

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。

进程概念是为了方便内核分配CPU的使用权而出现的。由于一个CPU在同一时间只能处理一个进程，为了使CPU的资源不浪费，内核需要将CPU的资源分配给不同的进程，以便在一个进程不使用CPU时资源不会被浪费。为了管理CPU的使用权，内核必须知道内存中有多少进程，目前正在使用的进程是什么，这就要求内核必须能够区分进程并可获取进程的相关属性，也就创造出了进程的概念。

2. 阅读以下代码段，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
int i;
for(i=0;i<5;i++)
{
    pid=fork();
}
```

```
//代码段B
int i;
for(i=0;i<5;i++)
{
    if((pid=fork())==0)
        break;
}
```

代码段A执行完会创建31个子进程，代码段B执行完只会创建5个子进程。因为在代码段A中每次循环每个进程都会创建自己的子进程，所以会有 $2^5-1=31$ 个子进程，而在代码段B中，由于fork()函数返回创建的子进程的pid，而被创建出的子进程的fork()的返回值为0，所以每次创建出的子进程的pid值为0，会break，只有一开始的父进程会做5次循环，并在每个循环中创建一个子进程，最终只会创建5个子进程。

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。

当子进程退出时并不会完全销毁自身，而是会留下一部分信息以便父进程调用，而这部分信息直到父进程调用之前都是不会清除的。如果父进程既没有利用wait()等函数等待子进程结束，也没有接收来自子进程的信号，那么子进程所留下的包括进程号在内的信息就会一直得不到释放，从而变成僵尸进程。需要注意的是如果父进程在子进程结束之前就已经结束，那么子进程不会变成僵尸进程，而是会被init进程收养，成为init进程的子进程。

父进程调用wait()时会阻塞自己，直到接收到一个子进程结束的信号才会继续运行，如果父进程创建了多个子进程，那么对每一个子进程都要wait一次才能确保回收了所有子进程。

4. 请简述信号在Linux系统中的作用。

信号全称软中断信号，其本质是软件层次上对中断机制的一种模拟，用于提醒某进程某件事情已经发生。信号的主要作用是实现Linux系统中的进程间通信，使某一事件发生时其他进程可以对其作出处理。

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，无法到达进程时，内核就会将该信号的状态设置为未决。

信号的未决状态一方面使目标进程的某些关键工作不被打断，同时又不会错过信号，因为当目标进程恢复时系统会将信号再次递达，以便目标进程处理。

6. 请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，文字阐述即可，不需要代码实现。（提示：在写进程操作未完成时，需要防止其他进程从共享内存中读取数据）

创建两个二维信号量，分别代表读写两种操作，当一个进程开始进行写操作时对读信号量和写信号量执行P操作，这样当其他进程想要进行读或者写操作时就会被阻塞，当写操作执行结束再执行V操作。当执行读操作时只对读信号量执行P操作，此时如果有写信号传入则向读操作传入term信号，中断读操作转而执行写操作，如果没有则完成读操作后再对读信号执行V操作。