

# 1.

进程描述的是一个二进制程序的执行过程，使用进程的概念可以方便计算机进行多线程的运算，内核需要根据进程的数量和特征以分配CPU的使用权。

# 2.

代码A中会产生32个进程，代码B会产生6个进程

在A中每次循环都会使得进程数量翻倍，总共产生  $2^5 = 32$  个进程，在B中在子进程中fork()函数返回0，因此子进程不会继续循环，只会产生6个进程。

# 3.

如果在父进程未结束的时候子进程已经结束，而且父进程没有回收子进程，那么子进程就会变成僵尸进程。

进程调用wait()函数时会被挂起，直到检测到一个僵尸子进程并回收子进程，若执行成功则返回子进程的id，若执行失败则返回-1。wait()函数可以传入一个int\*类型的参数，用来存放子进程退出时的信息。

# 4.

信号用于告诉进程发生的事情，可以在软件层次上模拟中断的发生，用于提醒进程处理相应的事情。

# 5.

当信号的掩码中对应的位被设为1，此时内核也会将未决信号集中这个信号的对应的位设为1，此信号处于未决状态，此时该信号还没有被传递处理，内核不会再向进程发送处于未决状态信号。

用于解决常规信号在传递到进程时，会由于进程正在处理其它信号而被丢弃的问题。

# 6.

读函数和写函数通过共享内存交互，由先运行的函数来创建共享内存，后运行的函数打开共享内存，在共享内存中保存读函数和写函数的进程号。读函数通过共享内存获取写函数的进程号，检测到写函数执行完成退出，则执行读函数，写函数的执行过程类似。使用进程处理函数如 kill, signal 等来实现同步读写的过程。