

- 接收信号。当目的进程被内核强迫以某种方式对信号的发送做出反应时，它就接收了信号。进程可以忽略这个信号，终止或者通过执行一个称为信号处理程序(signal handler)的用户层函数捕获这个信号。图 8-27 给出了信号处理程序捕获信号的基本思想。

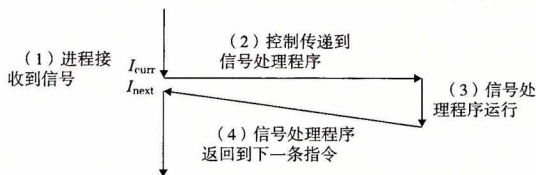


图 8-27 信号处理。接收到信号会触发控制转移到信号处理程序。在信号处理程序完成处理之后，它将控制返回给被中断的程序

一个发出而没有被接收的信号叫做待处理信号(pending signal)。在任何时刻，一种类型至多只会会有一个待处理信号。如果一个进程有一个类型为 k 的待处理信号，那么任何接下来发送到这个进程的类型为 k 的信号都不会排队等待；它们只是被简单地丢弃。一个进程可以有选择性地阻塞接收某种信号。当一种信号被阻塞时，它仍可以被发送，但是产生的待处理信号不会被接收，直到进程取消对这种信号的阻塞。

一个待处理信号最多只能被接收一次。内核为每个进程在 pending 位向量中维护着待处理信号的集合，而在 blocked 位向量[⊖]中维护着被阻塞的信号集合。只要传送了一个类型为 k 的信号，内核就会设置 pending 中的第 k 位，而只要接收了一个类型为 k 的信号，内核就会清除 pending 中的第 k 位。

8.5.2 发送信号

Unix 系统提供了大量向进程发送信号的机制。所有这些机制都是基于进程组(process group)这个概念的。

1. 进程组

每个进程都只属于一个进程组，进程组是由一个正整数进程组 ID 来标识的。getpgrp 函数返回当前进程的进程组 ID：

```
#include <unistd.h>

pid_t getpgrp(void);
```

返回：调用进程的进程组 ID。

默认地，一个子进程和它的父进程同属于一个进程组。一个进程可以通过使用 setpgid 函数来改变自己或者其他进程的进程组：

```
#include <unistd.h>

int setpgid(pid_t pid, pid_t pgid);
```

返回：若成功则为 0，若错误则为 -1。

setpgid 函数将进程 pid 的进程组改为 pgid。如果 pid 是 0，那么就使用当前进程

[⊖] 也称为信号掩码(signal mask)。