互斥：是指某一资源同时只允许一个访问者对其进行访问，具有唯一性和排它性。但互斥无法限制访问者对资源的访问顺序，即访问是无序的。

同步：是指在互斥的基础上（大多数情况），通过其它机制实现访问者对资源的有序访问。在大多数情况下，同步已经实现了互斥，特别是所有写入资源的情况必定是互斥的。少数情况是指可以允许多个访问者同时访问资源

进程的状态是基于一定的原因和条件而变化的.而这些原因和条件又常常是因为进程间的相互制约关系而引起的.系统中进程会有这种关系.主要是由于进程对资源的共享性,由于这种共享的特征,使系统中原来没有逻辑关系的进程因为互相竞争资源而发生了制约关系.这种关系的基本形式"进程-资源-进程"这是进程间通过资源而发生的一种间接关系.由于系统对进程所请求的许多资源常常是互斥满足的,所以这咱关系表现为互斥关系.又由于系统中为了完成同一个任务而创建若干进程,它们之间必然是伙伴进程.如作业的一组并行进程共同完成一项任务,有时它们要在某点上互相等待或互通消息,这种关系的基本形式是"进程-进程",这是进程之间的一种直接关系,表现了进程之间的协同工作的特性,称为进程间的同步关系.例如:现在A,B两个进程,A进程负责输入数据并送入缓冲区中,B进程必须负责加工缓冲区中的数据并输出,为了保证这两个进程之间能够协同工作.则B进程必须等待A进程发来缓冲区已满的信号后才开始工作.A进程必须等待B进程发来缓冲区中的数据加工并输出后才能再输入下一批数据.这体现 了进程之间的的同步关系.又由于这两个进程共用同一个缓冲区.因此,它们对缓冲区的操作还应该作到互斥.操作系统中进程的这种协作关系是屡见不鲜的.

**问题描述：**设公交车上，司机和售票员的活动分别如下:

               司机的活动：启动车辆；正常行车；到站停车。

               售票员的活动：关车门；售票；开车门。

               在汽车不断地到站，停车，行驶过程中，这两个活动有什么同步关系？用信号量和pv操作实现它们的同步。

**问题分析：**在汽车行驶过程中，司机活动与售票员活动之间的同步关系为：售票员关车门后，向司机发开车信号，司机接到开车信号后启动车辆，在汽车正常行驶过程中售票员售票，到站时司机停车，售票员在车停后开门让乘客上下车。因此，司机启动车辆的动作必须与售票员关车门的动作取得同步；售票员开车门的动作也必须与司机停车取得同步。应设置两个信号量：S1、S2;

                S1表示是否允许司机启动汽车（其初值为0）

                S2表示是否允许售票员开车门（其初值为0）

**用pv原语描述如下：**

                 the PV code using pascal

                 var S1,S2:semaphore;

                 S1=0;S2=0;

                 cobegin

                 Procedure driver

                 begin

                 while(TURE)

                 begin

                 P(S1);

                 start;

                 driving;

                 stop;

                 V(S2);

                 end

                 end

                 Procedure conductor

                 begin

                 while(TURE)

                 begin

                 关车门；

                 V(S1);

                 售票；

                 P(S2);

                 开车门；

                 上下乘客；

                 end

                 coend

