1.

进程是系统中描述程序运行和资源共享的结构体，有了进程的概念就可以控制和协调程序的执行，包括使用kill对进程进行终止操作等。同时有了进程的概念后，系统可以让多个进程分时使用cpu资源，达到与多进程同时进行相同的效果，不必在当前程序执行完成后再执行其他程序，使计算资源分配更加合理。

2.

代码段A中父进程fork出子进程后，父进程和子进程都将继续执行循环，子进程的i与父进程当前的i相同，子进程也会产生孙子进程，因此包括最初的父进程在内最终共有2^5=32个进程。

代码段B中父进程fork子进程后，子进程的fork函数返回0，因此立刻退出循环，不再产生孙子进程，因此只有父进程会执行fork函数，最终共有5个子进程被fork出。

3.

僵尸进程指当子进程比父进程先结束，但父进程未调用wait或waitpid函数对该子进程进行回收，此时尽管子进程已经终止，但他的资源并未被释放，被称为僵尸进程。

父进程调用wait函数，进入阻塞状态，直到有子进程变为僵尸进程，否则会一直阻塞。当出现成为僵尸进程的子进程，wait函数获得该进程的信息，并立即返回该进程的进程id，并彻底清除该僵尸进程。

4.

信号提供了进程间通信的机制，是一种异步事件处理方法。信号用来通知进程发生了某事件，是一种软件中断的实现方法，可以让进程对发生的事件进行处理，如键盘键入ctrl+c会发出SIGINT中断信号，停止正在进行的程序。

5.

当发送的信号由于阻塞无法被传递到进程，内核会将该信号的状态置为未决状态。在信号递达之前都将处于未决状态，此时信号不会因为未递达而丢失，当取消阻塞后，该信号依然可以递达进程。未决和忽略的区别在于忽略是递达之后的一种处理动作。

6.

读写程序均需执行创建或打开共享内存操作。若创建成功则说明该程序先运行，则创建信号量，并将信号量初值置1；若创建不成功则说明程序后运行，则打开已被创建的共享内存。创建或打开成功后读写程序进入读写循环，循环需先等待信号量，即P操作，当信号量为0时程序一直处于等待状态，当信号量为1时信号量减1，程序继续运行，此时由于信号量为0，另一程序无法继续运行，即保证对于共享内存同一时间只有读或写操作。循环最后当读或写操作完成后对信号量进行V操作，即信号量加1，此时另一程序检测到信号量变化则继续运行。