516

№ 练习题 8.2 考虑下面的程序:

```
code/ecf/forkprob0.c
    int main()
1
2
    {
3
        int x = 1;
4
        if (Fork() == 0)
5
             printf("p1: x=%d\n", ++x);
6
7
        printf("p2: x=%d\n", --x);
        exit(0);
    }
                                                          - code/ecf/forkprob0.c
```

- A. 子进程的输出是什么?
- B. 父进程的输出是什么?

8.4.3 回收子进程

当一个进程由于某种原因终止时,内核并不是立即把它从系统中清除。相反,进程被保持在一种已终止的状态中,直到被它的父进程回收(reaped)。当父进程回收已终止的子进程时,内核将子进程的退出状态传递给父进程,然后抛弃已终止的进程,从此时开始,该进程就不存在了。一个终止了但还未被回收的进程称为侵死进程(zombie)。

旁注 为什么已终止的子进程被称为僵死进程?

在民间传说中, 僵尸是活着的尸体, 一种半生半死的实体。僵死进程已经终止了, 而内核仍保留着它的某些状态直到父进程回收它为止, 从这个意义上说它们是类似的。

如果一个父进程终止了,内核会安排 init 进程成为它的孤儿进程的养父。init 进程的 PID 为 1,是在系统启动时由内核创建的,它不会终止,是所有进程的祖先。如果父进程没有回收它的僵死子进程就终止了,那么内核会安排 init 进程去回收它们。不过,长时间运行的程序,比如 shell 或者服务器,总是应该回收它们的僵死子进程。即使僵死子进程没有运行,它们仍然消耗系统的内存资源。

一个进程可以通过调用 waitpid 函数来等待它的子进程终止或者停止。

waitpid 函数有点复杂。默认情况下(当 options=0 时), waitpid 挂起调用进程的执行,直到它的等待集合(wait set)中的一个子进程终止。如果等待集合中的一个进程在刚调用的时刻就已经终止了,那么 waitpid 就立即返回。在这两种情况中,waitpid返回导致 waitpid 返回的已终止子进程的 PID。此时,已终止的子进程已经被回收,内核会从系统中删除掉它的所有痕迹。

1. 判定等待集合的成员

等待集合的成员是由参数 pid 来确定的: