

系统编程12周

18373206 陈泓瑞

1

进程：

1. 抽象出“进程”这一概念，可以更好的控制和协调程序的执行。
2. 进程有自己的虚拟地址空间，这实现了隔离。操作系统也可以对每个进程的内存进行管理、回收等操作。
3. 进程有自己的PID、SID、UID、RUID等，linux对文件或者系统调用的权限控制就是基于进程的这些id
4. CPU的调度（分时），也是以进程为单位的。
5. 可以方便的对程序进行管理：终止进程、向进程/进程组发送信号等。

2

代码段A：最后会存在 $2^5=32$ 个进程。计算：可以画一个5层的二叉树，叶子节点就是最终的进程。

代码段B：最后会存在6个子进程。计算：每次fork，产生的子进程不会执行fork。所以只会执行5次fork，共 $5+1=6$ 个进程。

3

fork后，应当用wait或者waitpid来等待子进程返回并回收资源。

僵尸进程：已经结束、但是未被回收的子进程。可能父进程没来得及调用wait，也可能父进程根本没打算wait，也可能父进程先结束了。这些情况结束的子进程都是僵尸进程。

调用wait函数，系统会将调用者阻塞，直到检测到任意一个子进程结束已成为僵尸进程，然后会回收资源，最后返回（修改*status）子进程的结束状态。

4

信号是一种通信机制，是一种异步消息。

程序在接受到信号时，会立即处理。这类似于汇编中的中断。同样的，还可以自定义信号处理例程。在接收到信号时，这些处理例程会被自动立即调用。

信号可以用于杀掉进程（SIGTERM，SIGKILL），传递一些消息（SIGCHLD 子进程终止），也可以传递一些自定义的消息（SIGUSR1，SIGUSR2）。

5

如果某种信号被屏蔽，那么该内核会标记该信号处于未决态。在解除屏蔽前，内核不会再向该程序发送此信号。

信号未决的作用：通过屏蔽信号，避免读写冲突的发生，保证程序的可靠性。

6

设信号量S，初始值为1.

每个读写者，都应当按照：P(S) - 进行读写操作 - V(S) 来进行操作。这样就可以保证隔离性。