

### Cigarette-smokers problem

Có 1 agent, sẽ phát sinh ra 2 trong 3 nguyên liệu sau: giấy, thuốc lá và diêm. Và có 3 người hút thuốc, một có nguồn giấy vô hạn, 1 có nguồn thuốc lá vô hạn, và 1 có nguồn diêm vô hạn.

Để làm một điếu thuốc thì cần 3 nguyên liệu trên. Khi làm xong điếu thuốc thì hút.

Nếu mà agent phát sinh ra 2 nguyên liệu, thì agent tạm ngưng cho đến khi có người tiêu thụ nó, người có nguyên liệu còn lại để có thể kết hợp 2 nguyên liệu trên làm thành điếu thuốc thì được nhận nguyên liệu.

### Bài toán đơn giản:

Agent có thể biết các tiêu trình người hút thuốc để yêu cầu người này nhận nguyên liệu.

Gợi ý bài giải:

Khởi tạo

Semaphore  $T = 1$ ;

Semaphore  $A[0] = A[1] = A[2] = 0$ ; //  $A[i]$ : cho smoker thứ  $i$

Chương trình của agent

```
while true {  
    Chọn ngẫu nhiên 2 nguyên liệu rồi signal cho người đang có nguyên  
liệu còn lại  
    up( $A[k]$ );  
    down( $T$ );  
}
```

Chương trình của smoker

```
while true {  
    down( $A[i]$ );  
    make a cigarette  
    up( $T$ );  
    smoke the cigarette  
}
```

### Bài toán khó hơn:

Agent không biết tiêu trình nào là đang cần thiếu những nguyên liệu nào

Gợi ý bài giải:

1  $isTobacco = isPaper = isMatch = False$

2 Semaphore  $tobaccoSem = 0$  // dùng để signal người hút có tobacco

3 Semaphore  $paperSem = 0$  // dùng để signal người hút có paper

4 Semaphore  $matchSem = 0$  // dùng để signal người hút có match

5 Semaphore  $tobacco = 0$ ,  $paper = 0$ ,  $match = 0$ : dùng để signal nguyên liệu đang có

6 Semaphore  $agentSem = 0$ ;

Các biến  $isTobacco$ ,  $isPaper$ ,  $isMatch$  là để báo cáo nguyên liệu nào đang có sẵn ở trên bàn. Chúng ta tạo ra 3 Pusher: A, B, C, mỗi Pusher chuyên chờ 1 nguyên liệu, khi mà pusher nào nhận được nguyên liệu thì sẽ signal cho smoker nếu đã đủ 2 nguyên liệu hoặc là bật cờ báo hiệu đã có nguyên liệu này.

Chương trình của agent

```
1 Phát sinh ngẫu nhiên nguyên liệu
2 if tobacco
    tobacco.signal(); // = up(tobacco)
3 else if paper
    paper.signal();
4 else
    match.signal();
```

Pusher A

```
1 tobacco.wait() // = down(tobacco)
2 mutex.wait()
3     if isPaper:
4         isPaper = False
5         matchSem.signal()
6     elseif isMatch:
7         isMatch = False
8         paperSem.signal()
9     else:
0         isTobacco = True
1 mutex.signal()
```

Smoker with tobacco

```
1 tobaccoSem.wait()
2 makeCigarette()
3 agentSem.signal()
4 smoke()
```

### **The barbershop problem (Dijkstra )**

Một tiệm hớt tóc có 1 phòng đợi n ghế, và phòng cắt tóc với 1 ghế cắt tóc. Nếu không có khách thì người thợ đi ngủ. Nếu một khách mới vào và không còn ghế trống thì người khách rời tiệm. Nếu người thợ đang cắt tóc, và vẫn còn ghế trống thì khách hàng chọn 1 ghế ngồi đợi. Nếu người thợ ngủ, thì khách hàng đánh thức anh ta. Viết chương trình để đồng bộ khách hàng và thợ cắt tóc.

Gợi ý bài giải:

Customer

```
1 mutex.wait()
2     if customers == n+1:
3         //rời quán
5     else customers += 1
6 mutex.signal()
7
8 customer.signal()
```

```

9 barber.wait()
10 getHairCut()
11
12 mutex.wait()
13     customers -= 1
14 mutex.signal()

```

```

Barber
1 customer.wait()
2 barber.signal()
3 cutHair()

```

### Hilzer barbershop

Tiệm cắt tóc có 3 ghế, 3 thợ, 4 chỗ đợi trên sofa, và 16 chỗ đứng. Khách hàng sẽ không vào tiệm nếu nó đã đầy khách. Khi vào tiệm, khách hàng sẽ ngồi đợi trên sofa hoặc phải đứng đợi nếu sofa đã đầy. Khi 1 thợ rảnh rồi thì khách ngồi đợi trên sofa lâu nhất sẽ được vào cắt, và nếu có khách hàng đứng đợi thì ai đứng đợi lâu nhất được ngồi vào sofa. Khi cắt tóc xong, thợ có thể thu tiền khách, nhưng chỉ có 1 hộp đựng tiền nên chỉ có thể nhận tiền của 1 khách tại 1 thời điểm. Các hành động của 1 thợ có thể là: cắt tóc, thu tiền, ngủ nếu không có khách hàng.

Giải thích thêm: thứ tự các hành động có thể của khách hàng là: Vào tiệm, ngồi trên sofa, được cắt tóc, trả tiền, rời khỏi tiệm.

Gợi ý bài giải:

```

1     customers = 0 //đếm số khách hàng
2     mutex = Semaphore(1) //mutual exclusion semaphore
3     standingRoom = Fifo(16) // semaphore cho 16 chỗ đứng đợi
4     sofa = Fifo(4) //semaphore cho ghế sofa
5     chair = Semaphore(3) //semaphore cho ghế cắt tóc
6     barber = Semaphore(0) //semaphore cho thợ cắt tóc
7     customer = Semaphore(0) //semaphore cho khách hàng sẵn sàng cắt
8     cash = Semaphore(0) //semaphore cho việc trả tiền
9     receipt = Semaphore(0) //semaphore cho hóa đơn

```

```

Customer
1 mutex.wait()
2     if customers == 20:
3         mutex.signal()
4         exitShop()
5     customers += 1
6 mutex.signal()
7
8 standingRoom.wait()
9 enterShop()
10

```

```
11 sofa.wait()
12 sitOnSofa()
13 standingRoom.signal()
14
15 chair.wait()
16 sitInBarberChair()
17 sofa.signal()
18
19 customer.signal()
20 barber.wait()
21 getHairCut()
22
23 pay()
24 cash.signal()
25 receipt.wait()
26
27 mutex.wait()
28     customers -= 1
29 mutex.signal()
30
31 exitShop()
```

Barber

```
1 customer.wait()
2 barber.signal()
3 cutHair()
4
5 cash.wait()
6 acceptPayment()
7 receipt.signal()
```