**1.**

程序被触发后，运行者的权限与属性、程序的程序码与所需数据等都会被加载内存中， 操作系统并给予这个内存内的单元一个识别码 (PID)，可以说，进程就是一个正在运行中的程序，最终该命令可以进行的任务则与这个 PID 的权限有关。

**2.**

代码段A每循环一次，进程数翻倍，因此最终有32个进程；代码段B每循环一次多一个进程，因此最终有6个进程。原因：父进程fork函数返回值为子进程PID；子进程fork函数返回值为0；出现错误fork函数返回负值。

**3.**

僵尸进程是当子进程比父进程先结束，而父进程又没有回收子进程，释放子进程占用的资源，此时子进程将成为一个僵尸进程；一个进程在调用exit命令结束自己的生命的时候，其实它并没有真正的被销毁，而是留下一个称为僵尸进程（Zombie）的数据结构（系统调用exit，它的作用是使进程退出，但也仅仅限于将一个正常的进程变成一个僵尸进程，并不能将其完全销毁）。

wait系统调用会使父进程暂停执行，wait()函数捕获到子进程的退出信息时才会转为运行态，回收子进程资源并返回，即直到它的任意一个(并不是所有的)子进程结束为止。

**4.**

信号是进程间通讯的一种有限制的方式。它是一种异步的通知机制，用来提醒进程一个事件已经发生。当一个信号发送给一个进程，操作系统中断了进程正常的控制流程，此时，任何非原子操作都将被中断。如果进程定义了信号的处理函数，那么它将被执行，否则就执行默认的处理函数。

**5.**

信号产生和传递之间的时间间隔内，称此信号是未决的；也就是说一个已经产生的信号，但是还没有传递给任何进程，此时该信号的状态就称为未决状态。

作用：对进程来说，若信号的发送过于密集，在处理信号的同时再次收到信号，那么进程会将后到的信号丢弃。对于信号的发送方来说，应该发送的信号已经发送，不会再次发送；对于作为信号接收方的进程来说，未对信号做出应有处理。这显然是不符合预期的，所以需要信号处于未决状态，防止要处理的信号丢失。

**6.**

两个进程对同一个共享内存读写，一个进程读、一个进程写。通过信号量可以控制读写进程执行顺序，保证在写进程操作未完成时，其他进程不会从共享内存中读取数据。进程先从外部文件读取数据，将数据放入共享内存中，再通过fork函数创造子进程，用exec函数族转换子进程为写进程，通过写进程将共享内存的数据写入外部文件。