1：在一个操作系统上，可以运行很多程序，同时可以并发、异发。而操作系统就可以给每个在运行的程序一个ID，把这么多程序看作许多个进程，系统来统一的管理这些进程的运行，通过进程管理，系统维护自身的运行正常，杀死一些进程

2：代码段A会产生32个进程，将for循环后的代码执行64遍

代码段B会产生10个进程，将for循环后的代码执行10遍

原因：父进程和子进程fork（）返回的数据不同，而子进程对于本进程fork（）产生的进程来说又是父进程。父进程的fork返回子进程的pid，子进程的fork返回0。所以代码段A产生的每个进程都会进入循环并且fork（）。而代码段B每次产生的子进程都会break出循环，不会继续产生子进程

3：僵尸进程是已经结束了的进程，比父进程先结束，没被父进程回收，内核释放该进程所有的资源,包括打开的文件,占用的内存等。 但是仍然为其保留一定的信息(包括进程号、退出状态、运行时间等)

调用wait（wait(int \*status); ）

(1)父进程阻塞，等待子进程退出

(2)父进程回收子进程残留资源

(3)Status获取子进程结束状态

4:进程和线程之间可以通过信号传递信息，通知发生了异步事件，避免异步产生的问题，也不会导致阻塞。除了传递消息之外，还可以附带参数

5:信号已经产生，还没有被传递到任何一个进程或者是阻塞了或者是来不及处理

作用：进程有多个未决状态信号时，会生成一个信号集，根据重要性来选择处理的优先顺序或者暂不处理。而未决状态的信号被进程处理时，会让进程暂停转去处理信号对应的信号处理函数，处理完后会唤醒原进程。未决信号是实现进程间异步通信的手段

6：利用互斥信号量表示读写资源是否被占有。读的时候可以一起读，不能有人写。读的时候会检查互斥信号量是否有效，有效则表明没有写，并申请一个同步信号量，如果此时还有同步信号量资源，则占有一个资源，读成功。写的时候，其他进程不能写也不能读，则令互斥信号量无效，并且申请所有的同步信号量，如果此时有进程在写，读，则申请不到所有的资源，堵塞，如果申请到所有的同步信号量则表明没有读进程，写成功