

第二章习题

Karry

1. D

解释：临界区（Critical section）的定义为：访问临界资源的那段代码

2. C

解释：由程序段、相关数据段和 PCB 三部分构成了进程实体，简称为进程

3. B 4.C 5.B 6.A 7.C 8.D 9.B 10.D

4. 答：进程之间存在直接相互制约关系（同步）和间接相互制约关系（互斥），前者是由于各进程共同完成一项任务相互等待产生的，后者是由于竞争使用共享资源而产生的。

（1）属于互斥关系：书是有限数量的公共资源，一本书在一个时间点只能借个一位同学。（当然如果考虑还书的话，也存在同步关系，借书前要先看书的数目，借用导致书的数目减少而还书导致书的数目增加）

（2）属于互斥关系：篮球是公共资源，篮球只有一个两队都要争夺。

（3）属于同步关系：各道中间工序的开始依赖前道工序的完成。

（4）属于同步关系：商品要先被生产出来才能被社会消费；已有的商品如果没有被消费完，生产也无需进行。

5. 答：这两个活动的同步关系为：

- 售票员关车门后，向司机发开车信号，司机收到开车信号后启动车辆；
- 在正常行车过程中售票员售票；
- 到站时司机停车后，向售票员发出开车门信号，售票员在收到开车门信号后开车门，乘客上下车。

所以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必须与司机停车的动作取得同步。因此设置两个信号量 S, O:

- S 表示是否允许司机启动汽车，初始值为 0

- O 表示是否允许售票员打开车门，初始值为 0

伪代码如下：

```
// 初始化信号量
int S = 0, O = 0;
// 司机
void driver(){
    do {
        P(S);
        启动汽车();
        正常行使();
        到站停车();
        V(O);
    } while(TRUE);
}
// 售票员
void conductor(){
    do {
        关车门();
        V(S);
        售票();
        P(O);
        开车门();
        下乘客();
        上乘客();
    } while(TRUE);
}
```

6. 答：该关系可以类比生产者消费者问题，只不过是有多多个生产者和多个消费者。设置以下信号量：

- mutex: 实现诸进程对盘子的互斥使用，初始值为 1
- empty: 表示盘子是否有空间，初始值为 1
- apple, orange: 分别表示是否有苹果和橘子，初始值均为 0

伪代码如下：

```
// 初始化信号量
semaphore mutex = 1, empty = 1, apple = 0, orange = 0;
// 爸爸
void father(){
```

```
        do{
            P(empty);
            P(mutex);
            削苹果();
            V(mutex);
            V(apple);
        } while(TRUE)
    }
    // 妈妈
    void mother(){
        do{
            P(empty);
            P(mutex);
            剥桔子();
            V(mutex);
            V(orange);
        } while(TRUE);
    }
    // 儿子
    void son(){
        do{
            P(orange);
            P(mutex);
            吃桔子();
            V(mutex);
            V(empty);
        } while(TRUE);
    }
    // 女儿
    void daughter(){
        do{
            P(apple);
            P(mutex);
            吃苹果();
            V(mutex);
            V(empty);
        } while(TRUE);
    }
```