

18373773 马壮

1

进程用途：是系统进行资源分配和调度的单位，用来描述传统意义上的程序不足以描述的 OS 中各种活动之间的动态性、并发性、独立性还有相互制约性。

2

代码段A进程数32：每次fork()函数后，父子进程均进入第二次循环，每次循环后进程数翻倍

代码段B进程数6：在循环中添加了判断，子进程直接跳出，故每次循环进程数增1

3

一个进程使用fork创建子进程，如果子进程退出，而父进程并没有调用wait或waitpid获取子进程的状态信息，那么子进程的进程描述符仍然保存在系统中。这种进程称之为僵尸进程。

进程调用wait后，就立即阻塞自己，由wait自动分析是否当前进程的某个子进程已经退出，如果让它找到了这样一个已经变成僵尸的子进程，wait就会收集这个子进程的信息，并把它彻底销毁后返回；如果没有找到这样一个子进程，wait就会一直阻塞在这里，直到有一个出现为止。

4

信号是Linux进程间通信的一种机制，本质上是在软件层次上对硬件中断机制的一种模拟。与其他进程间通信方式（例如管道、共享内存等）相比，信号所能传递的信息比较粗糙，只是一个整数。但正是由于传递的信息量少，信号也便于管理和使用，可以用于系统管理相关的任务，例如通知进程终结、中止或者恢复等。

5

信号的未决状态指从信号的产生到信号被处理前的这段时间，未决状态信号的产生主要是因为进程对此信号的阻塞。

信号存在未决状态作用：解除阻塞后，处于“未决状态”的信号能继续被处理，不会造成信号丢失。

6

假设要实现的共享内存为：一个进程对其更新，另一个进程从中读出数据。

通过信号量实现思路如下：

1.服务端首先建立一块共享内存的映射，然后创建一个信号量，信号量数目为1。通过信号量获得对共享资源的使用权限，更新内存中的内容，等待客户端进程读取共享资源中的数据后，释放共享内存和信号量，最后退出。

2.客户端获得共享内存的映射以及信号量的映射，通过信号量获得对共享资源的访问权限，然后读取其内容，接着解除与共享内存的映射后退出。客户端每次读取共享内存数据时首先调用semop（将sem_op置为0）检测信号量的值，当信号量值为0时才能读取共享内存数据。