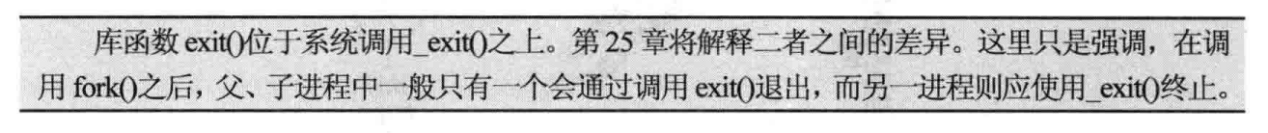
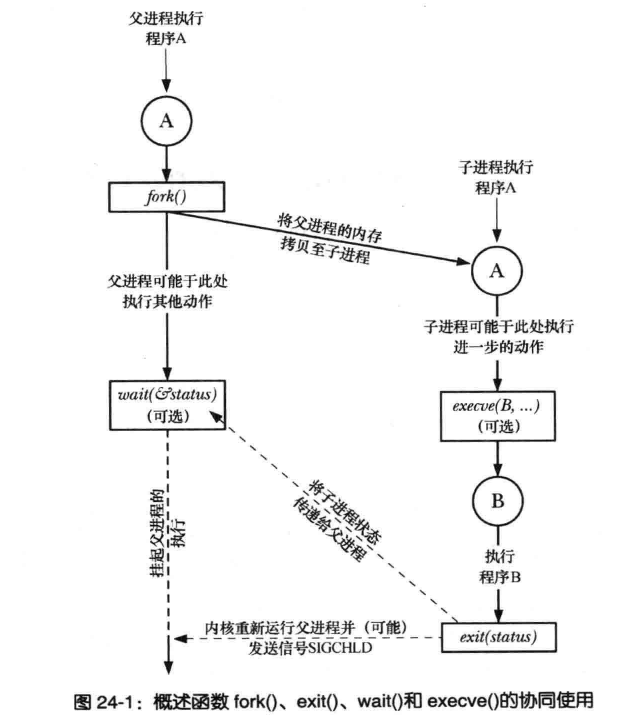
第二十四章

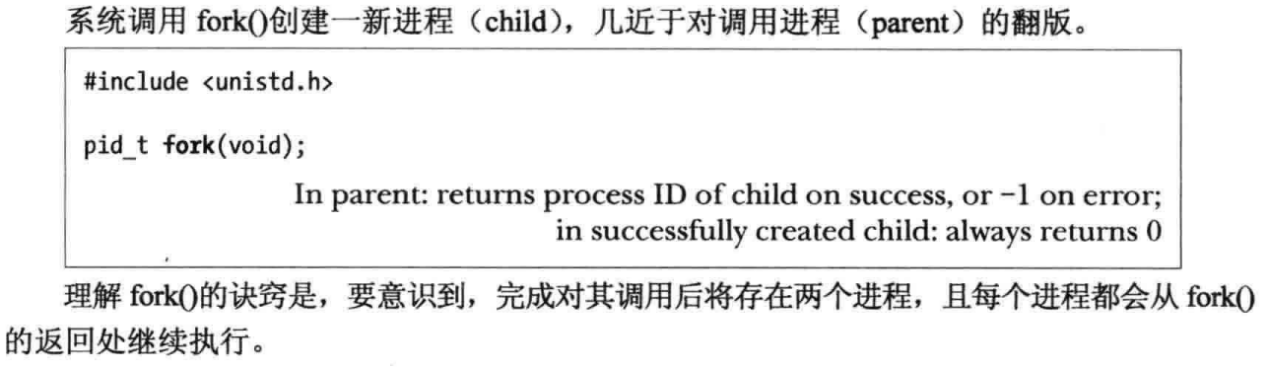
1. 遇到一个问题：



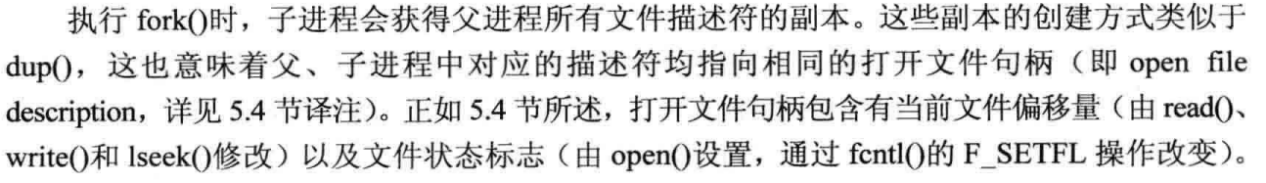
1. 概括地介绍了接下来几章的内容：

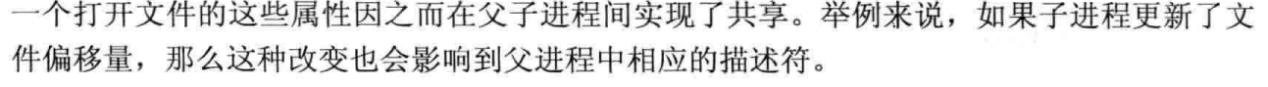


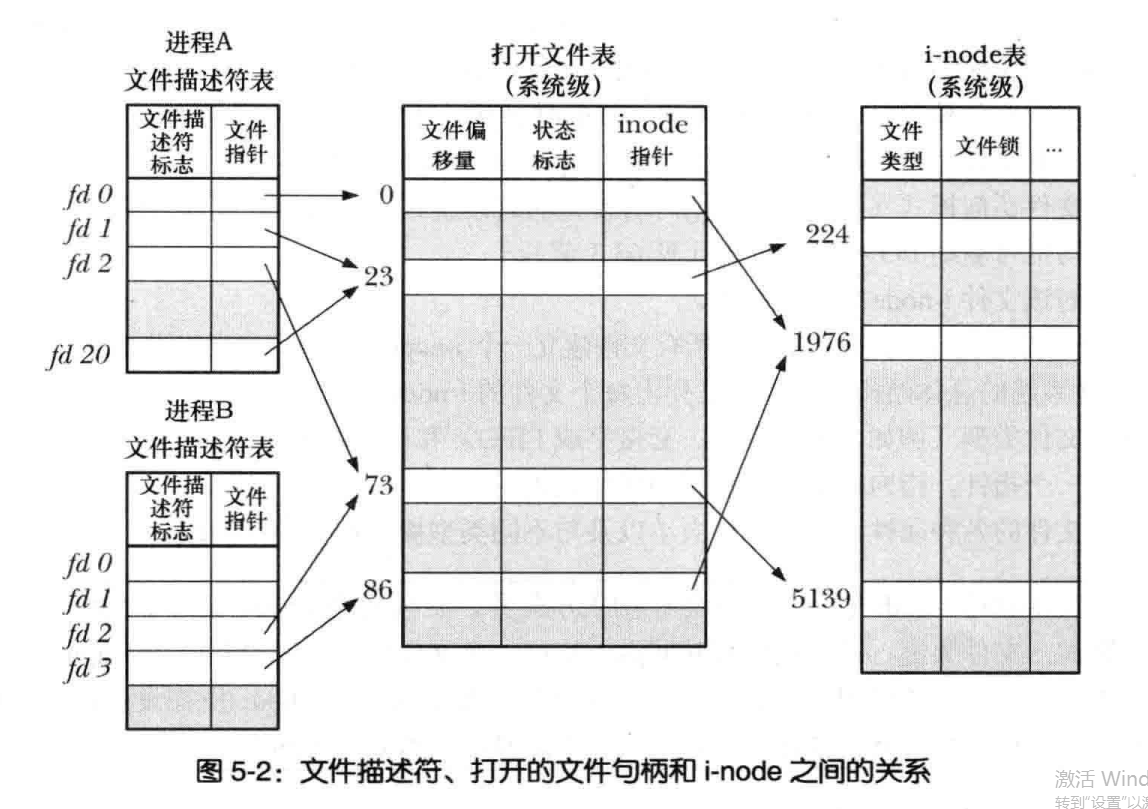
1. 创建新进程：



1. 父子进程的文件共享：



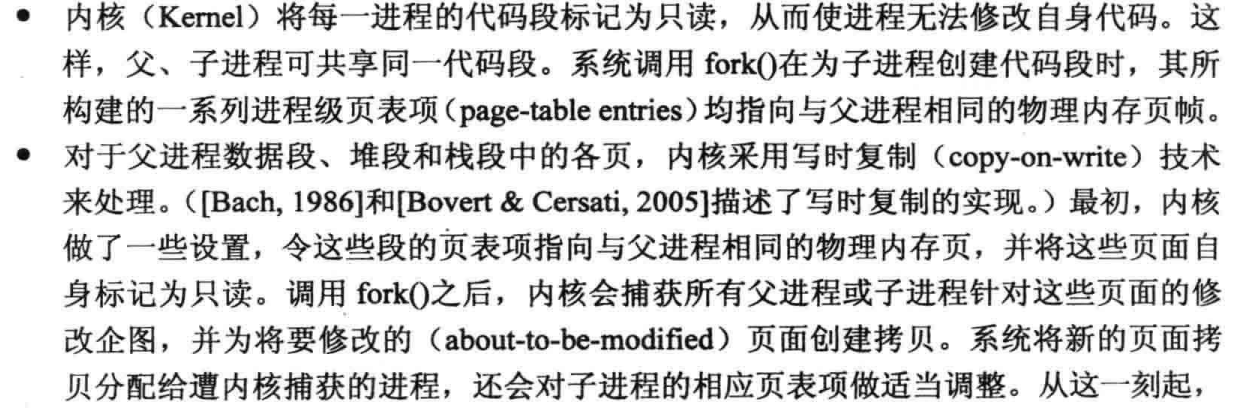




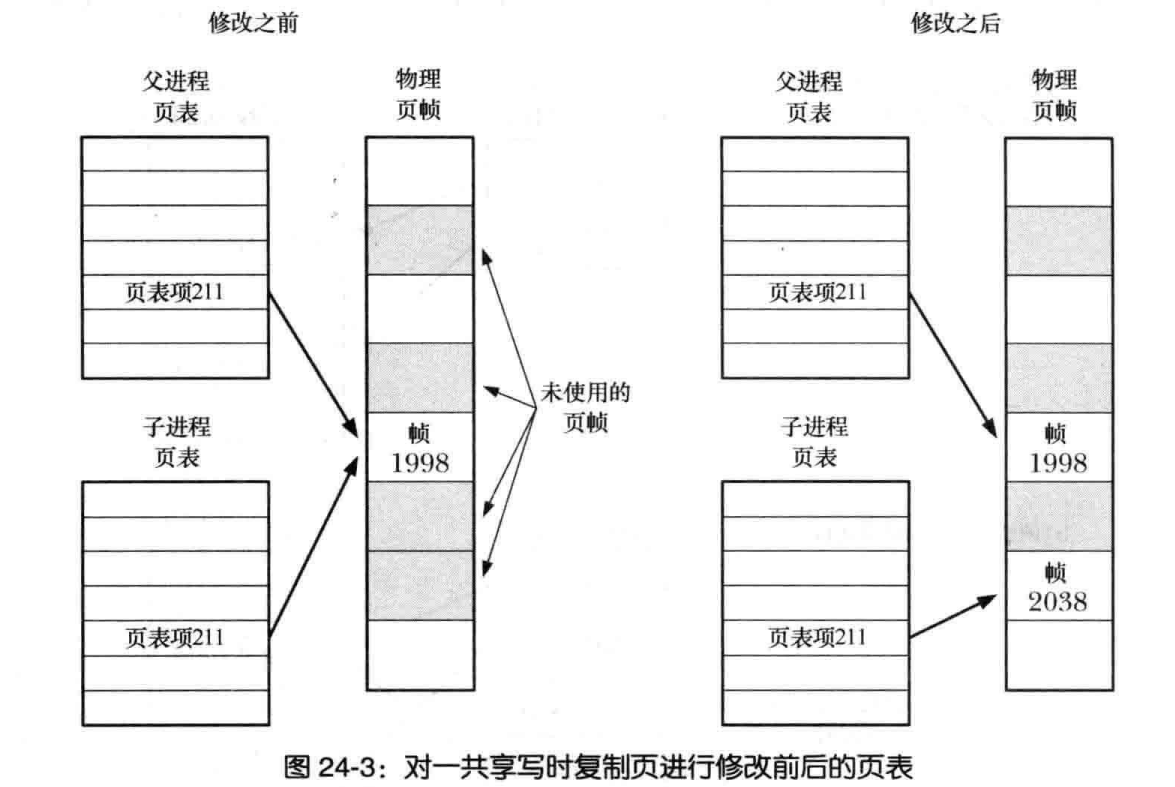
如上图的标号为73的文件句柄就可能由进程A fork 进程B 出现的情况。

1. Fork的内存语义：

即对于代码段采用只读方式共享物理内存，堆段和数据段也是共享内存，某一方试图改变的时候，采用“写时复制”的方式。



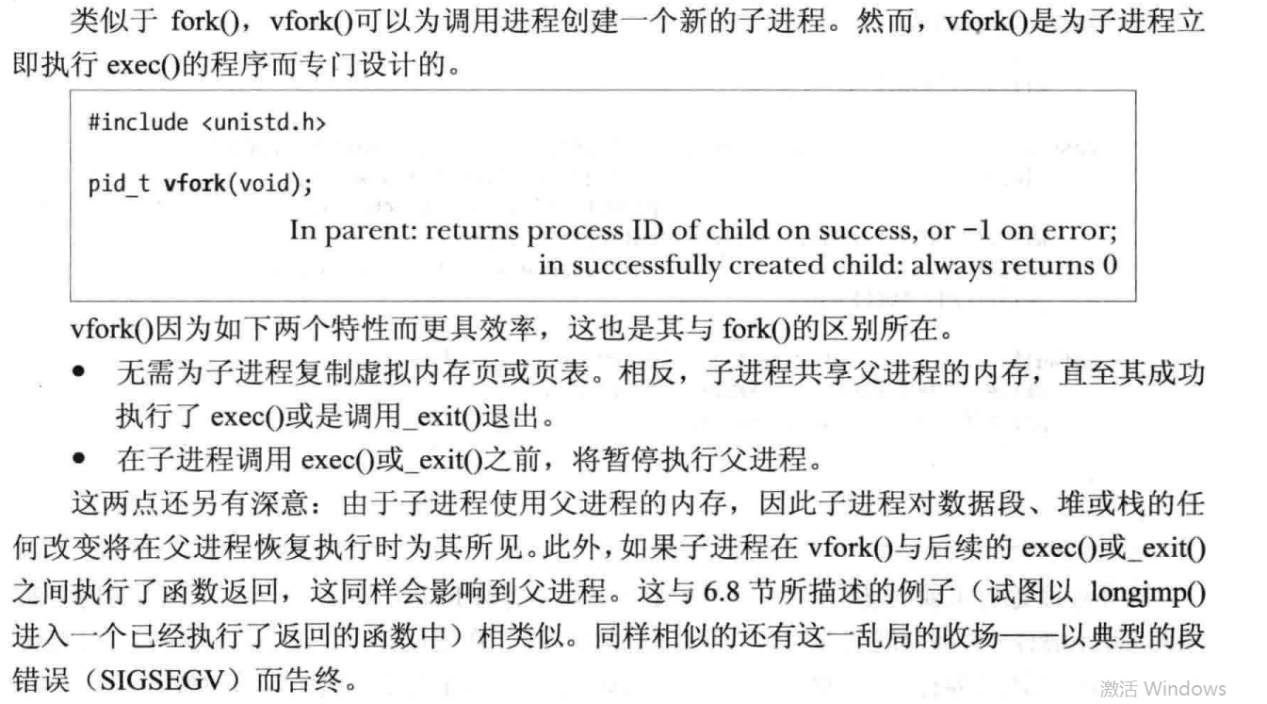




此外，利用fork，还可以把可能造成内存泄漏的程序放到子进程里执行，然后把结果返回给父进程，这样，父进程的内存使用量就不变，达到了控制进程内存的需求。

1. 系统调用vfork

此系统调用已废除，不要再使用。



注意，在exec之前，此系统调用的子进程与父进程共享内存，会导致很多奇怪的问题和限制。

进一步查阅资料得知，vfork之后，子进程系统进程表的页表和虚拟地址空间都是直接引用父进程的，只有在退出或运行程序才会脱离父进程，否则有可能改变父进程的内存空间。

比如，这个程序子进程使用return而不是\_exit或者exec的话，就会造成一定的错误（比如说死循环或者段错误）。延申出来，我更倾向于父子进程有各自的寄存器，而子进程return之后的某些操作导致栈的内容改变了，所以父进程从堆栈内容返回的时候，堆栈内容被改变，所以才会有不确定的错误。

1. Fork之后的竞争条件：

不要把fork之后的运行结果依赖于系统的调度，应该采用特定的约束方法。

1. 同步信号规避竞争条件：

可以使用信号避免竞争。