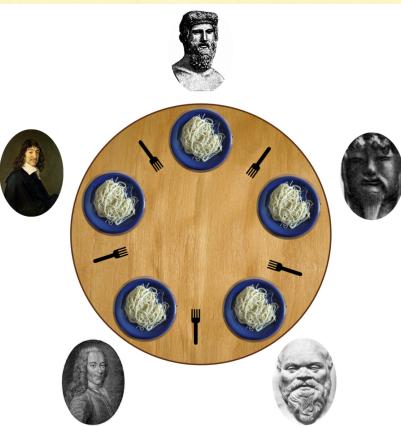


Bài toán:



- Có 5 ng^o ngồi quanh 1 chiếc bàn tròn, trc mặt l^oa mặt chⁱc d^o ān, mặt chiếc đĩa như hình
 - 1 triết gia chỉ có 2 trạng thái nghⁱ: Ng^ei, ko l^om gⁱ
ān: l^oay l^oay 2 chiếc đĩa. L^oay t^ung c^oi t^ui 1 thời điểm, l^oay cái bên trái
trc rồi l^oay cái bên phải. (đ^u 2 cái đĩa kh^o m^oi ān đ^c)

Khi muốn ān phải nháu 1 chiếc đĩa bên trái trc nh^ut cⁱc (do l^ong + đang dùng) \rightarrow đ^c
nh^ut cⁱc \rightarrow c^o g^{ang} xem C^o: Ăn
c^o nh^ut cⁱc đĩa kh^o 2 ko Ko : th^oi, nhưng điều quan trọng là
trong kh^o chⁱ đĩa bên phải, c^ong ta
k^o n^hu đĩa bên trái ra

Chi^osau Khi ān xong m^oi trả lại đĩa theo thứ tự ngược lại (phai tre tru^usau)

=> Tài nguyên găng (Critical resource lock): 5 cái đĩa, mỗi cái là 1 critical resource và giới hạn phục vụ qua 1 critical resource là 1. (v^o 1 chiếc đĩa chỉ có thể đ^c sd b^o);
 Số^r process : 5, trong ctⁱ giao^rдан găng c^o m^oi process l^oa khi n^o e^rtrong phai^r dùng đĩa L^ong t^ui 1 thời

Solution 1: Mỗi chiếc đĩa là 1 critical resource, ctⁱ kh^o hiên = 1 tên bao fork[i] \rightarrow khởi tạo 1

mảng 5 tên bao fork[5] = {1, 1, 1, 1, 1} Do gian phuc v^o chⁱl^oi 1 n^o t^ui 1 thời

```
do{
    wait(fork[i]) | Nếu đĩa bên trái
    wait(fork[(i+1)% 5]); | Khi đĩa trái thì cti tiếp đc đĩa phải, % 5 đc quay vòng vo cti sau, vo ng thứ 5  $\Rightarrow$  i+1 = 6
    { Ăn}
    signal(fork[(i+1)% 5]); | Giúp phêng đĩa phải
    signal(fork[i]); | Giúp phêng đĩa trái
    {Nghi}
} while (1);
```

phai^r l^oay ctⁱ quay l^oay l^oay chiếc đĩa thứ 1
 \Rightarrow làm bao^r t^ui 1 thời 1 chiếc đĩa chỉ đ^c 1 n^o sử dụng (mutual exclusive)

\Rightarrow Tuy nhiên, nếu TM c^o 5 n^o c^ong l^oay chiếc đĩa bên trái của họ t^ui 1 thời điểm \Rightarrow c^o 5 n^o c^ong phai^r ng^oi^r ctⁱ chiếc đĩa bên phải v^o t^on \Rightarrow Dead Lock

Solution 2: Chỉ cho phép 1 n^o l^oay đĩa t^ui 1 thời điểm. Như bên trên, khi c^o 5 n^o c^ong muôn l^oay đĩa t^ui

```
do{
    wait(mutex)
    wait(fork[i])
    wait(fork[(i+1)% 5]);
    signal(mutex)
    { Ăn}
    signal(fork[(i+1)% 5]);
    signal(fork[i]);
    {Nghi}
} while (1);
```

1 thời điểm, họ cùng gọi wait(mutex) \rightarrow chỉ c^o 1 n^o ctⁱ sel tên bao này, sau ctⁱ n^g a^ry gọi tiếp fork[i] va^r fork[i+1]. Ng^o này l^oay xong 2 cái đĩa rồi t^om^oi n^ha^r mutex ra
 cho n^g t^o có quyền l^oay đĩa
 $* 2 n^o k^o t^o chon c^ong l^oay 1 n^o chen xung, tr^o l^oai mutex cho n^g t^o quyền chon \Rightarrow c^o 2 n^o
 c^o utri phu hop v^on c^o t^o c^ong ctⁱ thời ān$

=> Tuy nhiên có thể xảy ra lỗi lầm cho 2 ng^o k^o t^o chung 1 thời điểm, và khi n^gs^ẽ 1 ăn (chiếm đⁱa 0 v^à 1), n^gs^ẽ 2 c^ó muốn ăn, n^g n^{ay}, c^tc phép truy cập vào mutex, nhưng khi gọi wait(fork[1]) s^ẽ phải đợi vⁱk^o r^õ c^ó đⁱa 1 trong th^g n^{ay} h^ov^àn chiếm dụng đèn báo mutex khiến cho n^gs^ẽ 3 đ^um^uon ăn vⁱc^ó 2 đⁱa c^ó hai bên n^gs^ẽ 3 đ^um^uon c^tth^é s^ử dụng nh^ung n^gn^{ay} l^à k^o c^tc quyền truy cập vào mutex c^t^onh^{ất} 2 chi^c đⁱa l^à m^uon => l^à m^uon c^tc => Vi pham điều kiện tiến triển (progress condition)

Solution 3: thu^t từ l^àu^y chia c^ó m^uin^ho^o l^àu kh^óc nhau < n^gs^ẽ hi^u ch^án l^ày đⁱa b^{ên} trái m^uo
n^gs^ẽ hi^u l^ào^y đⁱa^o ph^át tr^c

• Thuật toán cho Triết gia P_i

```
do{
    j = i%2
    wait(fork[(i + j)%5])
    wait(fork[(i+1 - j)% 5]);
    { Ăn}
    signal(fork[(i+1 - j)% 5]);
    signal(fork[(i + j)%5]);
    {Nghỉ}
} while (1);
```

=> C^ó c^tc quyết đⁱc vⁱc^é decablock

Nh^ung l^àu^y vi pham, thay đổi cách ban chia c^ó đⁱa b^{ên}

Solution 4: Tra^o l^àu^y đⁱa b^{ên} trái n^{ếu} k^o l^àu^y đⁱc đⁱa b^{ên} ph^át:

- l^àu^y đⁱa b^{ên} ph^át s^ẽ s^{ống} tr^c khi gọi wait(fork[i+1] % 5)

- N^{ếu} k^o có s^{àn} \rightarrow tra^o đⁱa tr^ailai, c^t^oi 1 th^g r^õ kh^ó l^àu^y

\rightarrow k^o p^í dead lock, nh^ung v^àn vi pham ng^ut^í tiến triển, g^ăp vⁱc^é Starvation (nan^lo^o): 5 ng^o c^ó g^ăo^o l^àu^y đⁱa tr^co^o,

đ^en check kh^óy đⁱa ph^át k^o c^ó s^{àn} \rightarrow c^ó g^ăo^o l^àu^y r^õ ngoⁱ d^ét^í ngoⁱn^hien kh^óc^hnhau

tuy nh^unhien vⁱ m^át l^ý thuy^êt ta k^o dam^hao c^tc điều n^{ày}, nh^u kh^ó

ch^ó c^ó 5 ng^o l^àu^y nh^un^hau? \rightarrow Ch^ó c^ó v^ôt^{án}

* L^àu^y g^ăo^o l^àu^y life lock: các process vẫn hot b^{inh} h^ong nh^ung k^o tiến triển c^tc, k^o kh^ó vⁱc^é dead lock l^àu^y k^o r^õ process đang ph^át ch^óm^á v^àn m^át processor.

Solution 5:

- Sử dụng đèn báo đồng thời $P_{Sim}(S_1, S_2, \dots, S_n)$
 - Th^u được tất cả đèn báo cùng một thời điểm hoặc không có bất kỳ đèn báo nào
 - Thao tác $P_{Sim}(S_1, S_2, \dots, S_n)$ s^ẽ block() tiến trình/luồng gọi khi có bất kỳ một đèn báo nào không thể thu được


```
 $P_{Sim}(\text{fork}[i], \text{fork}[(i+1)% 5]);$ 
{ Ăn}
 $V_{Sim}(\text{fork}[i], \text{fork}[(i+1)% 5]);$ 
```
 - Thuật toán
 - Khó cài đặt đèn báo đồng thời

Solution 6: D^úng c^{óng} c^ú Sync cap^o cao monitor