1. 进程是在程序运行时产生的。每个进程都有自己的地址空间，并占用一定的系统资源。程序是“死”的，进程是“活”的。虽然单核CPU同一时刻只能处理一个进程，我们可以运用进程，使宏观上可以表现为计算机可以并发地处理多个进程。
2. 代码段A:最后会有32个进程。代码段B:最后会有6个进程。

原因：代码段A中每次循环父进程会创建一个子进程，而之前创建的子进程也会创建新的子进程，所以每一次循环后进程总数变为原来的2倍。代码段B中如果当前进程是子进程就会跳出循环，所以最后只有5个子进程、1个父进程。

1. 当进程调用exit()函数后，会留下一个僵尸进程，它只在进程列表中有一个位置，记载进程的退出状态等信息供父进程收集。

调用wait()函数的进程会被挂起，进入阻塞状态，直到子进程变为僵尸态之后原进程才会转为运行态。如果当前进程有多个子进程，只要捕获到一个僵尸态的子进程，wait函数就会返回。

1. 信号是实现Linux系统中进程之间通信的方式之一。
2. 发送的信号被阻塞，无法到达进程，内核就会将信号的状态设置为未决状态。

作用：进程可以选择阻塞(Block)某个信号。被阻塞的信号产生时将保持在未决状态，直到进程解除对此信号的阻塞，才执行递达的动作。这个过程信号不会丢失，这样可以使进程之间通信变得更灵活。

1. 使用shmget函数获取共享内存，同时设置读写权限。
2. 创建一个read进程和一个write进程，使用shmat函数链接共享内存。
3. 利用信号量来实现两个线程的同步，如果信号量为1其他进程才能访问资源。例如read进程中当共享内存中有其他进程写入数据时，信号量被设置为0，程序等待。当它不为0时，表示没有进程写入数据，程序就能够从共享内存中读取数据；同样的如果write进程中在写数据，read进程就不能读取数据。这样就使资源同一时刻只能有一个进程去访问它。

4）shmdt函数拆卸贡献内存，最后删除共享内存。