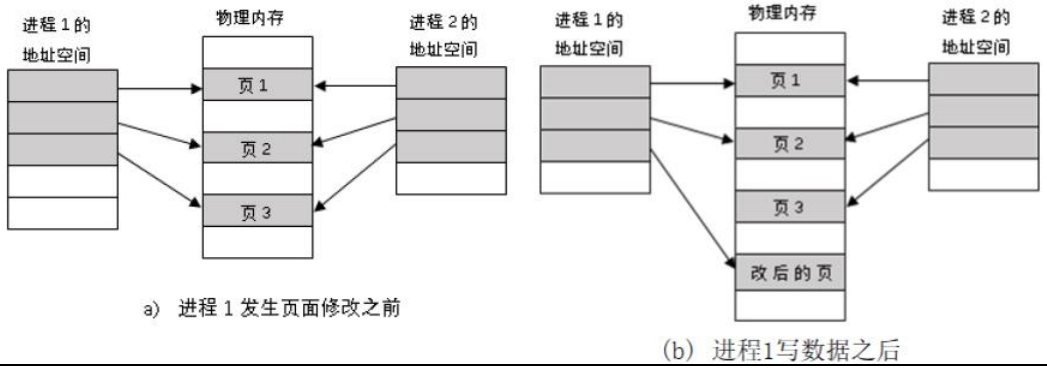
- 进程与程序的区别与联系
- 进程与线程的区别与联系
- 写时拷贝/写时复制



进程的定义：

进程有以下四个特征：
动态性、并发性、独立性、异步性

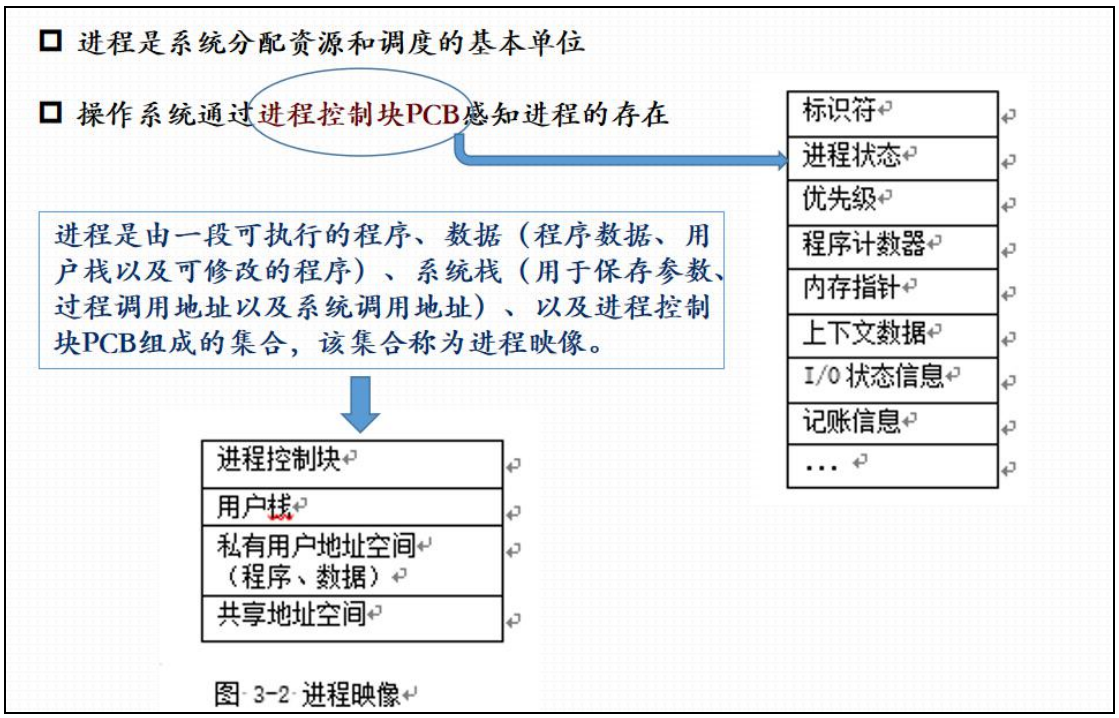
- 是程序的一次执行；
- 正在运行的程序的一个实例；
- 可以分配给处理器并由处理器执行的一个实体；
- 是可以和别的计算并发执行的计算；
- 是程序在一个数据集合上的运行过程，是系统进行资源分配和调度的一个独立单位。

程序	进程
静态的实体，指令的集合	动态的实体，有生命期
无并发性	进程具有并发性,能与其它进程并发运行
没有建立进程的程序不能运行	进程具有独立性，是一个能独立运行的基本单位

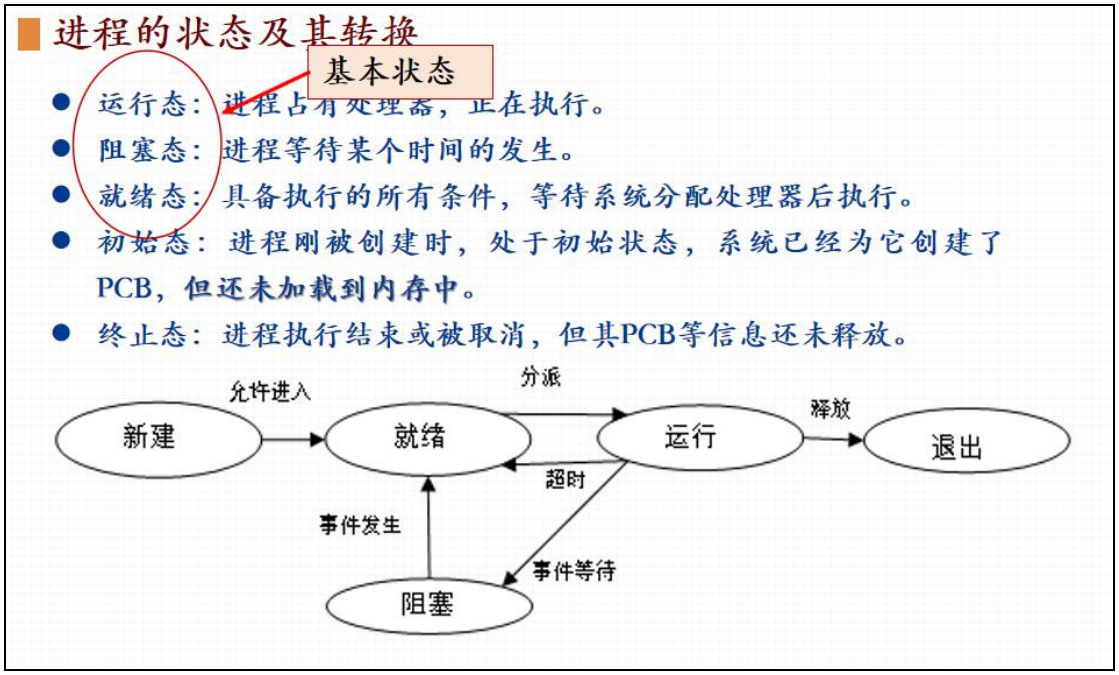
进程和程序的区别和联系:

1. 进程是一个动态的实体，有生命期；程序只是一个静态的实体，只是一组指令的集合
2. 进程具有并发性；程序不能并发执行
3. 进程具有独立性；没建立进程的程序不能运行

总结: 动态性，并发性，独立性



PCB 是 OS 感知进程的唯一标识



就绪态可以由多个状态转化而来

进程的执行模式

处理器的执行模式分为系统态和用户态

- 系统态：又叫控制态，或内核态，具有对处理器以及所有指令、寄存器和内存的控制能力。
- 用户态：只能执行规定的指令，访问指定的寄存器和存储区。

系统进程运行在系统态下，用户进程运行在用户态，不能执行操作系统的指令和访问操作系统区域。因此，可以**保护操作系统不受用户程序的破坏。**

> 程序状态字中有一位表示处理器的执行模式，
> 通过这一位的改变进行执行模式的设置。

进程切换时的动作：

(1) 保存处理器上下文环境。
(2) 更新当前运行进程的PCB。
(3) 将进程控制块移入相应队列（阻塞队列或就绪队列）。
(4) 选择另一进程运行。这部分内容涉及进程调度，在第6章中详细介绍。
(5) 更新所选择进程的PCB。如，状态由就绪变为运行。
(6) 恢复被选择进程的处理器上下文。

进程的创建：进程借助创建原语实现创建一个新进程

引起进程创建的原因有以下几种：

- 操作系统准备接纳一个新的批处理作业时，由作业调度程序为用户作业创建相应的新进程。
- 终端用户登录到系统时会创建进程。
- 操作系统为用户创建一个服务进程。如，用户请求打印一个文件，操作系统可以创建一个打印进程来完成。
- 现有进程可以派生其子进程，并与其并行执行任务。

创建原语需要完成以下步骤：

(1) 给新进程分配一个唯一的进程标识符，并申请一个空白的PCB(PCB是有限的)。若PCB申请失败则创建失败。

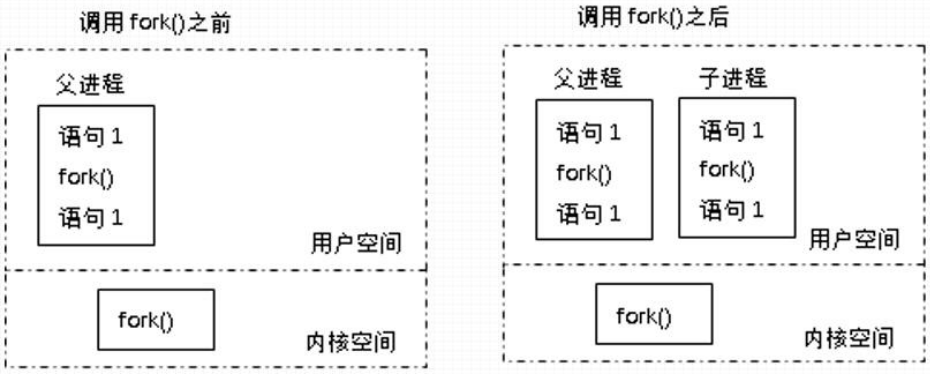
(2) 给进程分配空间分配包括程序、数据、用户栈等。

(3) 将新进程插入就绪队列和进程隶属关系族群中。

(4) 创建或扩充其它数据结构，如为进程创建记账文件等。

fork()函数功能：

该函数有两个返回值：子进程中返回0，父进程中返回进程id，出错时返回-1。



线程又被称为轻型进程，在引入了线程的系统中，线程是一个可以独立执行和调度的基本单位，但不再是拥有资源的独立单位。他只拥有少量运行中必不可少的资源（如程序计数器、一组寄存器和栈），它可与同属一个进程的线程共享该进程拥有的所有资源。

传统的进程等效于只有一个线程的进程

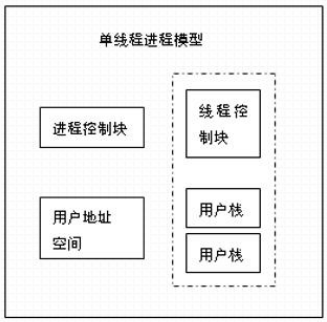


图3-8 单线程进程模型

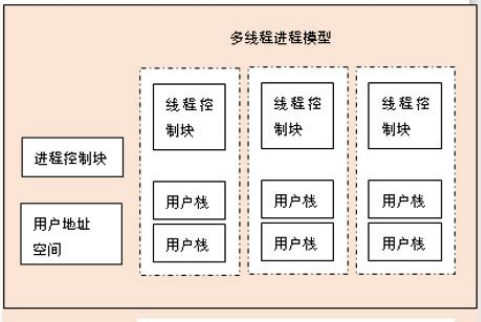


图3-7 多线程进程模型

进程和线程的区别和联系:

1. 线程是调度和分派的基本单位；进程是资源分配的基本单位；同一进程的线程切换不会引起进程切换
2. 进程之间可以并发执行，一个进程的多个线程之间也可以并发执行，因此可以更有效地利用系统资源，并发性更好
3. 进程是拥有资源的独立单位，线程只拥有少量的资源，但可以访问其进程的资源；因此，线程创建，终止，切换 的系统开销比进程小
4. 同一进程的线程共享内存和文件，在同一地址空间里，进程之间的通信无需调用内核，提高了通信效率

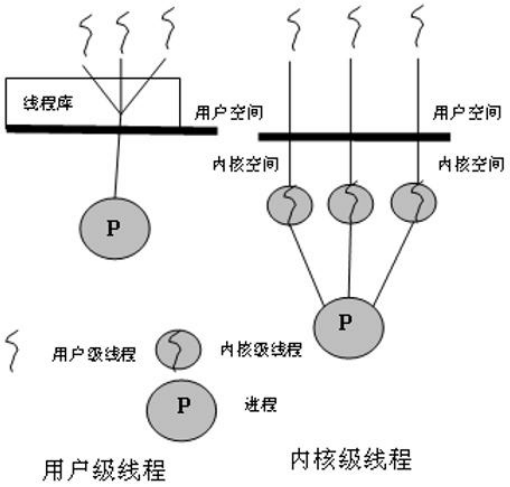
根据实现方式的不同，可以把线程分为两类：

1.内核级线程

线程管理的所有工作都是由内核完成的。用户通过操作系统给应用程序提供的应用程序编程接口API来进行进程管理。

2.用户级线程

线程管理的所有工作都是由应用程序完成的，内核意识不到线程的存在，内核以进程为单位进行调度。操作系统提供给用户一个线程库对线程进行操作。



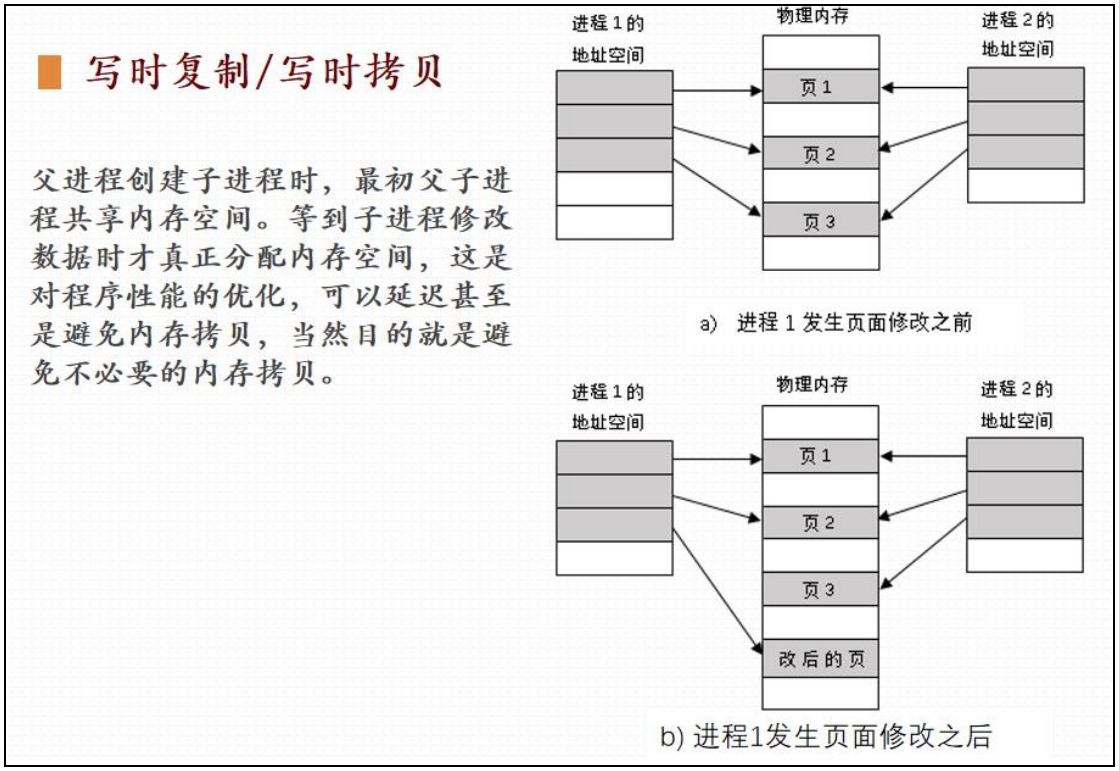
内核级线程 内核可以感知;
用户级线程 用户可以调度

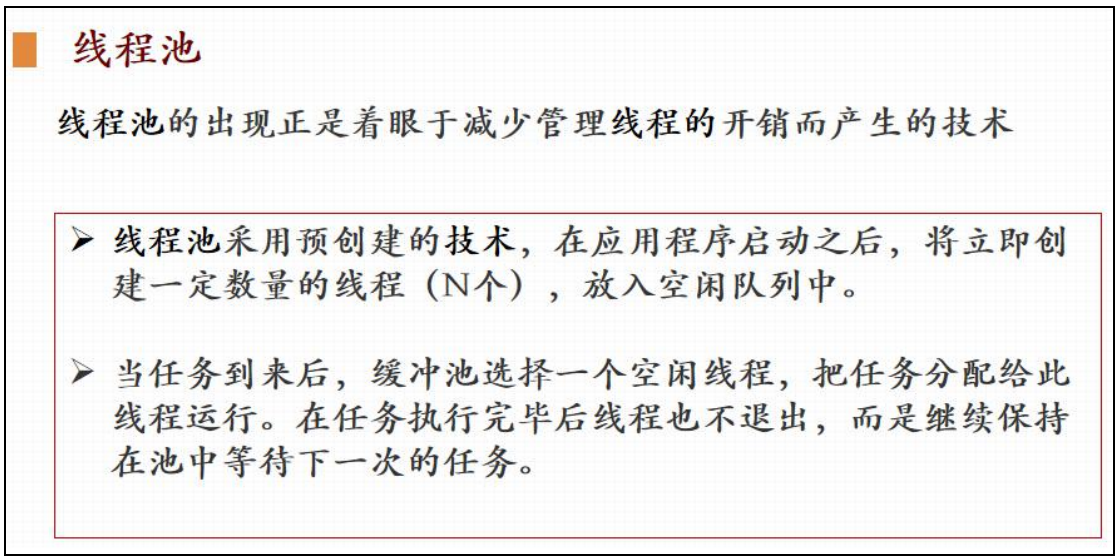
	用户级线程	内核级线程
优点	☐ 线程切换不需要内核参与，开销小	☐ 多个线程调度到不同的处理器上并行执行
	☐ 调度算法由用户自行选择	☐ 内核本身可多线程执行，提升系统并行性
	☐ 可以在任何操作系统中运行	
缺点	调用一个系统调用而阻塞时，其所属的进程中的所有线程都会被阻塞；无法调度到多个处理器上执行	线程切换需要内核参与，开销大

线程库: POSIX Pthread. / Windows API. / Java

写时拷贝/写时复制

这都是针对进程而言的
父/子　页面共享
父/子 any 写入时，创建共享页面的副本；未被修改的页面仍然共享
只有可以修改的页面才标记为 写时复制，不能修改的页面可以由父子共享





进程间远程通信(IPC)
基于 C/S
3ways：套接字 / 远程过程调用 RPC / 远程方法调用 RMI