

## 第十二周

### 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。

用于对正在运行的程序过程的抽象，是一种数据结构，目的在于刻画动态系统的内在规律，有效管理和调度进入计算机系统主存储器运行的程序。

### 2. 阅读代码段，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一共会产生32个子进程，因为其子进程会继续进行循环产生子进程，而代码段B只会产生5个子进程，因为其产生的子进程会跳出循环不会产生子进程。

### 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，就是进程调用 `exit()` 函数后的特殊进程状态，其只保留了一个在进程表中的位置而已经没有可执行代码，也不能被调度。

进程调用 `wait()` 函数后会被挂起，进入阻塞状态，直到子进程进入僵尸态，`wait()` 函数捕获到该子进程的退出信息时才会转为运行态。

### 4. 请简述信号在Linux系统中的作用。

信号是Linux系统进程间通信的方式之一，它是一种异步的通知机制，用来提醒进程一个事件已经发生。其本质是在软件层次上对中断机制的一种模拟。

### 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。处于未决状态的信号，并不会丢失，一旦进程取消阻塞信号依然会到来。存在未决状态可以让进程在阻塞信号时知道有哪些信号被其阻塞了，解决了常规信号不可靠的问题。

## 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）

---

设置两个信号量 `WR` 和 `lock` 均初始化为1，同时设置一个变量 `reader_num` 标识当前读线程的数量，当写操作开始时对 `WR` 进行P操作，写操作结束时对 `WR` 进行V操作。当读操作开始时首先对 `lock` 进行P操作，然后 `reader_num+=1`，之后如果 `reader_num==1` 则对 `WR` 进行P操作，然后对 `lock` 进行V操作；读操作结束时首先对 `lock` 进行P操作，然后 `reader_num-=1`，之后如果 `reader_num==0` 则对 `WR` 进行V操作，然后对 `lock` 进行V操作。

在这个设计中信号量 `WR` 用于保证读写操作的互斥，信号量 `lock` 则保证不同的读进程之间修改 `reader_num` 时的互斥，只有在启动第一个读进程时才会对 `WR` 进行P操作，只有当最后一个读进程结束时才会对 `WR` 进行V操作。