1. 进程是用来描述程序执行过程和资源共享的基本单位。有了进程，就可以控制和协调程序的执行。
2. A代码段的每个子进程还会创建新的子进程，每次循环后的总进程数都会\*2，最后共有32个进程。

B代码段在循环中进行了判断，如果是子进程则跳出循环，只有父进程始终在循环中，每个循环中创建一个子进程，最后共有6个进程。

1. 若子进程已经结束，但父进程并没有调用wait或waitpid函数回收子进程的话，结束的子进程就会变成僵尸进程。

父进程调用wait()函数后会被挂起，进入阻塞状态，直到子进程结束变为僵尸态，wait()函数捕获到该子进程的退出信息时，父进程转为运行态，回收子进程资源并返回。

1. 信号是发送给进程的特殊异步消息，进程在收到信号时会立即进行处理。可以通过信号即时改变进程的运行状态，进行进程管理。
2. 当某个信号的掩码被置1时，发送的该信号会被接收进程屏蔽，进入阻塞状态。此时该进程的未决信号集中该信号的相应位会被置1，该信号进入未决状态。

当未决信号信号的屏蔽被解除时，内核会再次向进程发送这个信号。所以未决信号的作用在于，被屏蔽的信号不会直接被忽略，而是等待屏蔽解除时再次发送给进程。

1. 步骤如下
2. 创建共享内存段。
3. 为共享进程段创建一个信号量集，其中有两个元素，分别用来表示共享内存段的读资源和写资源占用情况。
4. 初始化所有信号量的值为1。
5. 将读写进程连接到共享内存段。
6. 若某个进程需要对共享内存进行写操作，此时不能有其他进程正在读写共享内存。所以操作前要对2)中创建的写信号量元素进行P操作（-1）。若此时写信号量为0，表示有进程正在写，那么此进程暂时被阻塞，直到写信号量均为1时解除阻塞。再判断读信号量是否>0，若大于0，说明有进程正在读，那么此进程暂时被阻塞，直到读信号量为0时解除阻塞。写操作结束后对写信号量元素进行V操作（+1），表示此进程已结束写操作。
7. 若某个进程需要对共享内存进行读操作，此时不能有其他进程正在写共享内存，但可以有其他进程同时进行读操作。若写信号量为0，说明有进程正在进行写操作，那么此进程暂时被阻塞，直到写信号量为1。操作前要对2)中创建的读信号量元素进行V操作（+1），用来计算一共有多少个进程在同时进行读操作。读操作结束后对读写信号量元素进行P操作（-1），表示此进程已结束读操作。。