写回外存）

编号	文件名	读写指针	访问权限	…
1	test.txt	…	读和写	…

“记事本”进程的“打开文件表”

进程使用 write 系统调用完成写操作，需要指明是哪个文件（在支持“打开文件”操作的系统中，只需要提供文件在打开文件表中的索引号即可），还需要指明要写出多少数据（如：写出 1KB）、写回外存的数据放在内存中的什么位置

操作系统在处理 write 系统调用时，会从用户指定的内存区域中，将指定大小的数据写回写指针指向的外存。

知识点回顾与重要考点

思考：文件重命操作应该做些什么？

文件的基本操作

- 创建文件：分配外存空间，创建目录项
- 删除文件：回收外存空间，删除目录项
- 打开文件：
  - 将目录项中的信息复制到内存中的打开文件表中，并将打开文件表的索引号返回给用户
  - 打开文件之后，对文件的操作不再需要每次都查询目录，可以根据内存中的打开文件表进行操作
  - 每个进程有自己的打开文件表，系统中也有一张总的打开文件表
  - 进程打开文件表中特有的属性：读写指针、访问权限（只读？读写？）
  - 系统打开文件表中特有的属性：打开计数器（有多少个进程打开了该文件）
- 关闭文件：
  - 将进程打开文件表中的相应表项删除
  - 系统打开文件表的打开计数器减1，若打开计数器为0，则删除系统表的表项
- 读文件：根据读指针、读入数据量、内存位置 将文件数据从外存读入内存
- 写文件：根据写指针、写出数据量、内存位置 将文件数据从内存写出外存

打开文件时并不会把文件数据直接读入内存。“索引号”也称“文件描述符”

“读/写文件”用“文件描述符”即可指明文件，不再需要用到“文件名”

公众号：王道在线

b站：王道计算机教育

抖音：王道计算机考研