1. 每个概念被创造都有其意义，请简述“进程”这个概念在Linux系统中有什么用途。

通过进程，Linux系统可以以操纵进程的状态来实现对程序的合理调用。使多个程序能够合理地使用CPU、内存等资源，提高资源的利用效率。

2.阅读以上代码段，回答代码段A和代码段B的执行结果有什么区别？并解释为什么会有这样的区别。

代码段A的执行结果要远多于6条（32条），而代码段B的执行结果为恰好6条。

因为代码段A在每一次执行fork函数后，产生的子进程也会在接下来的循环过程中执行下一次fork函数，不仅仅只是父进程执行fork，这样每次循环都会将进程数量翻倍。而代码段B对该进程是否是一个子进程进行了判断，如果是子进程那么就将跳出循环，因此循环过程中只有父进程创建了新的子进程。

3.用自己的话阐述什么是僵尸进程，并描述进程通过调用 wait() 捕获僵尸态的子进程的过程。

僵尸进程是一个进程在即将结束之时，放弃了之前执行时占用的资源（如内存），而且也没有代码去执行，也不能被调度，仅仅在进程列表中保留了一个位置，这里记载了该进程的退出状态等信息，等待父进程收集。

wait函数的调用者会被阻塞，直到子进程变为僵尸态，调用者捕捉到子进程的退出信息时，才会变回运行态，回收资源并返回子进程的PID，否则该进程将会一直保持阻塞状态，若当前有多个子进程，只要捕获其中一个子进程的信息，wait函数就会返回。若捕获失败，wait函数返回-1，并errno被设置为ECHILD。

4. 请简述信号在Linux系统中的作用

信号全程为软中断信号，实际上是软件层面对中断机制的一个模拟，用于提醒进程某件事情已经发生。

5. 请简述信号什么时候处于未决状态，并简述信号存在未决状态的作用。

发送的信号被阻塞，无法到达进程，内核就会将该信号的状态设置为未决。

有些时候由于信号发送的频率过于密集，即在处理信号的同时又有信号送达，那么进程会将后来的信号丢弃，对于信号的发送方来说，自然也不会再次发送信号，而该信号又没被接收方处理，因此这显然不合理。

信号屏蔽机制专门解决常规信号不可靠问题，在进程的PCB中存在两个信号集，一个称为信号掩码（signal mask），另一个称为未决信号集（signal pending）。这两个信号集的实质都是位图，其中的每一位对应一个信号；若mask中某个信号对应的位被设置为1，信号会被屏蔽，进入阻塞状态，此时内核会修改pending中该信号对应的位为1，使该信号处于未决态，之后除非该信号被解除屏蔽，否则内核不会再向进程发送这个信号。

我的理解是，顾名思义，未决信号就是还未被处理的信号，设置未决信号就是为了当信号过于密集时可以对违背处理的信号做标记，使得该信号在之后的时间里可以被进程处理。

6.设计一种通过信号量来实现共享内存读写操作同步的方式

可以考虑设置两个信号量来实现进行同步。A信号量控制读，初始化为0。B信号量控制写，初始化为1。（这个是百度到的…<http://www.manongjc.com/article/117134.html>）