# 1、

在LINUX里面，进程好像C++中一个一个的类一样。操作系统管理着这些类，让这些类运行或者不运行。这个类在Linux下叫做task\_struct，所以我们可以说task\_struct就是一个进程。所有task\_struct都是以双向链表链接这进行组织。

# 2、

A中父进程和子进程进入循环都会再fork一次，而B中子进程进循环就会break出去。

因为fork()后会返回一个数，当数大于0的时候，进入的是父进程，当返回值大于1的时候，就进入的是子进程。

# 3、

僵死进程就是当fork()创建子进程后，子进程直接调用exit()函数退出进程，父进程没有在进程结束后处理子进程而导致子进程仍在进程列表中保存一个位置。

进程一旦调用了wait，就立即阻塞自己，由wait自动分析是否当前进程的某个子进程已经

退出，如果让它找到了这样一个已经变成僵尸的子进程， wait就会收集这个子进程的信息，

并把它彻底销毁后返回；如果没有找到这样一个子进程，wait就会一直阻塞在这里，直到有

一个出现为止。

# 4、

信号是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。

# 5、

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:进程的信号屏蔽字规定了要阻塞传送到该进程的信号集。当产生了一个该进程已阻塞的信号，系统会为该进程生成一个信号集，并将该阻塞的信号保存在此信号集中即将此信号集对应的信号置为未决状态。此时用户可以通过在程序中以自定义的set位图与mask位图进行位操作，以达到屏蔽或解除屏蔽的目的，进一步对pending造成影响。

# 6、

1）reader和writer通过信号通信必须获取对方的进程号，可利用共享内存保存双方的进程号。

2）reader和writer运行的顺序不确定，可约定先运行的进程创建共享内存并初始化。

3）利用pause, kill, signal等函数可以实现该程序。