```
semaphore mutex1, mutex2, mutex3, mutex0; //限制同一信箱同时只能有一个进程操作
mutex1 = mutex2 = mutex3 = mutex0 = 1;
semaphore put1, put2, put3, put0; //限制能否放信件
put1 = 3; put2 = 2; put3 = 2; put0 = 0;
semaphore get1, get2, get3, get0; //限制能否取信件
get1 = 0; get2 = 0; get3 = 0; get0 = 3;
int in0, in1, in2, in3, out0, out1, out2, out3; //存放和取出信件的位置指针
in0 = in1 = in2 = in3 = out0 = out1 = out2 = out3 = 0;
cobegin
```

```
Process P0(){
  while(1){
    P(get0);
      P(mutex0);
        {从M0[out0]取一条消息};
        out0 = (out0 + 1) % 3;
      V(mutex0);
    V(put0);
    {加工消息};
    P(put1);
      P(mutex1);
        {消息存M1[in1]};
        in1 = (in1 + 1) % 3;
      V(mutex1);
    V(get1);
  }
}
```

```
Process P1(){
  while(1){
    P(get1);
      P(mutex1);
        {从M1[out1]取一条消息};
        out1 = (out1 + 1) % 3;
      V(mutex1);
    V(put1);
    {加工消息};
    P(put2);
      P(mutex2);
        {消息存M2[in2]};
        in2 = (in2 + 1) % 2;
      V(mutex2);
    V(get2);
  }
}
```

```
Process P2(){
  while(1){
    P(get2);
      P(mutex2);
        {从M2[out2]取一条消息};
        out2 = (out2 + 1) % 2;
      V(mutex2);
    V(put0);
    {加工消息};
    P(put3);
      P(mutex3);
        {消息存M3[in3]};
        in3 = (in3 + 1) % 2;
      V(mutex3);
    V(get3);
  }
}
```

```
Process P3(){
  while(1){
    P(get3);
      P(mutex3);
        {从M3[out3]取一条消息};
        out3 = (out3 + 1) % 2;
      V(mutex3);
    V(put3);
    {加工消息};
    P(put0);
      P(mutex0);
        {消息存M0[in0]};
        in0 = (in0 + 1) % 3;
      V(mutex0);
    V(get0);
  }
}
```

```
coend
```