1、

在程序运行之后，操作系统需要保存程序运行的状态，管理程序拥有的资源，“进程”这个概念就是为了方便这样的目的而来的。程序在某个上下文的一次运行就是一个进程，将这样的概念抽象出来，为多任务执行的独立性和异步性提供了良好基础。

2、

在代码段a执行完会有16个进程，而代码段b执行完只有6个进程。在代码a中，每一个进程在循环中都会继续fork下去，而在b中，父进程的fork()返回是非0值，子进程（新被创建进程）的fork()返回0，就不会继续创建新进程。

3、

Linux中的进程在退出时，并不会释放全部空间，而是留下了一些数据等待其父进程收集。仅当父进程收集了其退出信息后，这个进程才能够完全退出，释放其所占用资源。而若父进程还未收集其推出信息，出于等待状态的，结束了的子进程就被称为僵尸进程。

父进程会用wait()收集已经结束的僵尸态的子进程的消息，而在其收集到之前，父进程挂起。若调用时无僵尸态子进程，则父进程会等到有子进程结束为止。在父进程调用了wait()后，子进程可以释放掉其占用的资源，完全退出。

4、

信号是进程之间相互通信的异步消息，通过信号，linux可以提醒一个进程某事件的发生，当信号发送和接收时，会中断任何非原子操作的操作，立即进行处理。

5、

信号未决是在信号产生到信号递达之间，信号所处于的状态。通常是阻塞信号导致了信号未决。信号未决这种状态的存在可以让信号在递达之前被存储，同时让进程在信号递达前可以知道信号正在被阻塞，避免了当前任务的打断。

6、

首先创建共享内存和信号量，然后将共享内存挂载到两个进程的地址空间，读写时，使用两个信号量sigr为0和sigw初始为1来控制读写。在写时，写的进程sem\_wait(sigw)，然后执行写操作，再sem\_post(semr)；在读取时，读取的进程执行sem\_wait(sigr)，读取操作，之后sem\_post(sem\_sigw)，如此循环。