

- 临界区问题的解决方案必须满足的三个要求是什么？下面是 Dekker 给出的关于 2 个进程的、纯软件的临界区问题解决方案。2 个进程 P_0 和 P_1 ，共享如下变量：

```
boolean flag[2]; /* initially false */  
int turn;
```

2 个进程 P_0 和 P_1 的程序结构如右图所示。**证明**右边的算法满足临界区问题解决方案的三个要求。

```
do{  
    flag[i]=false;  
    while(flag[j]){  
        if(turn==j){  
            flag[i]=false;  
            while(turn==j)  
                ;//do nothing  
            flag[i]=TRUE;  
        }  
    }  
    //critical section  
    turn=j;  
    flag[i]=FALSE;  
    //remainder section  
}while(TRUE);
```

- 7.1 术语**忙等**的含义是什么？操作系统中其他种类的等待有哪些？忙等能否完全避免，why？
What is the meaning of the term **busy waiting**? What other kinds of waiting are there in an OS? Can busy waiting be avoided altogether? Explain your answer.
- 使用记录型信号量设计一种避免死锁并且不会饿死的哲学家问题解决方案。
- 7.8 **理发店问题**。一个理发店有一间配有 n 个椅子的等待室和一个有理发椅的理发室。如果没有顾客被服务，理发师就睡觉。如果顾客来时所有的椅子上都有人，那么顾客离去。如果理发师在忙而有空闲的椅子，那么顾客就会坐在其中一个空闲的椅子上。如果理发师在睡觉，顾客会摇醒他。写一个程序来协调整理发师和顾客。