# 进程控制学习笔记

## 进程标识

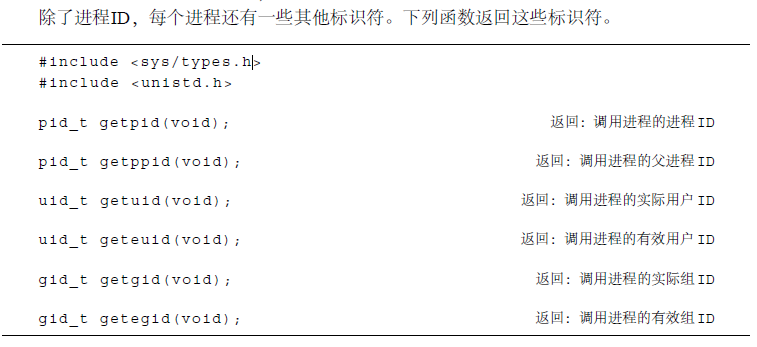
每个进程都有一个非负整型的唯一进程I D。因为进程I D标识符总是唯一的，常将其用做其他标识符的一部分以保证其唯一性。

有某些专用的进程：

进程ID 0是调度进程，常常被称为交换进程( s w a p p e r )。该进程并不执行任何磁盘上的程序—它是内核的一部分，因此也被称为系统进程。

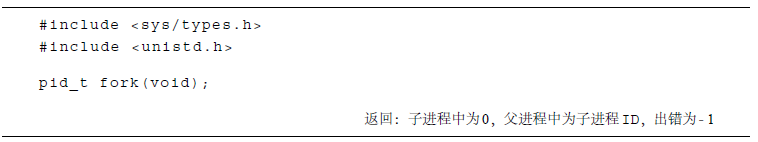
进程ID 1通常是i n i t进程，在自举过程结束时由内核调用。i n i t进程决不会终止。它是一个普通的用户进程(与交换进程不同，它不是内核中的系统进程)，但是它以超级用户特权运行。本章稍后部分会说明i n i t如何成为所有孤儿进程的父进程。

在某些U N I X的虚存实现中，进程ID 2是页精灵进程( p a g e d a e m o n )。此进程负责支持虚存系统的请页操作。与交换进程一样，页精灵进程也是内核进程。



## fock函数

一个现存进程调用f o r k函数是U N I X内核创建一个新进程的唯一方法(这并不适用于前节提及的交换进程、i n i t进程和页精灵进程。这些进程是由内核作为自举过程的一部分以特殊方式创建的)。



由f o r k创建的新进程被称为子进程（ child process）。该函数被调用一次，但返回两次。两次返回的区别是子进程的返回值是0，而父进程的返回值则是新子进程的进程I D。

子进程和父进程继续执行f o r k之后的指令。子进程是父进程的复制品。例如，子进程获得

父进程数据空间、堆和栈的复制品。注意，这是子进程所拥有的拷贝。父、子进程并不共享这些存储空间部分现在很多的实现并不做一个父进程数据段和堆的完全拷贝，因为在f o r k之后经常跟随着e x e c。作为替代，使用了在写时复制( C o p y - O n - Write, COW)的技术