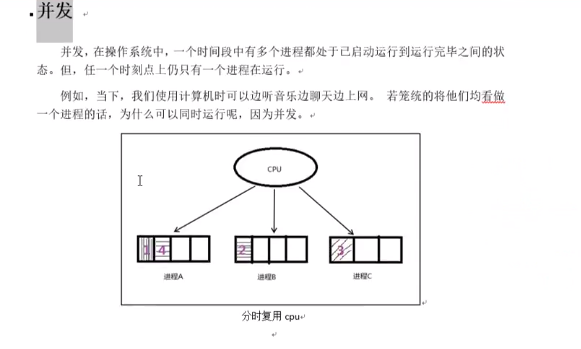
# 进程

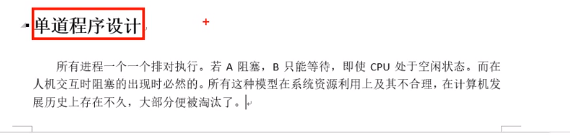
程序：死的。只占用磁盘空间

进程：活的。运行起来的程序。占用内存、cpu等系统资源

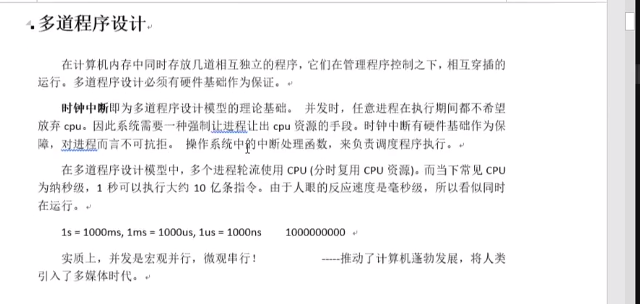
并发：



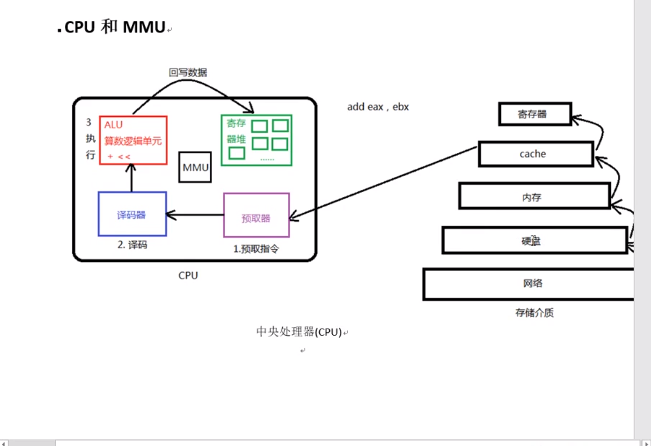
单道程序设计：串行



多道程序设计：并行



CPU和MMU:



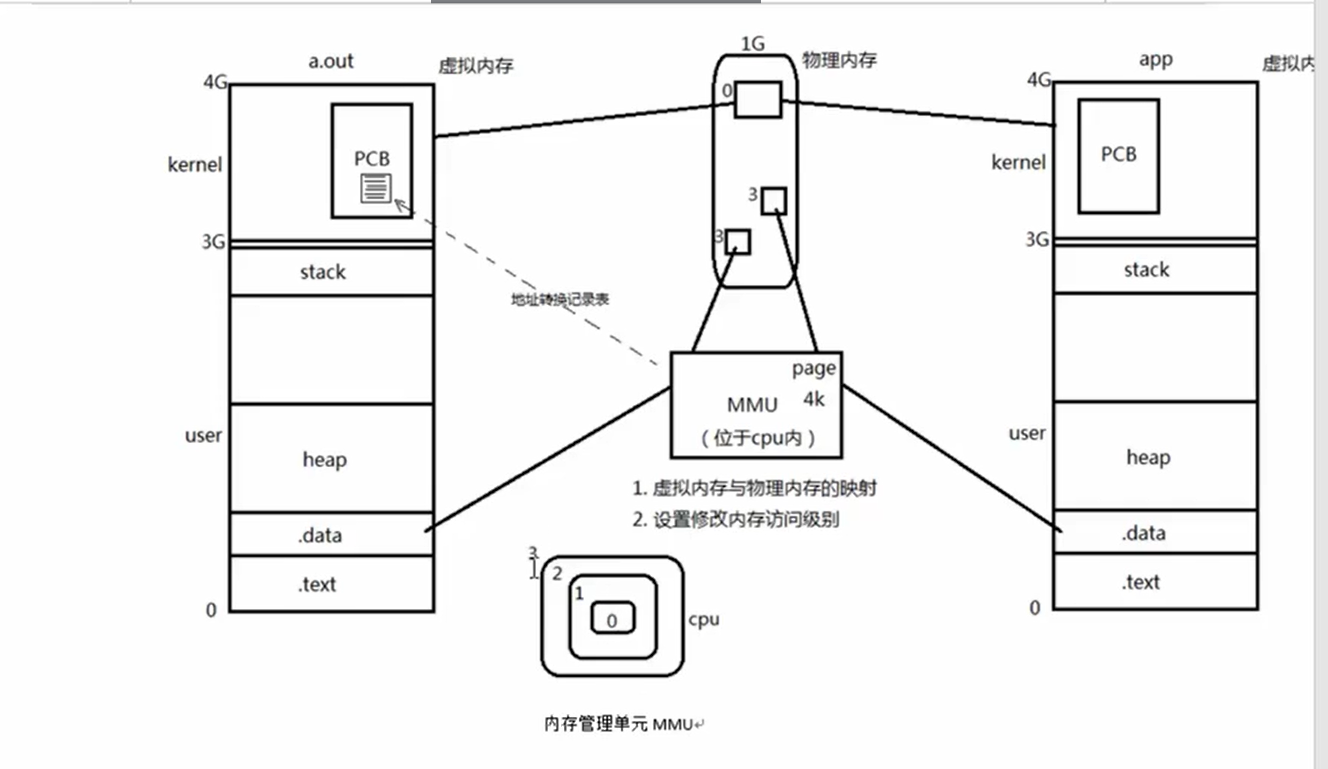
一个寄存器大小：4K。

MMU：虚拟内存映射单元

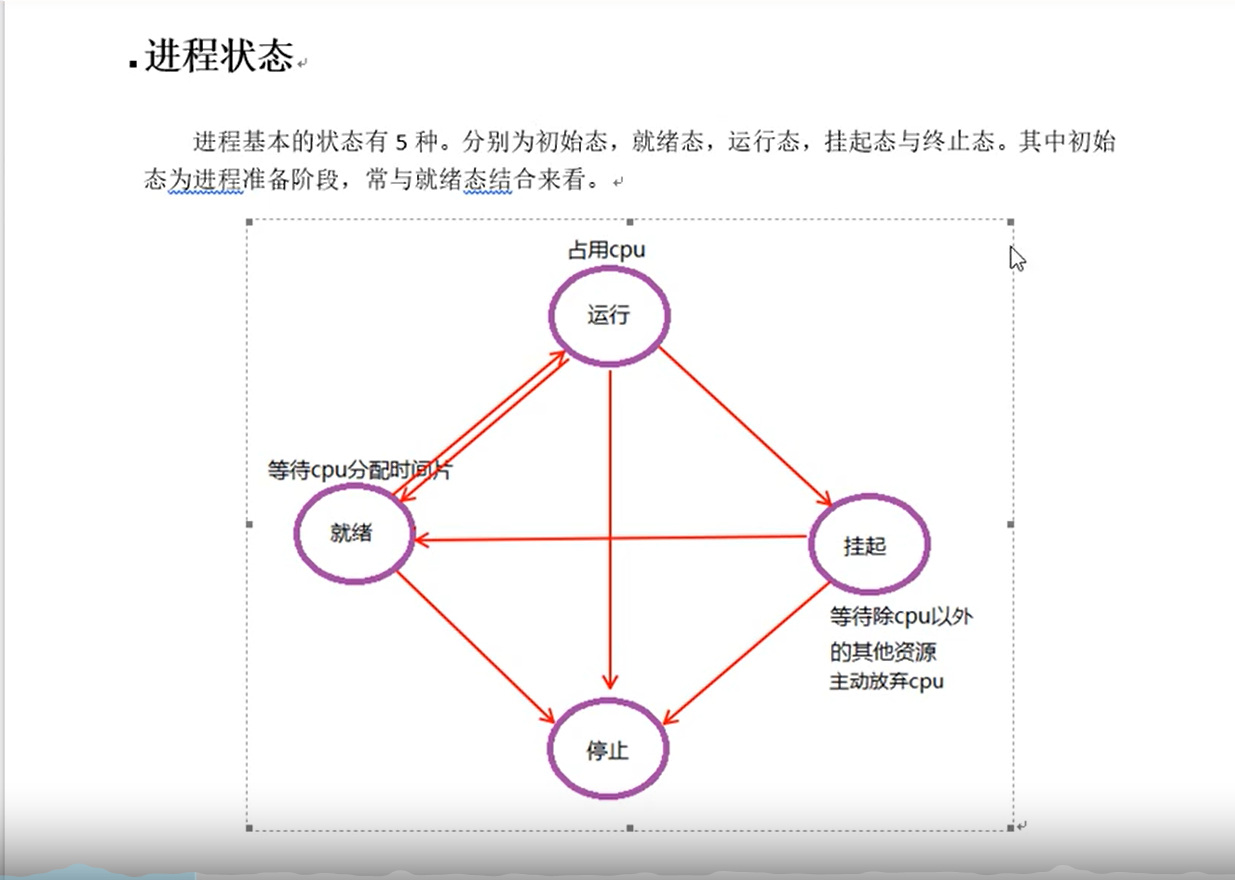
虚拟内存和物理内存映射关系：

一页的大小是4K

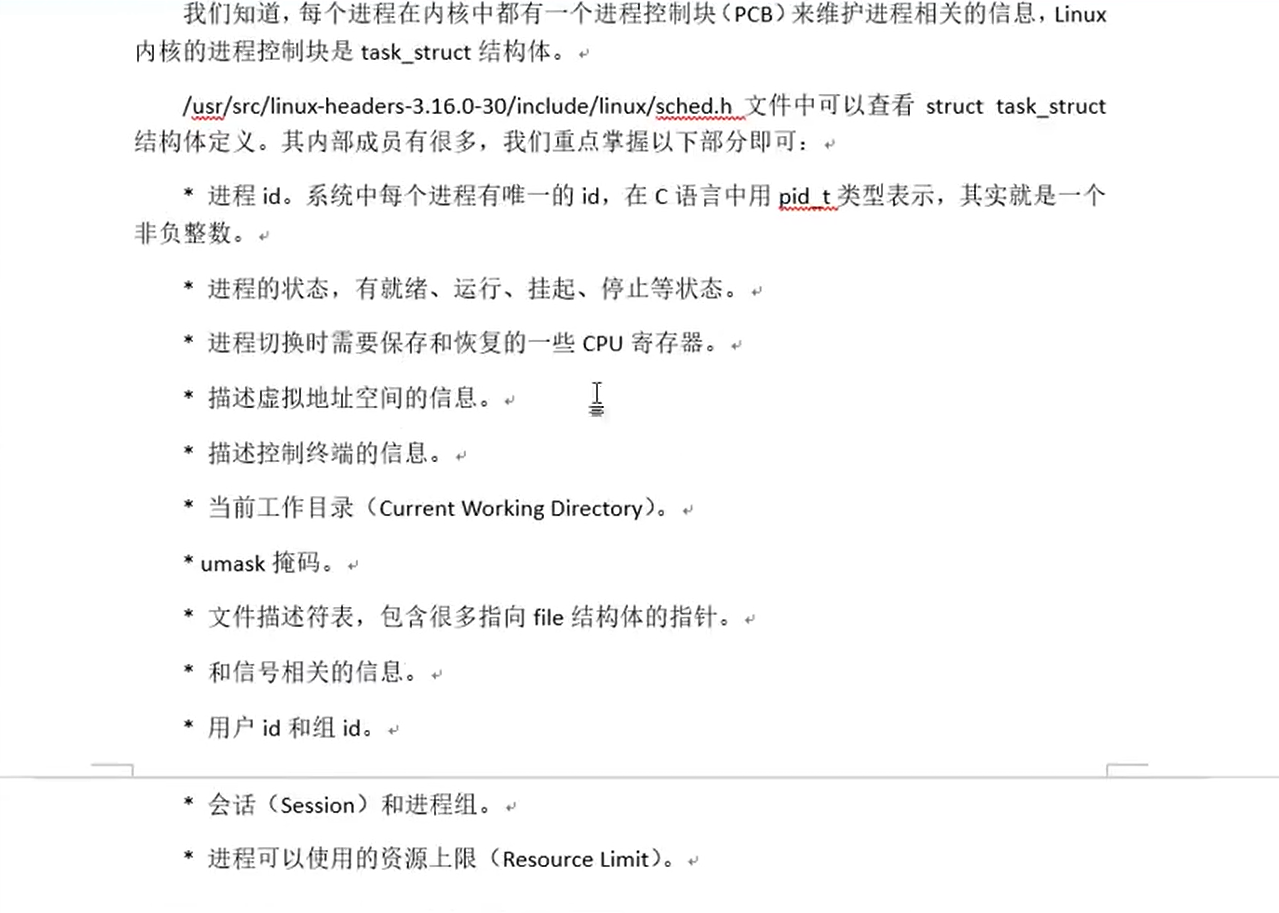
内核空间是共享的，不同进程的内核空间映射到同一块内核内存



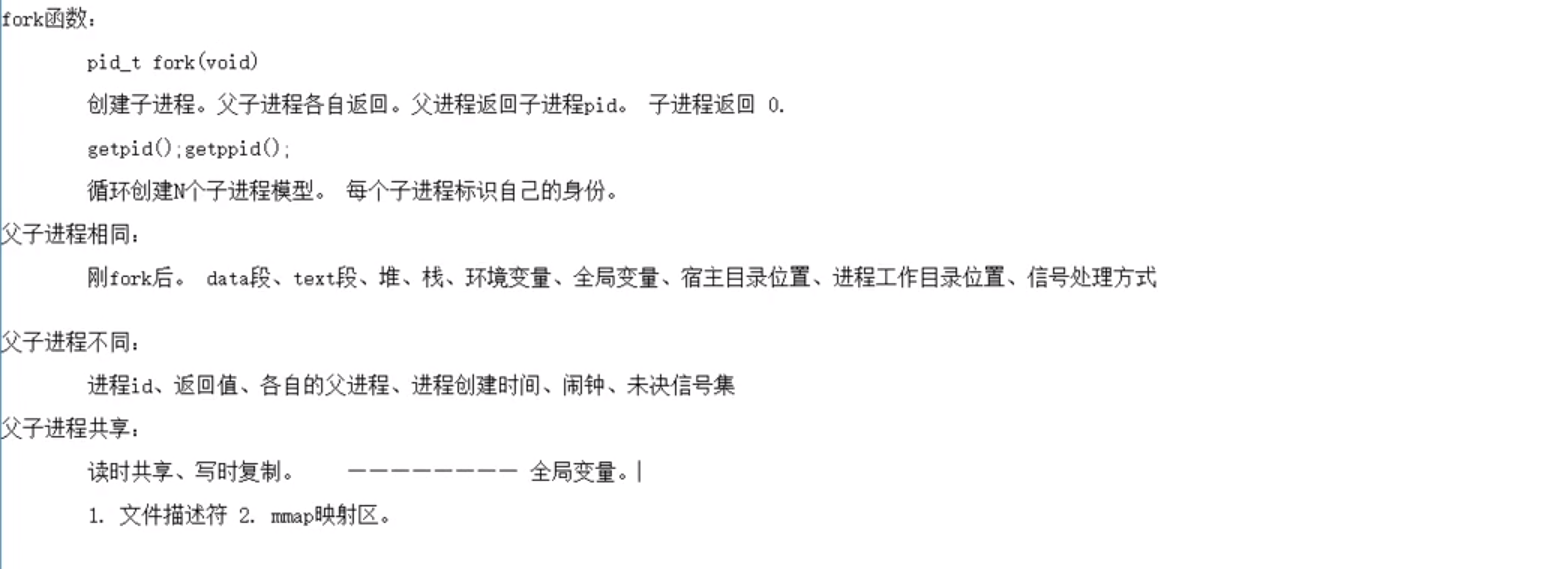
进程的5种状态：



PCB进程控制块：



fork函数：



父子进程gdb调试：

Set follow-fork-mode child 跟随子进程进行调试

Set follow-fork-mode parent 跟随父进程进行调试

Exec函数族：

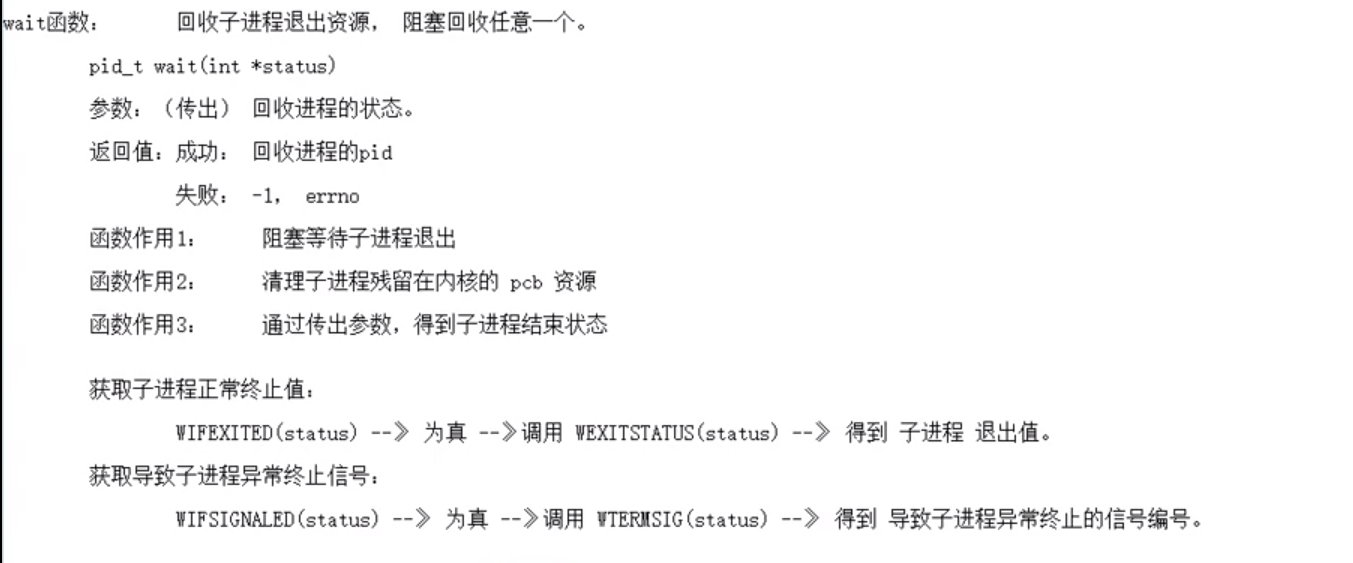


孤儿进程：父进程先于子进程结束

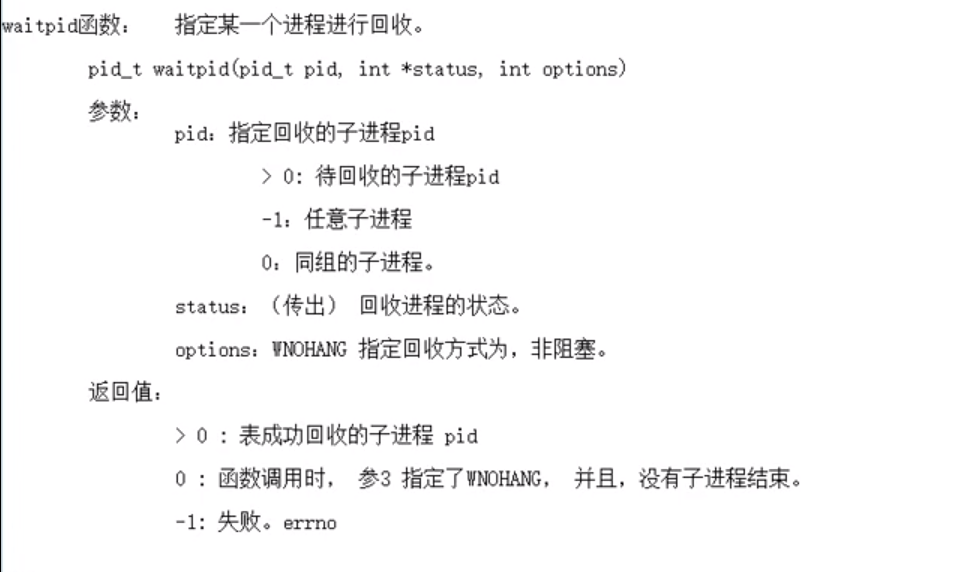
僵尸进程：进程终止，父进程尚未回收，子进程残留资源（PCB）存放于内核中，变成僵尸进程

Wait函数：回收进程终止信息

一次wait函数调用只能回收一个子进程



Waitpid函数：可根据子进程PID指定回收子进程



Wite和witepid函数回收多个子进程：

采用循环回收

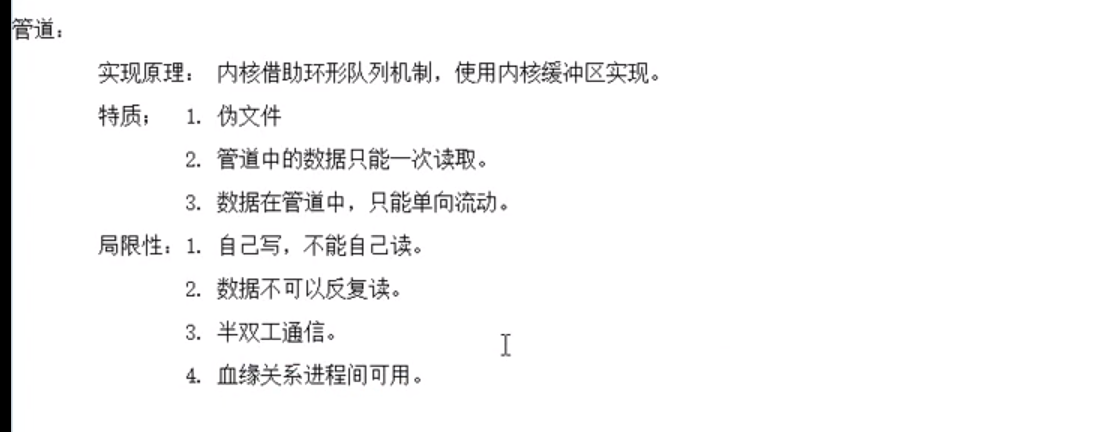
进程间的通讯：本质就是内核空间中的一块缓冲区（一般都是4096大小）

1. 管道（使用最简单）
2. 信号（开销最小）
3. 共享映射区（没有血缘关系）
4. 本地套接字（最稳定）

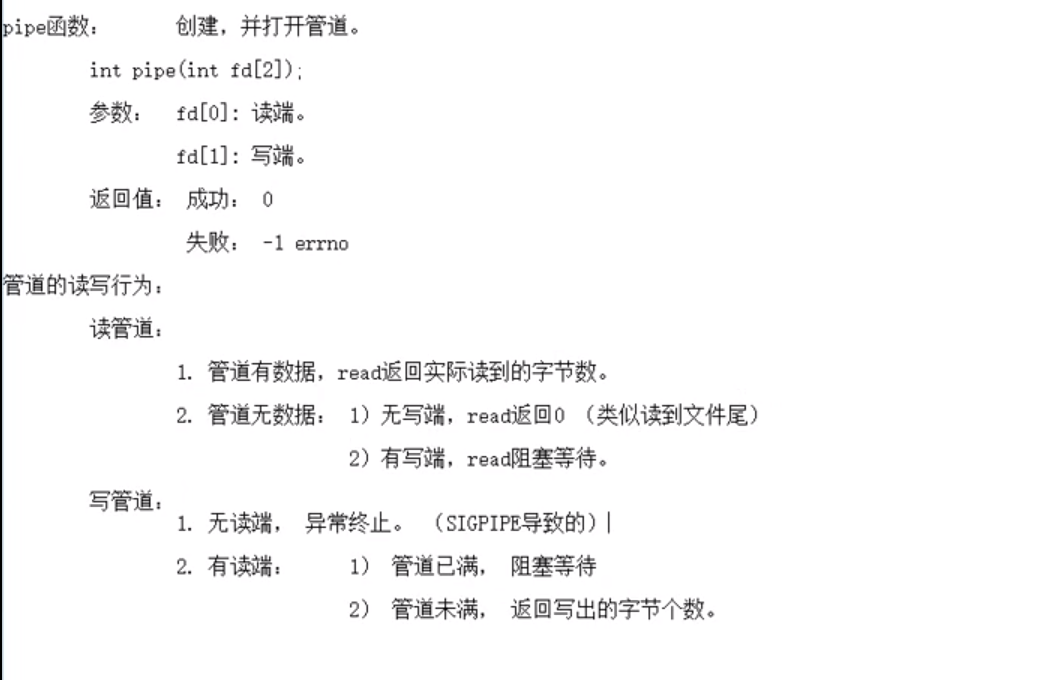
管道：

Pipe/mkfifo函数进行创建

本质是循环队列，双向半双工，只能在有公共祖先之间的进程间使用



Pipe函数：创建并打开管道



管道满了linux内核会自动扩容

wc命令：统计行数

pipe函数创建出的管道用于有血缘关系的进程间的通讯。

父进程和子进程间的通讯中：父进程关闭读端进行写操作，子进程关闭写端进行读操作，也可以反过来，尽量保证管道中数据单向流通。

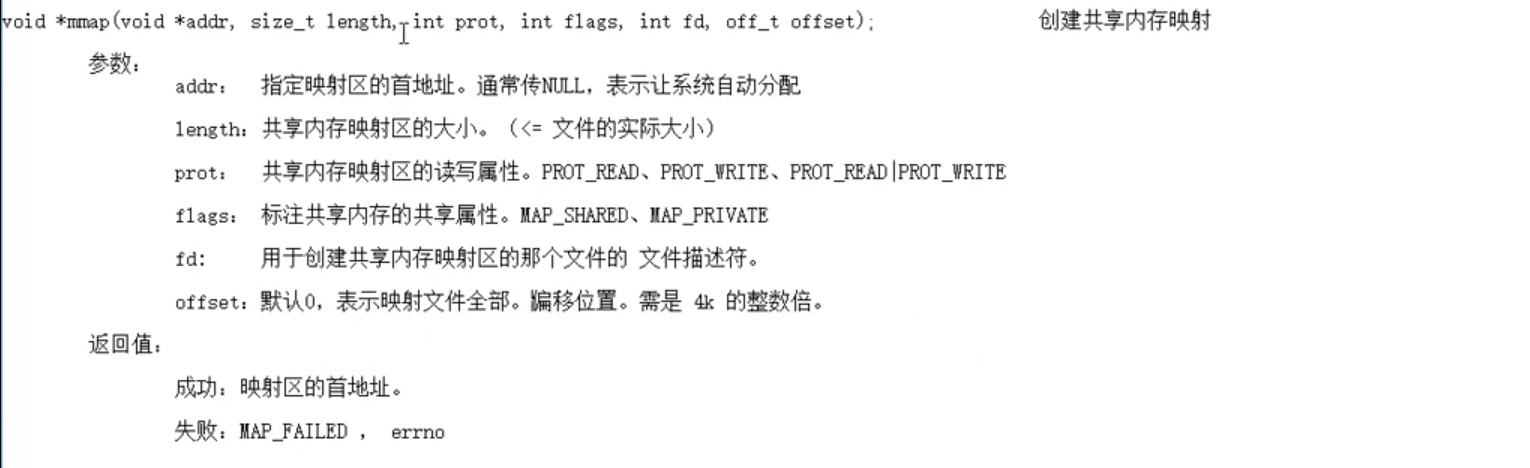
兄弟进程间的通讯：父进程创建两个子进程，父进程关闭管道的两端，两个子进程各关闭一端进行通讯。

mkfifo函数：创建有名管道，适用于没有血缘关系的进程之间的通讯。本质上是创建一个文件进行通讯

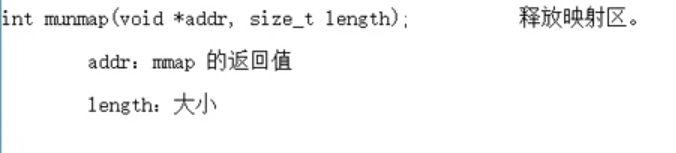
读普通文件不会堵塞，读管道文件和套接字会堵塞

存储映射：将磁盘上的文件映射到内存，这样就可以使用指针对文件进行操作

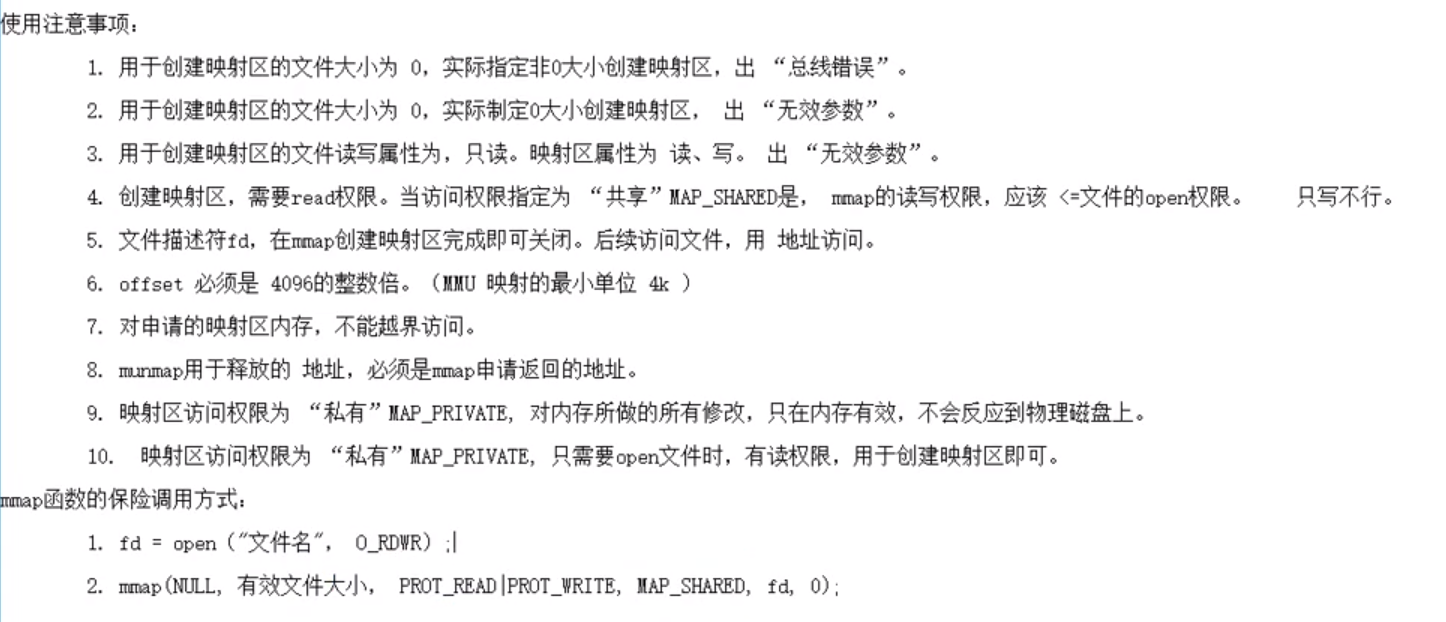
mmap函数：创建共享内存映射



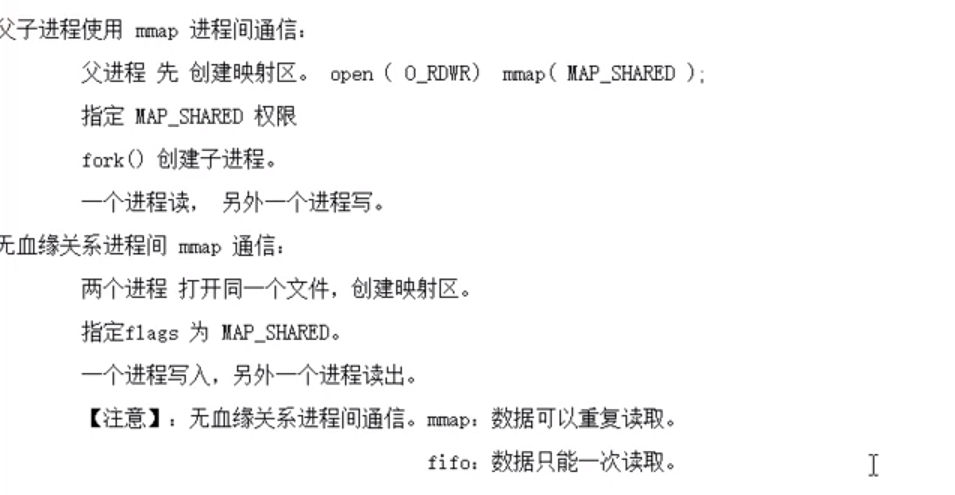
munmap函数：释放映射区



使用mmap函数的注意事项：



Mmap进行进程间通讯：



匿名映射区：

