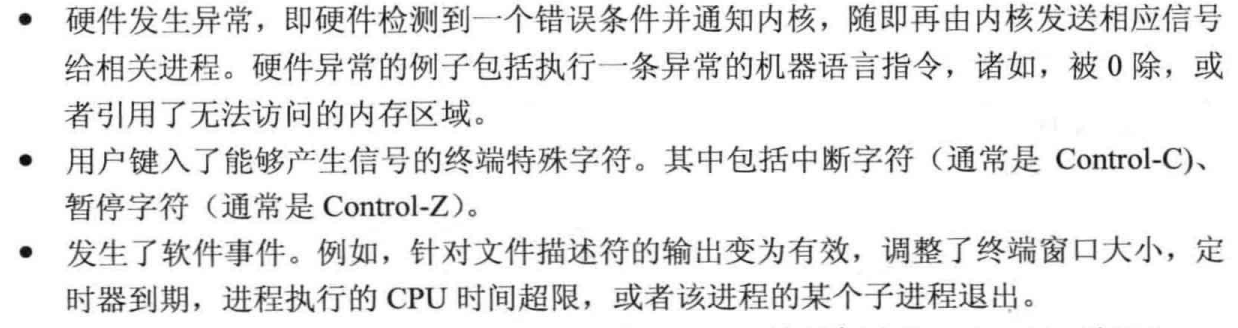
第二十章

1. 概念和概述：

信号是事件发生时对进程的通知机制，也称为软件中断。根据我的了解，硬件中断是CPU去检查CPU引脚查看是否有异常信号，而信号则是在进程的pending结构里设置信号位检查发现。

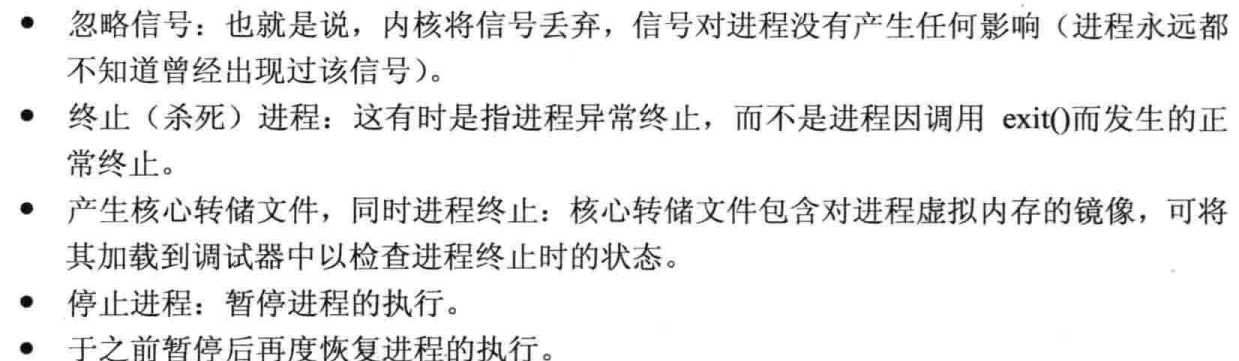
要了解一些细节：比如多个相同的信号接收，pending数据结构会仅仅保存一个还是多个，阻塞信号呢？

引起内核为进程产生信号的各类事件如下：

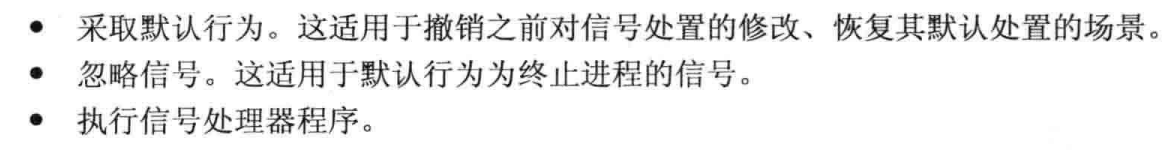


注意以后了解实时信号和标准信号，可靠信号与不可靠信号的区别。

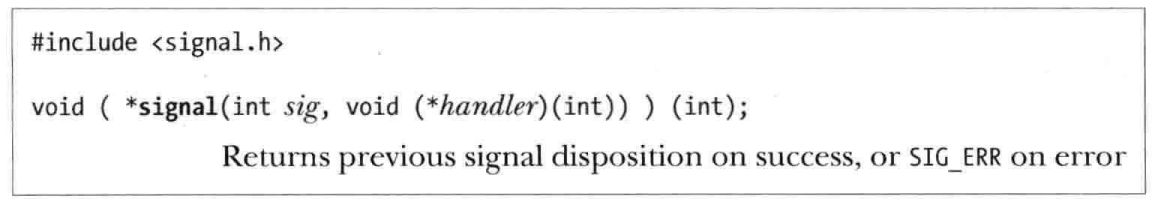
信号到达后，默认操作如下：



程序对信号的处置：

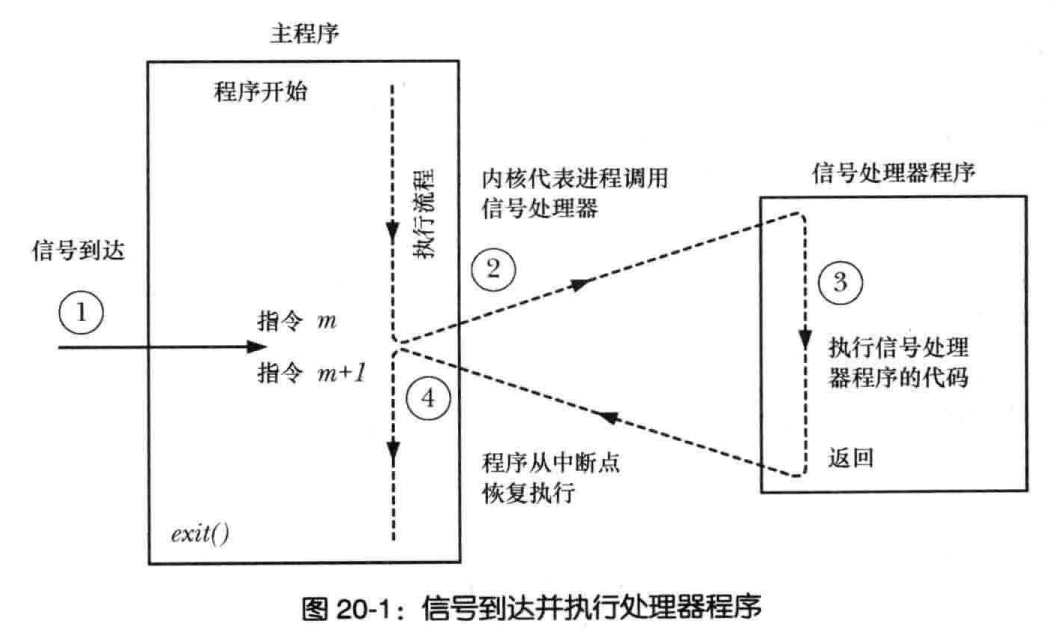


1. 信号的类型和默认行为：介绍了信号的类型，信号值，信号的默认行为，以及什么情况下会出现该信号。具体查看7 signal文档。
2. 改变信号处置：

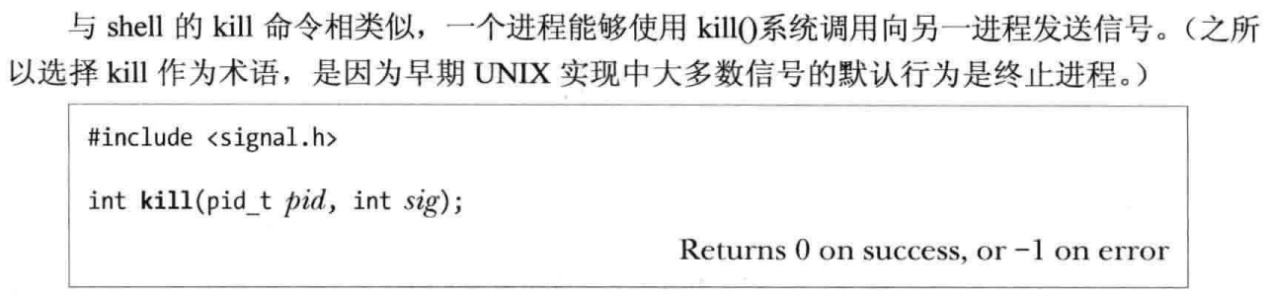


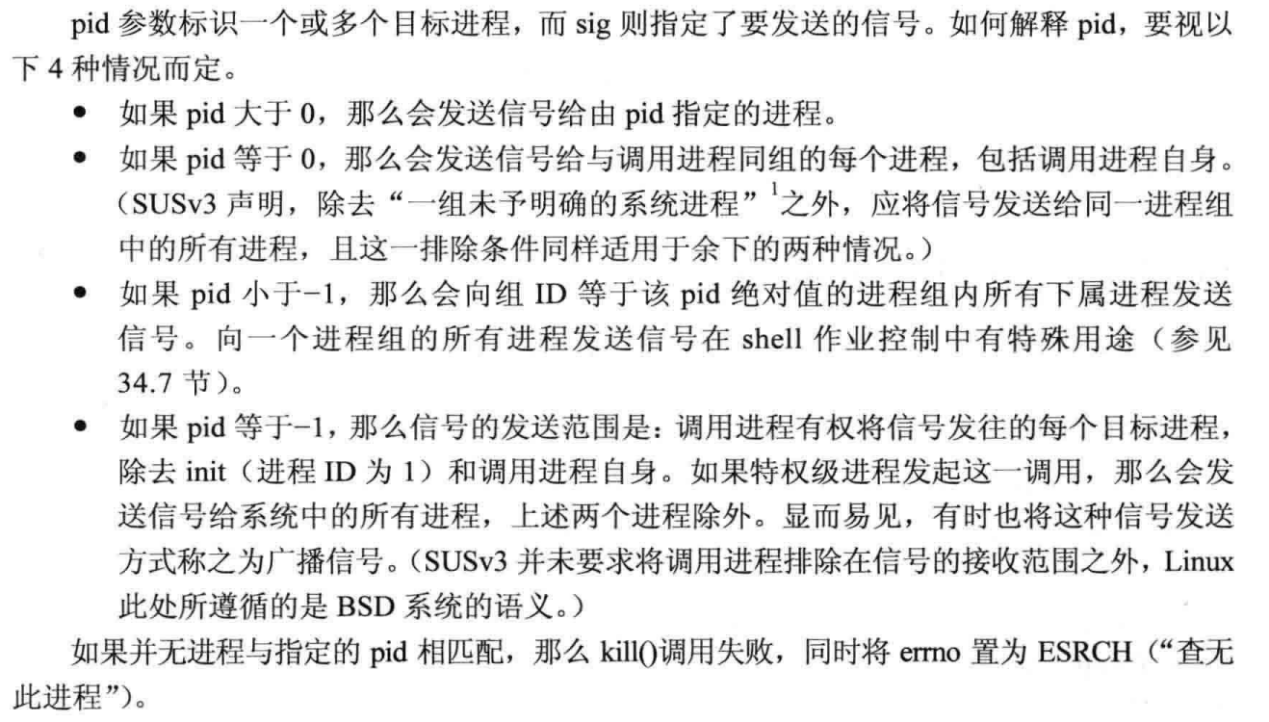
注意，这个函数不兼容，一般不使用。

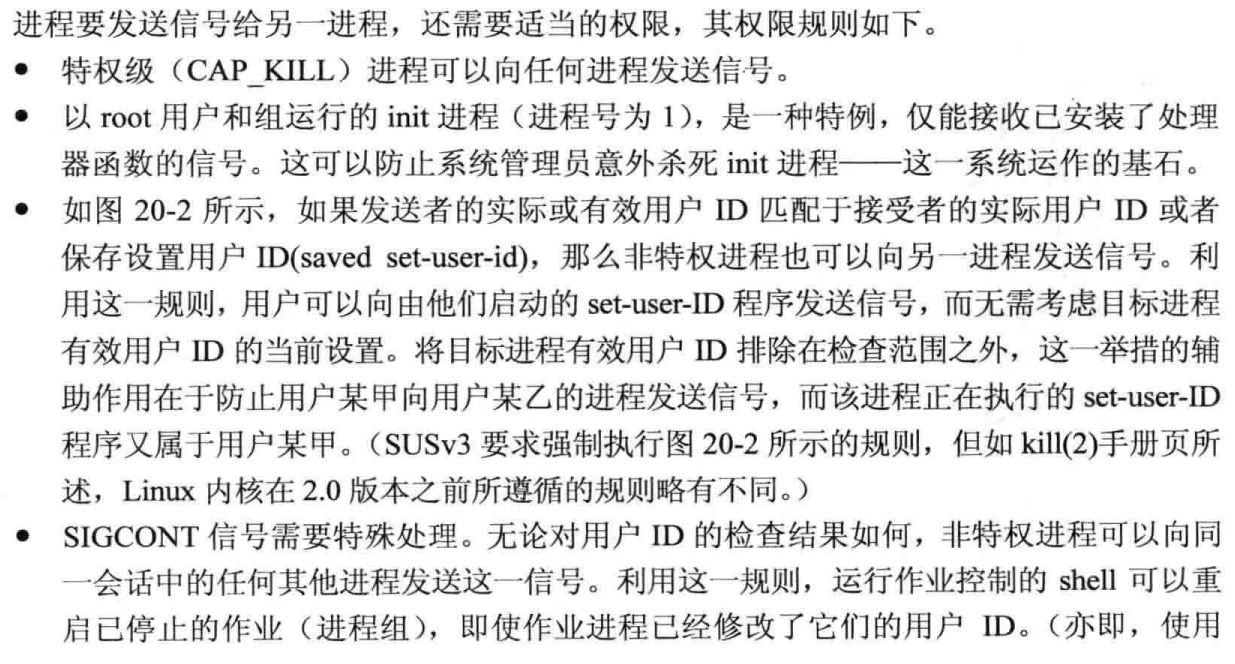
1. 信号处理器简介：

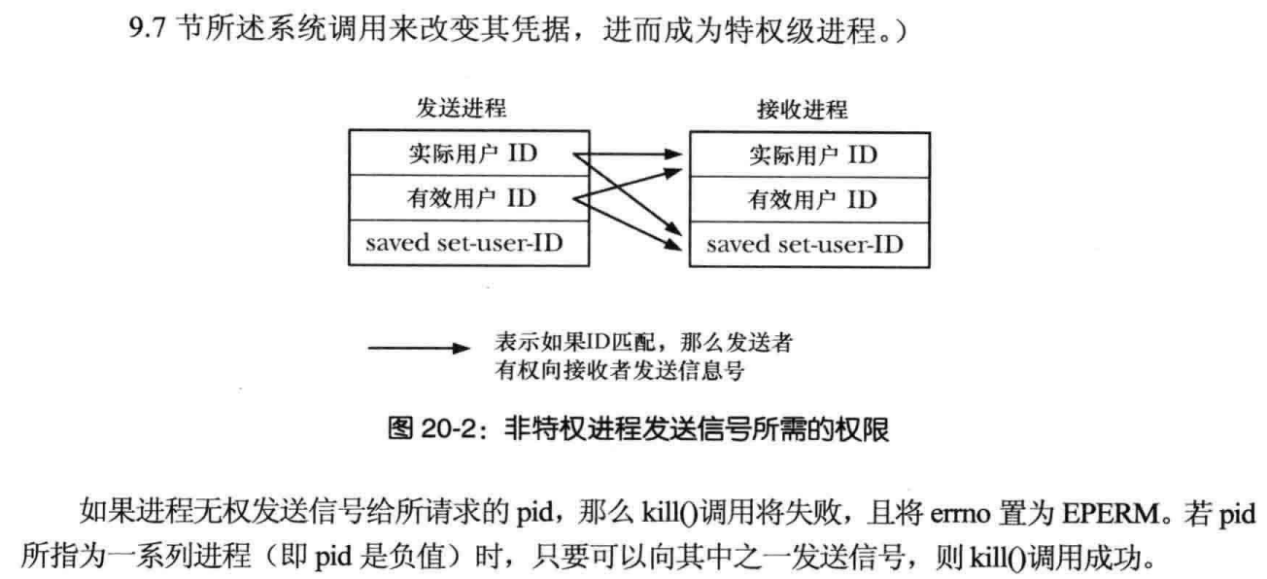


1. 发送信号：







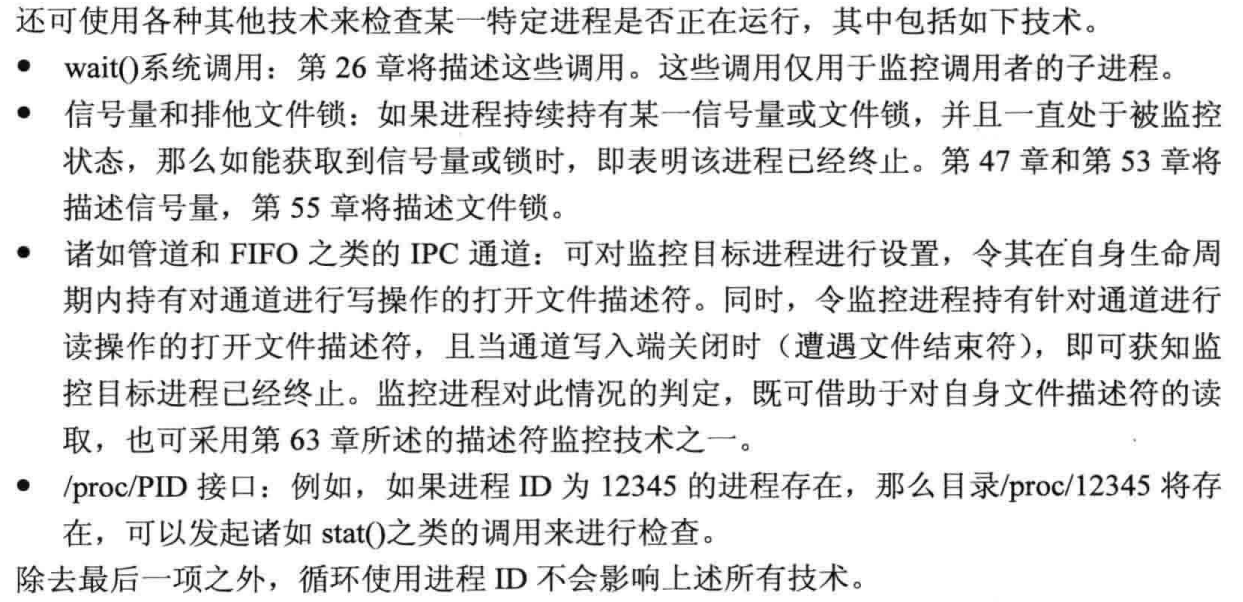


注意，如果发送给一个组，则只要发送其中部分信号成功，那么调用就会成功。

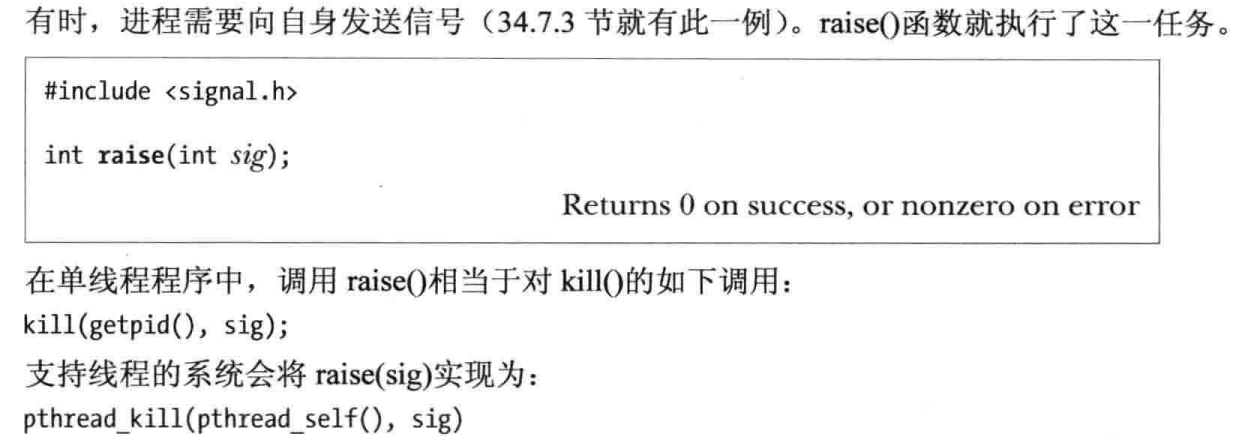
1. 使用kill系统调用检查进程是否存在：

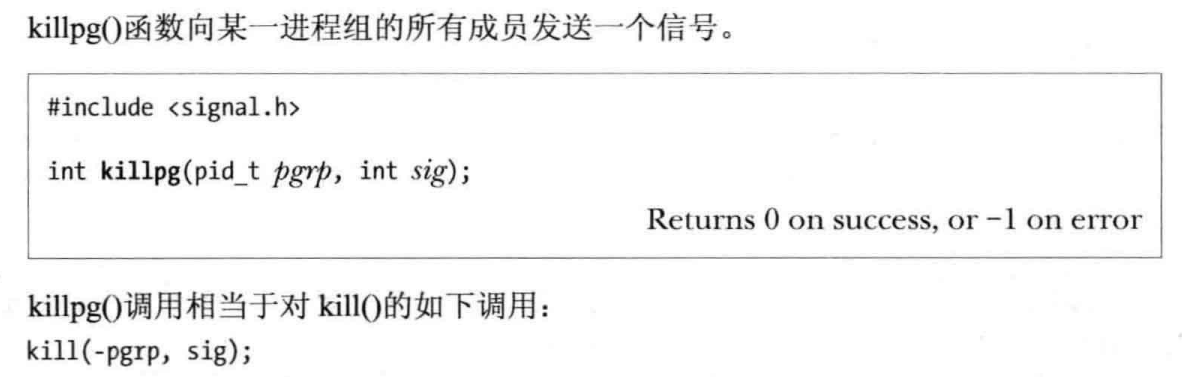
（指定sig参数为0，判断返回值，但要注意无法判断是否为僵尸进程或者是否仍然是那个进程，因为PID是会循环改变的。）

检查进程是否存在的其他方法：

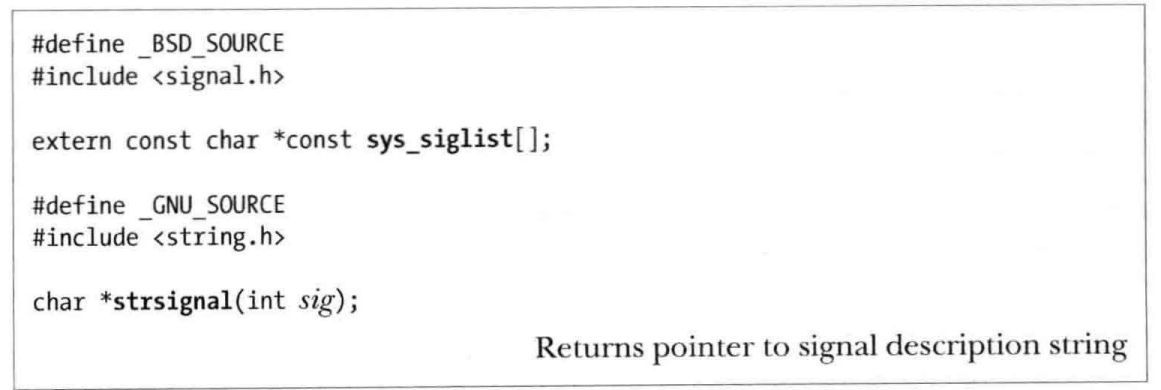


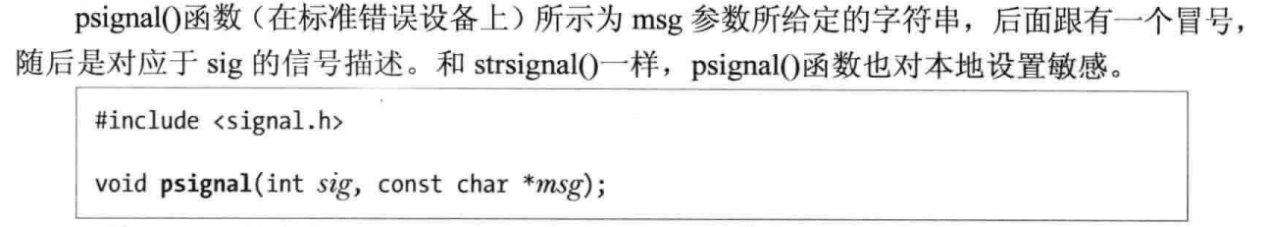
1. 发送信号的其他方式：





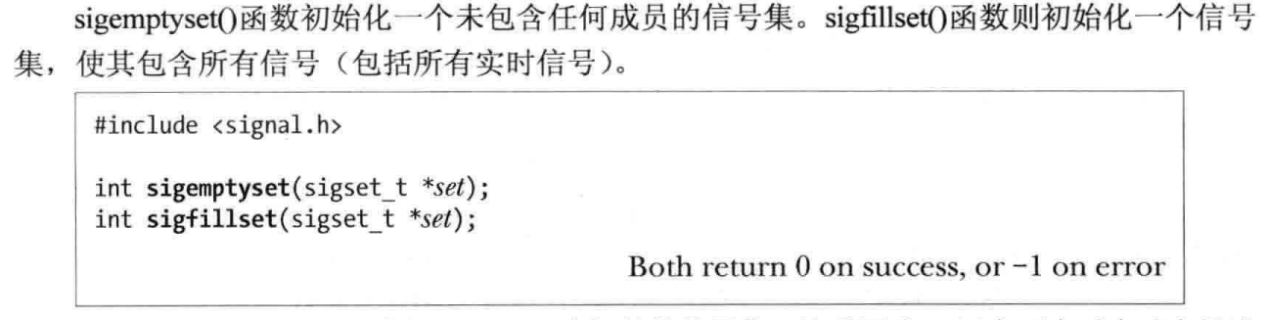
1. 显示信号描述

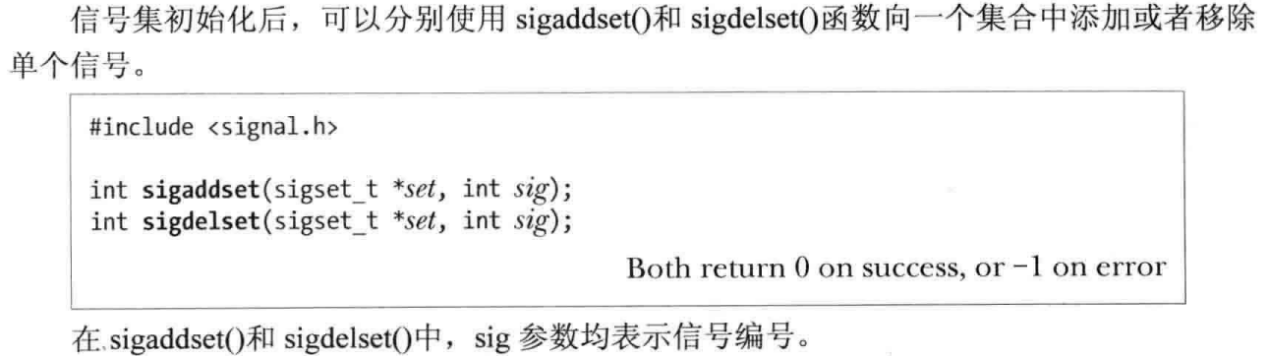


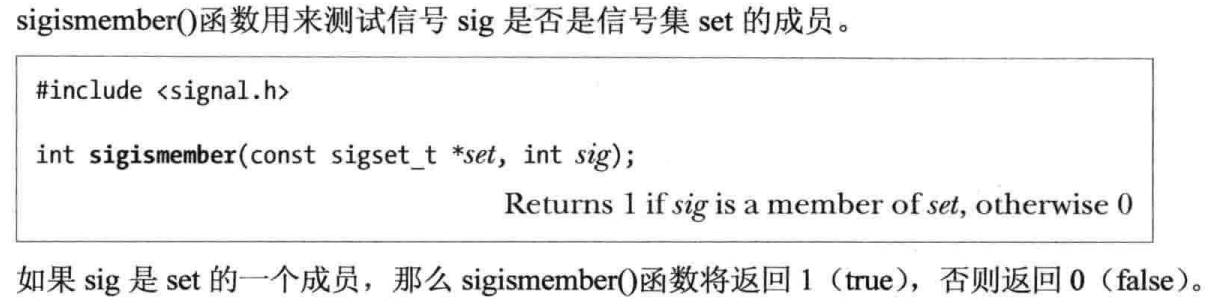


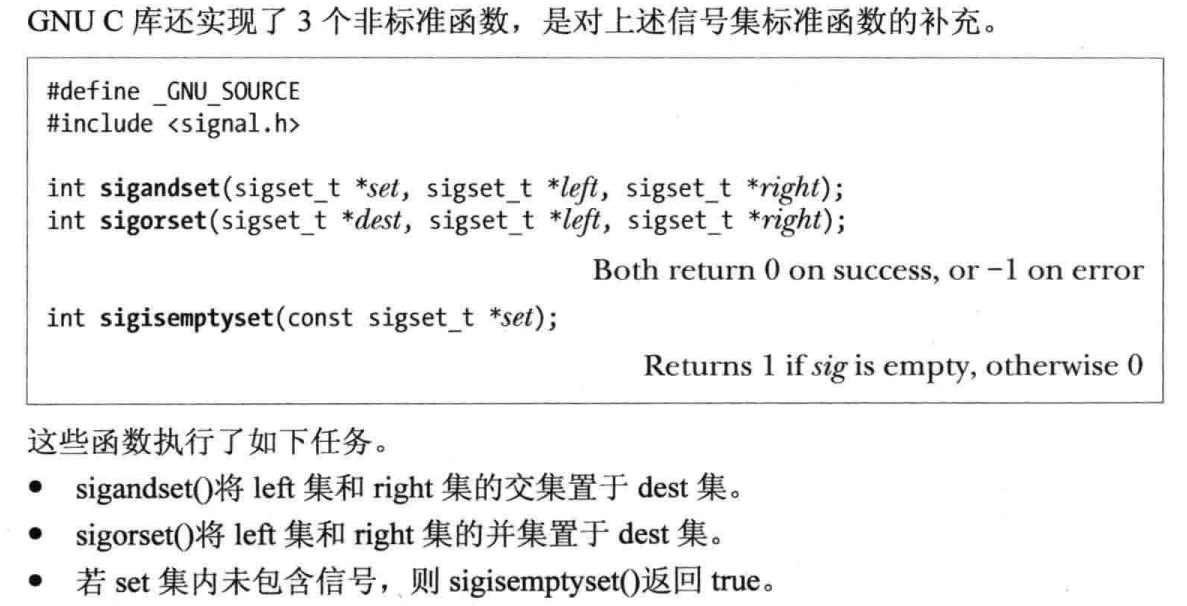
1. 信号集：

初始化信号集：



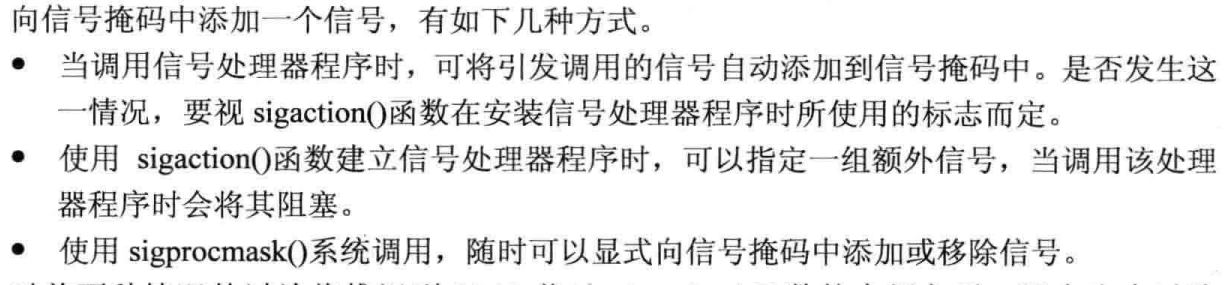


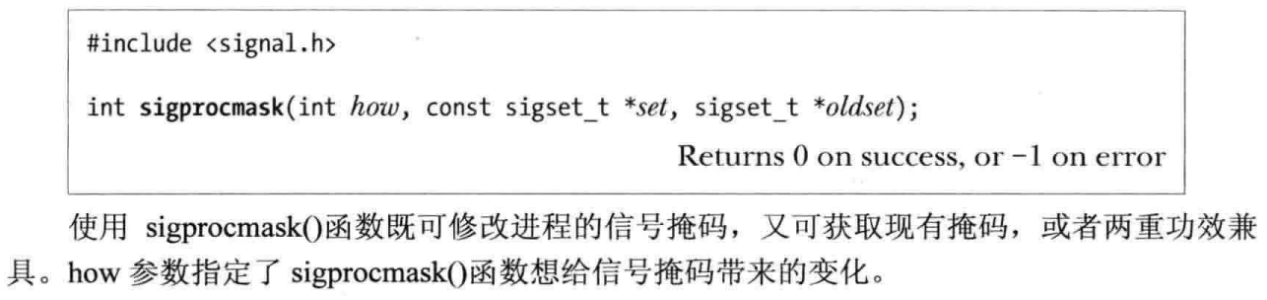




1. 信号掩码（阻塞信号传递）：

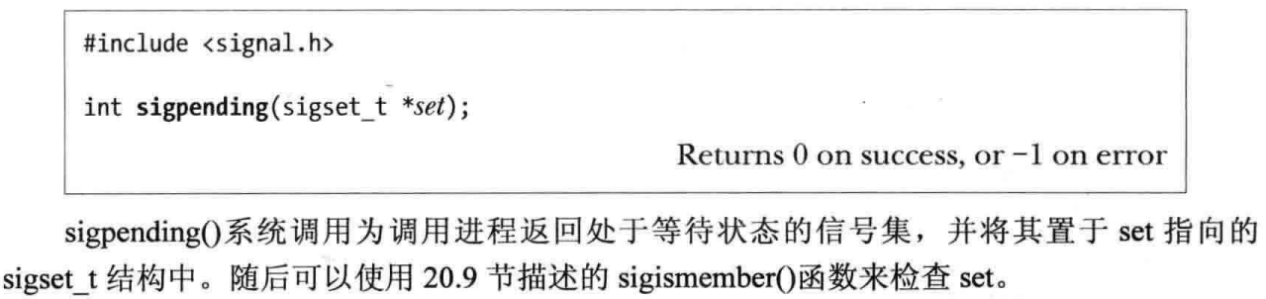
注意，信号掩码属于线程属性，向信号掩码添加信号如下情况：





注意，系统将忽略阻塞SIGKILL和SIGSTOP信号的的请求。

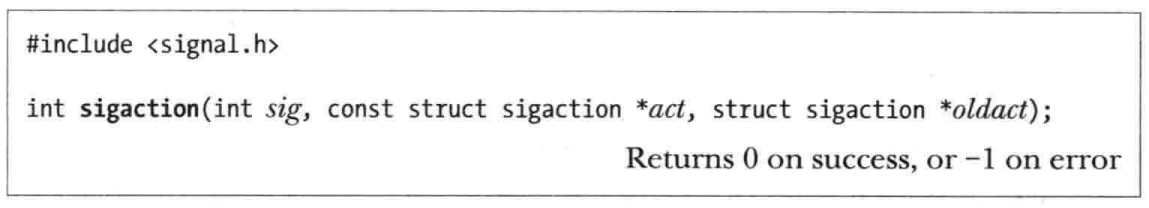
1. 处于等待状态的信号：



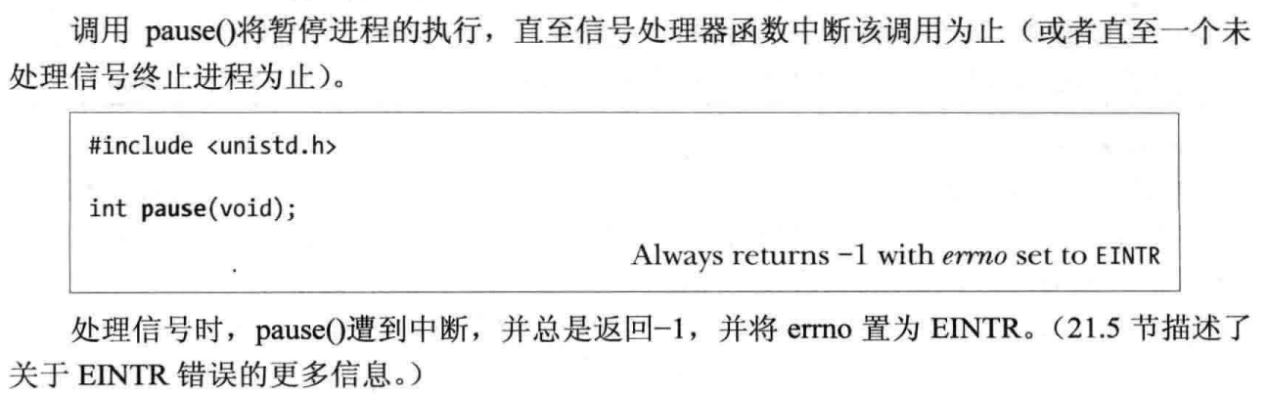
1. 不对信号进行排队处理

通过一个程序说明标准信号不排队，也就是说，阻塞情况下，多个信号来临的时候，标准信号只传递一次。

1. 更好的改变信号处置：



1. 等待信号：使用这个比一直循环更好利用CPU。



遇到一个问题，20-7的程序有个问题：被阻塞的函数没有办法执行信号函数。