1.进程是描述程序执行过程和资源共享的基本单位，Linux提供了一系列进程相关函数，可用于控制和协调程序的执行。

2.代码段A返回=32个pid，代码段B返回6个pid。其原因是，对于代码段A，其父进程和子进程都会执行fork(); 代码段B只有最初的父进程执行fork()产生子进程。

3.进程调用了exit()函数后，如果没有被父进程回收，则该进程成为僵尸进程。僵尸进程只在进程列表中保留一个位置，记载进程的退出状态等信息供父进程收集。

进程调用wait()函数后被挂起，进入阻塞状态，直到子进程变为僵尸态，wait()函数捕获到该子进程的退出信息时才会转为运行态，回收子进程资源并返回；如果没有变为僵尸态的子进程，wait()函数会让进程一直阻塞。

4.信号全程软中断信号，本质是软件层次上对中断机制的一种模拟，用于提醒进程某件事情已经发生。信号是一种重要的进程通信机制。

5.当信号掩码（signal mask）中某个信号对应的位被设置为1，信号会被屏蔽，进入阻塞状态；此时内核修改未决信号集(signal pending)中该信号对应的位为1，使该信号处于未决状态。

信号存在未决状态的作用是解决常规信号不可靠这一问题，避免由于在处理信号的同时再次收到信号导致的该信号被丢弃。

6.一个进程读，一个进程写。

写进程：首先打开输入文件，创建并初始化共享内存和信号量（指示读写资源的信号量，能否读初始化为0，能否写初始化为1），读取一帧，对写信号量执行P操作，写入共享内存，释放共享内存权限，对读信号量执行V操作。处理完所有帧后，设置结束标志，释放相关资源。

读进程：打开输出文件，获得共享内存和信号量，对读信号量执行P操作，读取一帧，并获取结束标志，处理后释放共享内存权限，对写信号量执行V操作，知道结束标志为真。最后释放相关资源。