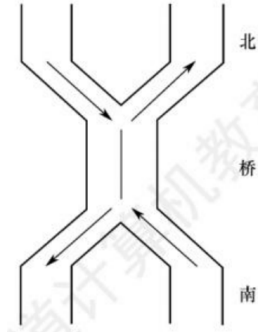


《操作系统》期末速通教程

8. PV 问题强化训练

[例] [单行道问题] 某桥如下图所示, 箭头表示车流方向. 桥上不允许两车交会, 但同方向一次允许多辆车通过. 用信号灯的 PV 操作实现桥的交通管理.



[解]

```

1  int cntSN = cntNS = 0; // 南到北、北到南的车数
2  Semaphore mutexSN = mutexNS = 1; // 互斥访问 cntSN 和 cntNS
3  Semaphore bridge = 1;
4
5  S2N() {
6      P(mutexSN);
7      if (cntSN == 0) {
8          P(bridge);
9      }
10     cntSN++;
11     V(mutexSN);
12
13     过桥;
14
15     P(mutexSN);
16     cntSN--;
17     if (cntSN == 0) {
18         V(bridge);
19     }
20     V(mutexSN);
21 }
22
23 N2S() {
24     P(mutexNS);
25     if (cntNS == 0) {
26         P(bridge);
27     }
28     cntNS++;
29     V(mutexNS);
30
31     过桥;
32
33     P(mutexNS);
34     cntNS--;
35     if (cntNS == 0) {

```

```

36     V(bridge);
37 }
38 V(mutexNS);
39 }

```

[例] [取号-叫号问题 1] 面包师有很多面包, 由 n 名销售人员推销. 每名顾客进店后按序取一个号并等待叫号. 一名销售人员空闲时 会按序叫下一个号. 用两个整型变量记录当前的取号值和叫号值, 设计一个销售人员和顾客同步的算法.

[解]

```

1  int i = j = 0; // 取号值、叫号值
2  Semaphore mutex_i = 1, mutex_j = 1;
3
4  Consumer() {
5      进店;
6      P(mutex_i);
7      取号 i;
8      i++;
9      V(mutex_i);
10
11     等叫号;
12 }
13
14 Seller() {
15     while (1) {
16         P(mutex_j);
17         if (j < i) { // 有 i 号顾客
18             叫号 j;
19             j++;
20             V(mutex_j);
21
22             推销;
23         }
24         else {
25             V(mutex_j);
26
27             休息;
28         }
29     }
30 }

```

[例] [取号-叫号问题 2] 理发店有一位理发师、一把理发椅和 n 把给等候顾客的椅子. 若无顾客, 则理发师在理发椅上睡觉. 顾客到来时, 若理发师在睡觉, 需叫醒理发师. 若在理发师理发时有顾客来到, 若有空椅子, 则顾客坐下等待; 否则顾客离开. 用 PV 操作实现理发师与顾客同步.

[解]

```

1  int waiting = 0; // 等待的顾客数
2  Semaphore mutex; // 互斥访问 waiting
3  int chairs = n; // 给顾客的空闲椅子数
4  Semaphore customers = barbers = 0;
5
6  Barber() {
7      while (1) {
8          P(customers);
9
10         P(mutex);
11         waiting--;
12         V(mutex);
13
14         理发;
15         理发结束;
16         V(barbers);
17     }
18 }
19
20 Customer() {
21     P(mutex);
22     if (waiting >= chairs) { // 无空位
23         离开;
24         V(mutex);
25     }
26     else {
27         waiting++;
28         V(mutex);
29
30         等待理发;
31         V(customers);
32
33         轮到理发;
34         P(barbers);
35     }
36 }

```