

课程作业

课程名称 : 操作系统
作业名称 : 课后习题1
学号 : 21281280
姓名 : 柯劲帆
班级 : 物联网2101班

1. 问题1

哲学家进餐问题的变形。假设5支筷子都放在圆桌的中间，哲学家进餐时拿起任意两支筷子就可以进餐。其他条件不变。请用信号量实现5个哲学家进程的同步。

```
1  Var
2      chopsticks, eating_philosophers: Semaphore := 5, 4;
3
4  Procedure Philosopher(i: integer);
5  Begin
6      Repeat
7          wait(eating_philosophers);
8          wait(chopsticks);
9          wait(chopsticks);
10
11         Eat;
12
13         signal(chopsticks);
14         signal(chopsticks);
15         signal(eating_philosophers);
16
17         Think;
18
19     Until False;
20 End;
```

2. 问题2

四个进程P0,P1,P2,P3和四个信箱M0,M1,M2,M3进程间借助相邻的信箱传递消息：Pi每次从Mi中取出一条消息，经加工送入Mi+1(mod 4)中。其中M0,M1,M2,M3分别设有3,3,2,2个格子，每个格子放一条消息，初始时，M0装满了三条消息，其余为空。写出使用信号量实现进程 (i=0,1,2,3)同步的算法。

```
1  Var
2      full, empty, mutex: array[0..3] of Semaphore := (3, 0, 0, 0), (0, 3, 2,
3      2), (1, 1, 1, 1);
4
5  Procedure ProcessPi(i: integer);
6  Var
7      message: Message;
8  Begin
9      Repeat
10         wait(full[i]);
11         wait(mutex[i]);
```

```

11     message := Get_Message_From_Mailbox(i);
12     signal(mutex[i]);
13     signal(empty[i]);
14
15     ProcessMessage;
16
17     wait(empty[(i + 1) mod 4]);
18     wait(mutex[(i + 1) mod 4]);
19     Send_Message_To_Mail_box((i + 1) mod 4, message);
20     signal(mutex[(i + 1) mod 4]);
21     signal(full[(i + 1) mod 4]);
22
23     Until False;
24 End;

```

3. 问题3

汽车司机与售票员之间必须协同工作，汽车每到一个站后，一方面，只有当司机已经停下，售票员才能开门上下客，另一方面，只有售票员把车门关好了司机才能开车。假定某辆公共汽车上有前后门各有一名售票员以及一名司机，汽车初始状态是正在始发站停车上客。试设必要的信号量及赋初值，实现售票员进程和司机进程的同步。

```

1  Var
2      driver, ticket_seller: Semaphore := 1, 2;
3
4  Procedure Driver;
5  Begin
6      Repeat
7          wait(driver);
8
9          Move;
10
11         Stop;
12
13         signal(ticket_seller);
14         signal(ticket_seller);
15
16         Until False;
17 End;
18
19 Procedure Ticket_seller(i: integer);
20 Begin
21     Repeat
22         wait(ticket_seller);
23         wait(ticket_seller);
24
25         Open;
26
27         signal(driver);
28
29         Close;
30
31         Until False;
32 End;

```

