1. **进程在Linux系统中的用途：**

进程存在于系统内存之中，是具有独立功能的程序在数据集上运行的过程，它占用系统资源，是抽象的。在Linux系统之中，每输入一行命令就会启动一个进程，进程具有并发性，正因为进程的存在计算机才可以允许多个相互独立的程序同时进入内存，在内核的管理下相互之间交叉运行，使计算机能够同时并发执行多个程序并提供程序的相关属性。

1. **代码段A、B的执行结果：**

**代码段A：**

for循环执行5次，每次创建一个子进程，子进程同样创建，总共会有25=32个进程。

**代码段B：**

for循环同样执行5次，每次创建一个子进程，但是由于子进程的fork()函数返回值为0，所以子进程进入if语句，执行break退出，只有父进程继续创建，总共会有1(父进程)+5=6个进程。

1. **僵尸进程：**

僵尸进程即调用的exit()函数之后的进程。因为在调用exit()函数之后，进程并不会马上消失，而是清除进程占用的大部分内存后在进程列表中保留一个位置，记载进程的退出状态等信息供父进程调用，直到父进程回收子进程。

**wait()捕获僵尸态子进程的过程：**

进程调用wait()后，该进程进入阻塞状态，等待其子进程的结束（即进入僵尸态），在子进程进入僵尸态之后，wait()获取子进程的退出状态，并将其存放在参数states的低八位中，并返回僵尸态子进程的进程id，之后父进程继续执行。

1. **信号的作用：**

信号是软件层次上对中断机制的一种模拟，用于提醒某个进程某件事情已经发生。信号是实现Linux系统中进程间通信的重要方式之一。

1. **未决状态：**

当发送的信号被阻塞，无法达到进程，内核就会将该信号的状态设置为未决状态

**作用：**

信号存在未决状态后，进程则可以选择性阻塞信号，信号则会一直存在于未决信号集中，直到进程关闭阻塞，对未决信号进行处理，这样可以使进程能够暂时不处理产生的信号，先将其标注为未决，之后在适当时候对未决信号进行处理。

1. **实现思路：**

准备工作：

使用ftok()函数创建共享内存空间的key值，使用semget()函数创建信号量

定义读操作：

使用semop()函数获取信号量，即执行P操作

使用shmat()函数将共享内存映射到虚拟内存

进行读操作

使用shmdt()函数解除映射

使用semop()函数释放信号量，即执行V操作

定义写操作：

使用semop()函数获取信号量，即执行P操作

使用shmat()函数将共享内存映射到虚拟内存

进行写操作

使用shmdt()函数解除映射

使用semop()函数释放信号量，即执行V操作