1.

进程是描述程序执行过程和资源共享的一个基本单位，用于控制和协调程序的执行。相对于程序而言，程序是“死”的，是一个存放于内存当中的二进制文件，而进程是“活”的，将会占用所需要的系统资源，而且可以使用CPU，受内核控制，进程的状态可以发生改变，包括运行，终止，睡眠等。

2.

代码A总共会产生32个进程（包括父进程），代码B只会产生五个子进程。

因为代码A没有限制子进程break，所以在第一次执行的时候，i=0的时候，子进程复制了一份父进程的代码。然后在i=1的时候，子进程和父进程都会再执行fork()，各自多出一个子进程，子进程再次复制代码，在i=2的时候继续执行fork()，这就使得每次循环都会使进程数量翻倍，五次循环以后进程数量就为2^5=32个。

代码B则是做了个判断，如果是子进程，则退出循环，因此只有父进程会继续执行循环，每次循环产生一个子进程，最后产生了五个子进程。

3.

僵尸进程是进程执行完以后所留下的特殊数据结构，它几乎放弃了所有进程退出前所占用的内存和数据结构，但是会留在进程列表里，等待其父进程收集数据并对它进行回收。如果僵尸进程过多，会占用太多内存和进程id，会导致系统无法创建新的进程。

进程调用wait()函数的时候，会挂起，进入阻塞状态，等待它的子进程变成僵尸态，如果子进程一直没有变成僵尸态，则父进程将一直阻塞，直到其中某个子进程变成僵尸态。这时候wait()函数捕捉到子进程的退出状态，转为运行态，回收子进程资源并返回，之后父进程才会恢复执行。

4.

信号的本质是软件层次上对中断机制的模仿，用于进程间的通信，提醒进程某件事情已经发生。进程接收到信号以后，将执行某些操作。

5.

信号产生以后，如果发送被阻塞，无法到达进程，内核将会将这条信号的状态设置为未决状态。

未决状态其实是用来防止已发送信号被抛弃，被忽略的一种机制。如果进程发送某个信号过于密集，接收信号的进程可能处理不过来，以至于信号被抛弃，这个过程就可能产生信息丢失。未决状态结合信号阻塞机制，则可以解决这个问题，这也是未决状态的作用。

6.

用有名信号量来进行同步。利用两个有名信号量，其中一个信号量控制能否读，初始化为0，另一个信号量控制能否写，初始为1。在进行读或者写操作之前时，要先检查信号量，判断是否能够执行将要执行的操作，如果可以，则执行操作，并且把信号量设置为相应的状态，以免其他进程干扰（比如说读的时候，要设置其他进程不能写），执行完以后，再将信号量恢复，以便于其他进程使用共享内存。