1、

进程的概念被创建出来主要是用于描述一个正在运行的程序，以及其所使用的内存资源。进程不同于程序，进程是正在运行的程序，同一个程序可以运行多次，也就建立了不同的进程。Linux系统创建进程的概念，也就方便了对正在运行中的程序以及所占用的计算机资源有了准确的定义，能够在理论上研究并发编程。

2、

代码段A不仅仅在父进程中创建子进程，当创建完子进程之后，若循环还没有结束，那么创建好的子进程还会继续创建孙子进程。而代码段B中，在循环中判断了fork函数的返回值，当fork的返回值为0时，表示当前进程是fork函数创建的子进程，这时跳出了循环，子进程中便不在创建孙子进程。

3、

当一个进程执行完毕或者exit退出后，在其被父进程或者init进程回收之前，它会留下一个被称为僵尸进程的数据结构。它几乎不占据计算机的资源，也不能被执行和调度，仅仅是等待回收。

4、

Linux中的信号是在软件层次上对中断机制的一种模拟，用于提醒进程某件事情已经发生了。当目标进程接受到信号后，会执行这个信号的处理函数。通过信号，可以使得程序不是完全按照语句的顺序和逻辑执行，而是在需要的时候执行相应的处理过程。

5、

若发送的信号被阻塞，没有得到及时的处理，那么信号就会被设置为未决状态。对于可靠信号，如果处于未决状态，那么会被放入信号队列，而不是简单的丢弃，从而当目标进程解除阻塞之后，该信号得以被处理，而不会被丢失。

6、

当申请一块共享内存之后，可以返回一个指向共享内存的键值，并且申请一个该键值对应信号量，将信号量置为2，当有进程对共享内存执行写操作时，将信号量-2，信号量为0，此时，其他进程无论说读还是写操作都会被阻塞，当写操作执行完毕之后，将信号量+2，此时信号量再次为2，其他进程可以读写操作。当信号量为2时，所有进程都可以申请读操作，并且将信号量置为1，当有进程再进行读操作时，即信号量为1时，其他进程也可以同时进行读操作，但当信号量为1，也就是有进程在读时，会阻塞写操作。