



Đồng bộ hóa

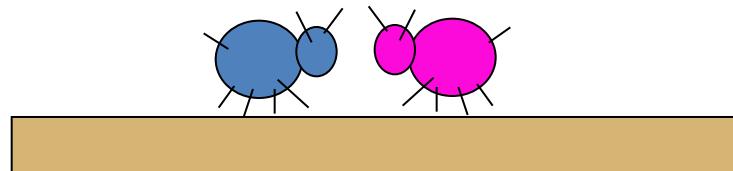
Môn học: Hệ điều hành

Noái dung

- Xöû lyù ñoàng haønh vaø caùc vaán ñeà:
 - Vaán ñeà tranh ñoaít ñieàu khieân (Race Condition)
 - Vaán ñeà phoái hôïp xöû lyù
- Baøi toaùn ñoàng boä hoùa
 - Yeâu caàu ñoäc quyeàn truy xuaát (Mutual Exclusion)
 - Yeâu caàu phoái hôïp xöû lyù (Synchronization)
- Caùc giaûi phaùp ñoàng boä hoaù
 - Busy waiting
 - Sleep & Wakeup
- Caùc baøi toaùn ñoàng boä hoaù kinh ñieân
 - Producer – Consumer
 - Readers – Writers
 - Dinning Philosophers

Nhieàu tieán trình “chung soáng hoaø bình” trong heä thoång ?

- NÖØNG HY VOÏNG
- An toaøn khi caùc tieán trình hoaøn toaøn ñoäc laäp
 - Laøm sao coù ñööïc ??
- Thöïc teá
 - Caùc tieán trình chia seû taøi nguyeân chung (file system, CPU...)
 - Concurrent access => bugs.
 - Ví duï : Deâ con qua caàu



- Xöû lyù ñoàng haønh = ...nhöùc ñaàu



Cauc vaán ñeà

- Tranh chaáp
 - Nhieàu tieán trình truy xuaát ñoàng thôøi moät taøi nguyeân mang baûn chaát khoâng chia seû ñööïc
 - Xaûy ra vaán ñeà tranh ñoaït ñieàu khieân (Race Condition)
 - Keát quaû ?
 - Khoù bieát , thôôøeng laø ...sai 
 - Luoân luoân nguy hieåm ?
 - ...Khoâng, nhöng ñuû ñeà caân nhaéc kyõ caøng 
- Phoái hôïp
 - Caùc tieán trình khoâng bieát töông quan xöû lyù cuûa nhau ñeå ñieàu chænh hoaït ñoäng nhòp nhaøng
 - Caàn phoái hôïp xöû lyù (Rendez-vous)
 - Keát quaû : khoù bieát, khoâng baûo ñaûm aên khôùp



Noái dung baøi giaûng

- Xöû lyù ñoàng haønh vaø caùc vaán ñeà:
 - Vaán ñeà tranh ñoait ñieàu khieân (Race Condition)
 - Vaán ñeà phoái hôïp xöû lyù
- Baøi toaùn ñoàng boä hoùa
 - Yeâu caàu ñoäc quyèàn truy xuaát (Mutual Exclusion)
 - Yeâu caàu phoái hôïp xöû lyù (Synchronization)
- Caùc giaûi phaùp ñoàng boä hoaù
 - Busy waiting
 - Sleep & Wakeup
- Caùc baøi toaùn ñoàng boä hoaù kinh ñieân
 - Producer – Consumer
 - Readers – Writers
 - Dinning Philosophers

Tranh nhau nhanh (Race condition) - Ví dụ

- Neam soa ngooi vao Altavista : duong 2 threads caap nhaat biean neam **hits=> P1 va P2 chia seu biean hits**

hits = 0

P1
read hits;

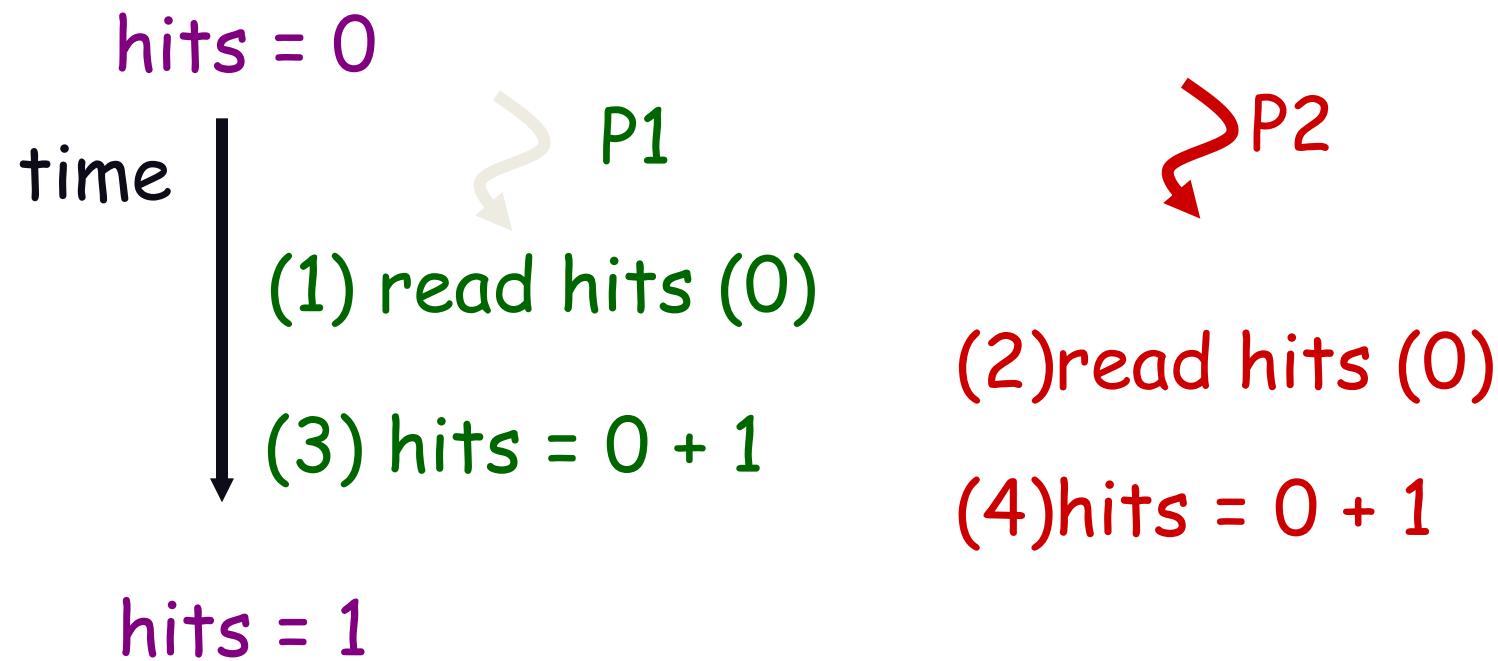
hits = hits + 1;

P2
read hits;

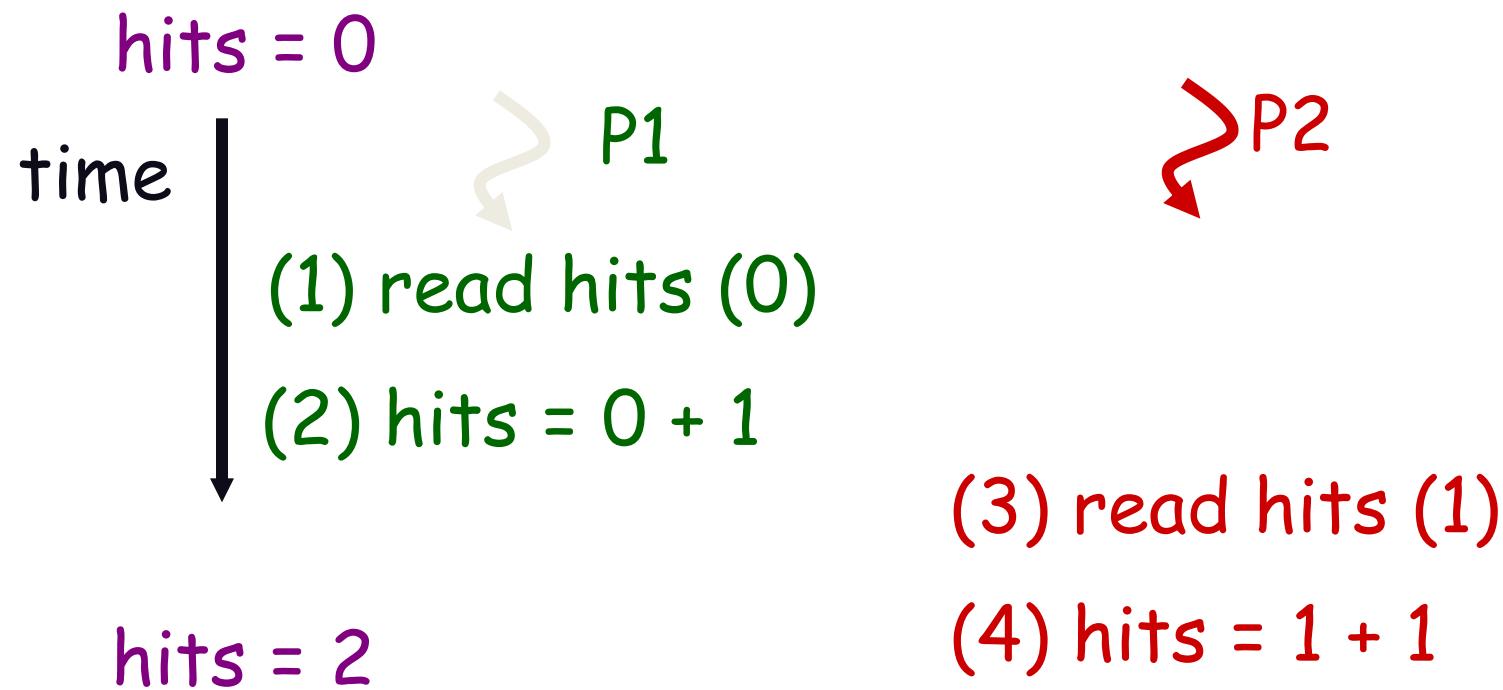
hits = hits + 1;

Keat qua cuoi cuong lao bao nhieu ?

Tranh nhau nhanh (Race condition) - Ví dụ



Tranh nhau nhanh (Race condition) - Ví dụ



Tranh nhauit nieau khiean (Race condition) - Ví dui (tt)

i=0;

Thread a:

```
while(i < 10)  
    i = i +1;
```

```
print "A won!";
```

- Ai thaéng ?

Thread b:

```
while(i > -10)  
    i = i - 1;
```

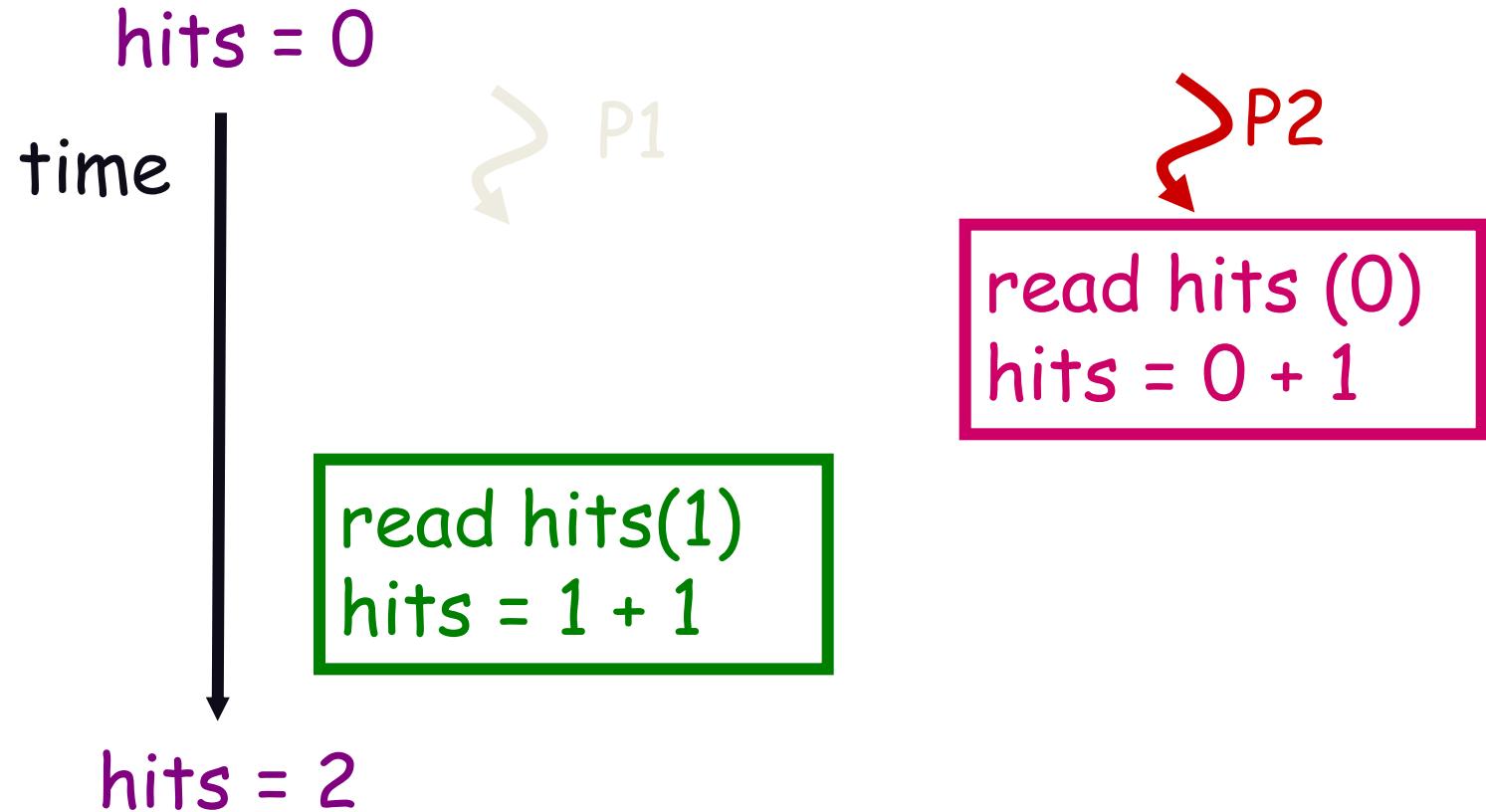
```
print "B won!";
```

- Coù baûo ñaûm raèng seõ coù ngöôoi thaéng ?
- Neáu moãi tieán trình xöû lyù treân 1 CPU thì sao ?

Tranh ñoäit ñieàu khieän (Race condition)-Nhaän xeùt

- Keát quaû thöïc hieän tieán trình phuï thuôäc vaøo keát quaû ñieàu phoái
 - Cuøng input, khoâng chaéc cuøng output
 - Khoù debug loãi sai trong xöû lyù ñoàng haønh
- Xöû lyù
 - Laøm lô
 - Deä , nhöng coù phaûi laø giaûi phaùp
 - Khoâng chia seû taøi nguyeân chung : duøng 2 bieán hits1,hits2; xaây caàu 2 lane...
 - Neân duøng khi coù theå, nhöng khoâng bao giôø coù theå ñaûm baûo ñuû taøi nguyeân, vaø cuõng khoâng laø giaûi phaùp ñuùng cho moïi tröôøng hôïp
 - Giaûi phaùp toång quaùt : coù hay khoâng ?
 - Lyù do xaûy ra Race condition ? → **Bad interleavings** : moät tieán trình “xen vaøo” quaù trình truy xuaát taøi nguyeân cuûa moät tieán trình khaùc
 - Giaûi phaùp : baûo ñaûm tính **atomicity** cho pheùp tieán trình hoaøn taát troïn veïn quaù trình truy xuaát taøi nguyeân chung tröôùc khi coù tieán trình khaùc can thieäp

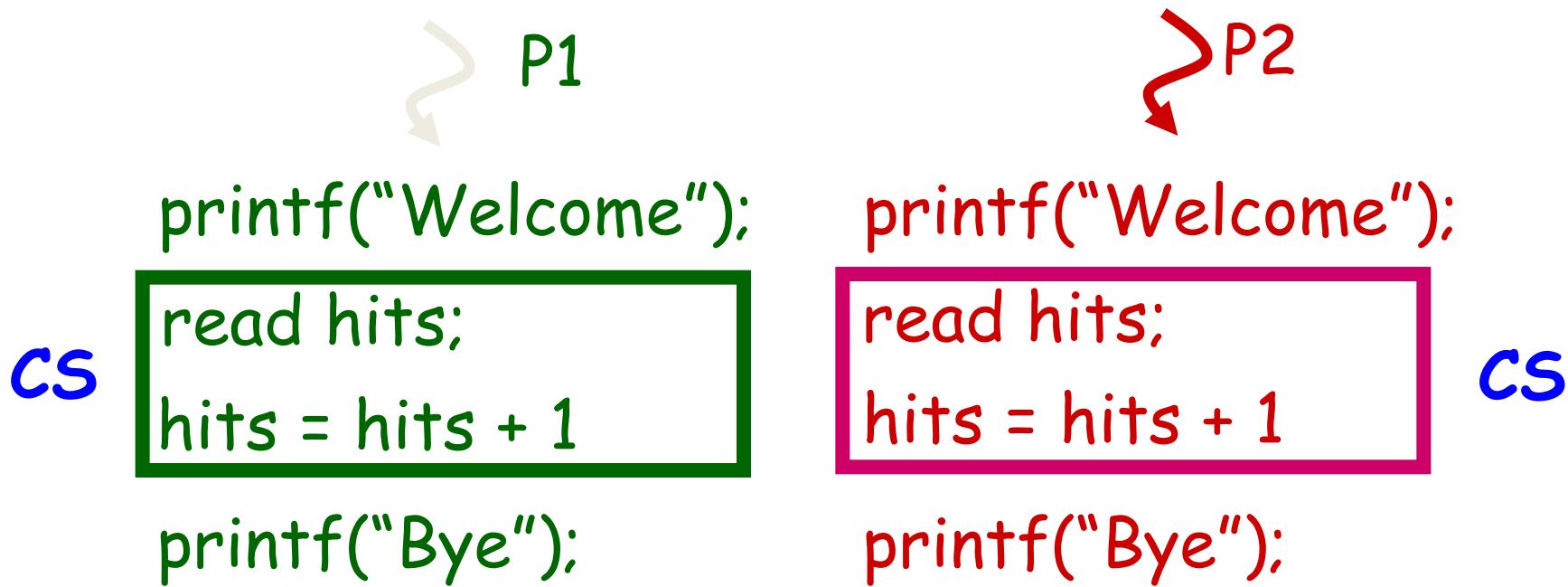
Atomicity : loại bỏ Race Condition



Mieàn gaêng (CriticalSection)

& Khaû naêng ñoäc quyèan (Mutual Exclusion)

- Mieàn gaêng (CS) laø ñoaiñ chöông trình coù khaû naêng gaây ra hieän töôïng race condition

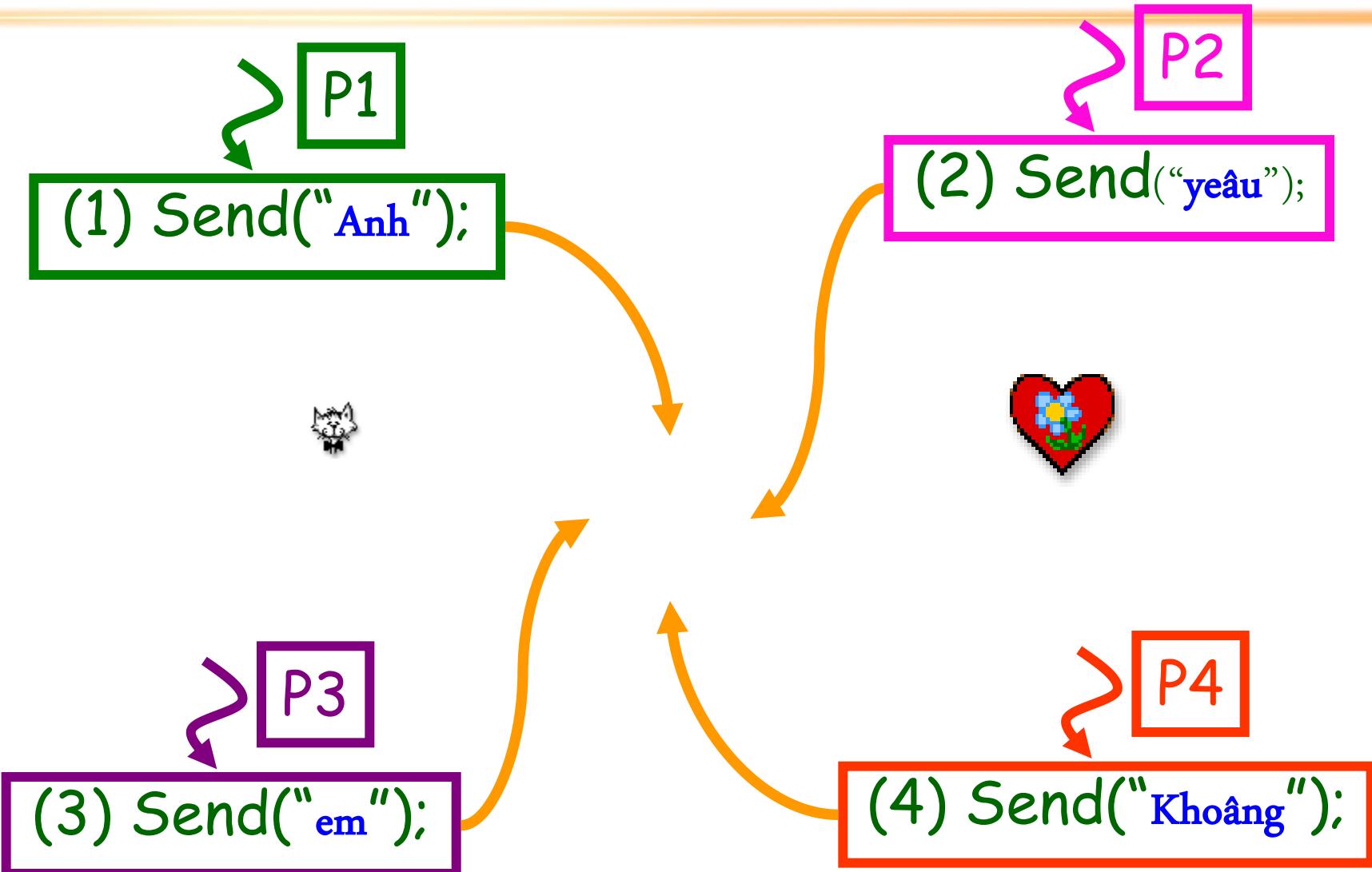


- ☞ Hoã trôï Atomicity : Caàn baûo ñaûm tính “ñoäc quyèan truy xuaát” (**Mutual Exclusion**) cho mieàn gaêng (CS)

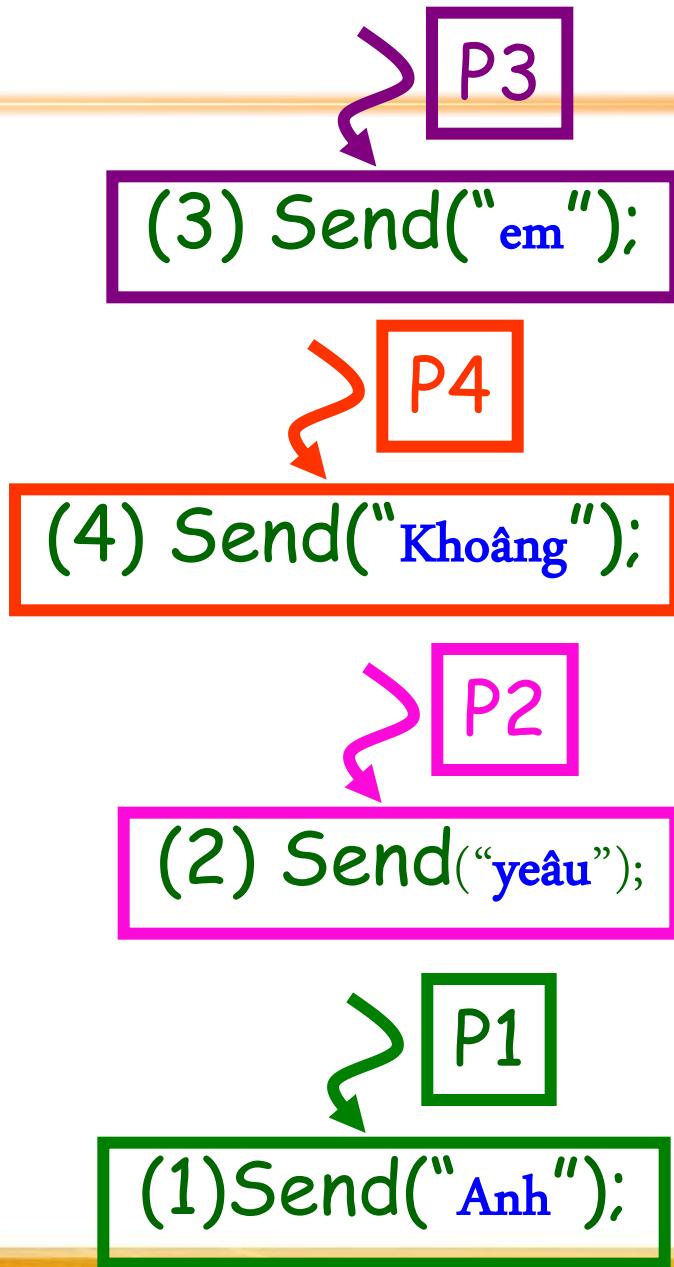
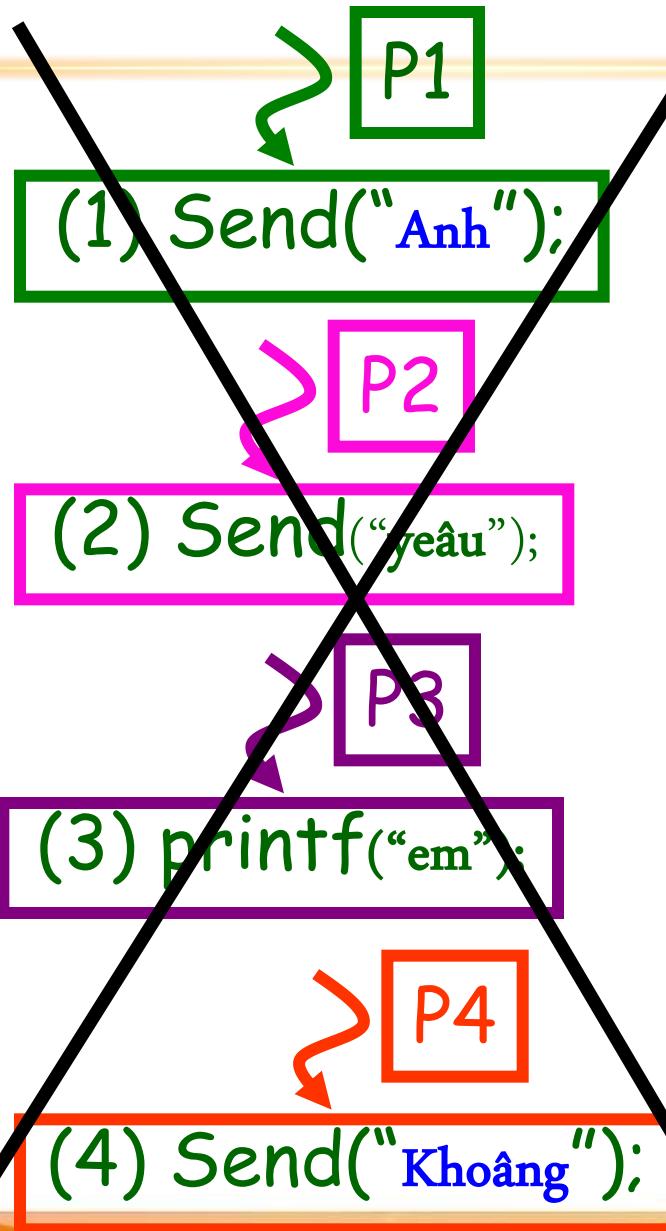
Noái dung baøi giaûng

- Xöû lyù ñoàng haønh vaø caùc vaán ñeà:
 - Vaán ñeà tranh ñoaít ñieàu khieân (Race Condition)
 - Vaán ñeà phoái hôïp xöû lyù
- Baøi toaùn ñoàng boä hoùa
 - Yeâu caàu ñoäc quyèàn truy xuaát (Mutual Exclusion)
 - Yeâu caàu phoái hôïp xöû lyù (Synchronization)
- Caùc giaûi phaùp ñoàng boä hoaù
 - Busy waiting
 - Sleep & Wakeup
- Caùc baøi toaùn ñoàng boä hoaù kinh ñieân
 - Producer – Consumer
 - Readers – Writers
 - Dinning Philosophers

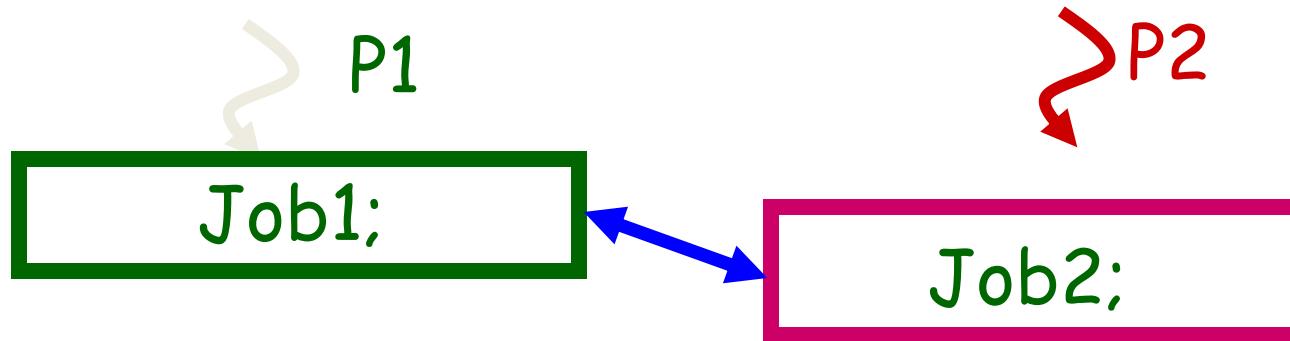
Phoái hôip hoait ñoäng



Chuyễn gì nǎo xuây ra ?



Phoái hôip xôû lyù



- Laøm theá naøo baûo ñaûm trình töï thöïc hieän Job1 - Job2 ?
 - P1 vaø P2 thöïc hieän “heïn hoø” ([Rendez-vous](#)) vôùi nhau
- Hoã trôï Rendez-vous : Baûo ñaûm caùc tieán trình phoái hôip vôùi nhau theo 1 trình töï xôû lyù ñònh tröôùc.

Noái dung baøi giaûng

- Xöû lyù ñoàng haønh vaø caùc vaán ñeà:
 - Vaán ñeà tranh ñoäit ñieàu khieân (Race Condition)
 - Vaán ñeà phoái hôïp xöû lyù
- Baøi toaùn ñoàng boä hoà
 - Yeâu caàu ñoäc quyèàn truy xuaát (Mutual Exclusion)
 - Yeâu caàu phoái hôïp xöû lyù (Synchronization)
- Caùc giaûi phaùp ñoàng boä hoaù
 - Busy waiting
 - Sleep & Wakeup
- Caùc baøi toaùn ñoàng boä hoaù kinh ñieân
 - Producer – Consumer
 - Readers – Writers
 - Dinning Philosophers

Baøi toaùn ñoàng boä hoaù (Synchronization)

- Nhieàu tieán trình chia seû taøi nguyeân chung ñoàng thôøi :
 - Tranh chaáp \Rightarrow Race Condition
 - Nhu caùu “ñoäc quyèän truy xuaát” (Mutual Exclusion)
- Caùc tieán trình phoái hôïp hoaït ñoäng :
 - Töông quan dieän tieán xöû lyù ?
 - Nhu caùu “hoø heïn” (Rendez-vous)
- Thöïc hieän ñoàng boä hoaù :
 - Laäp trình vieân ñeà xuaát chieán lööïc
 - Caùc tieán trình lieân quan trong baøi toaùn phaûi toân troïng caùc lúaätñoàng boä
 - Giaûi phaùp söû duïng caùc **cô cheá** ñoàng boä :
 - Do laäp trình vieân /phaàn cõùng / HÑH / NNLT cung caáp

Moâ hình ñatûm baô Mutual Exclusion

- Nhieäm vuï cuâa laäp trình vieân:
 - Theâm caùc ñoäin code ñoàng boä hoà vaøo chöông trình goác
 - Theâm theá naøo : xem moâ hình sau ...

Kieäm tra vaø daønh quyèàn vaøo CS

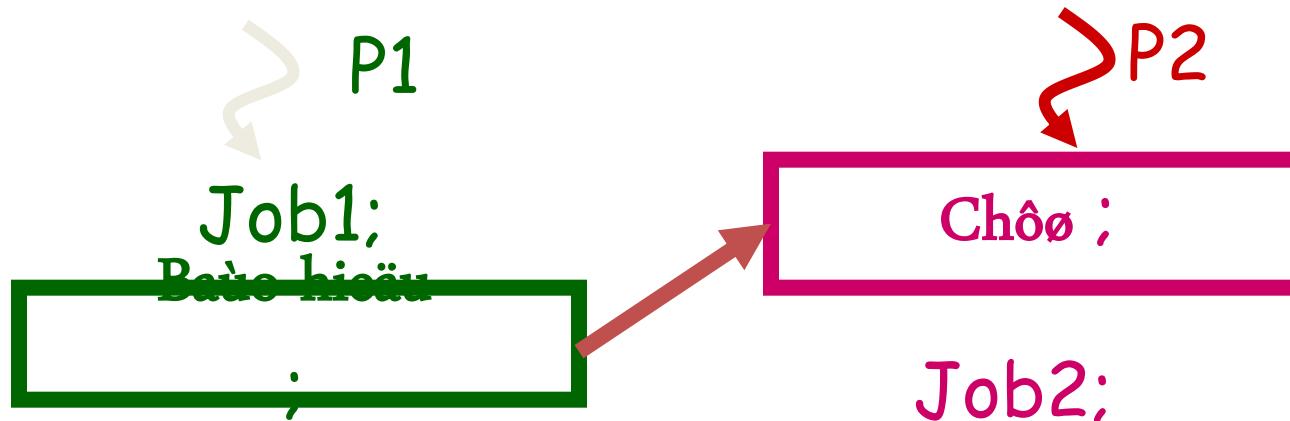
CS;

Töø boû quyèàn söû duïng CS

Moâ hình toå chöùc phoái hôip giöõa hai tieán trình

- Nhieäm vuïi cuâa laäp trình vieân:

- Theâm caùc ñoaïn code ñoàng boä hoùa vaøo 2 chööng trình goác
- Theâm theá naøo : xem moâ hình sau ...



- Nhieäu tieán trình hôn thì sao ?

- Khoâng coù moâ hình toång quaüt
- Tuøy thuoaäc baïn muoán heïn hoø ra sao ☺

Noái dung baøi giaûng

- Xöû lyù ñoàng haønh vaø caùc vaán ñeà:
 - Vaán ñeà tranh ñoaít ñieàu khieân (Race Condition)
 - Vaán ñeà phoái hôïp xöû lyù
- Baøi toaùn ñoàng boä hoùa
 - Yeâu caàu ñoäc quyèàn truy xuaát (Mutual Exclusion)
 - Yeâu caàu phoái hôïp xöû lyù (Synchronization)
- Caùc giaûi phaùp ñoàng boä hoaù
 - Busy waiting
 - Sleep & Wakeup
- Caùc baøi toaùn ñoàng boä hoaù kinh ñieân
 - Producer – Consumer
 - Readers – Writers
 - Dinning Philosophers

Giaûi phaùp ñoàng boä hoaù

Moät phöông phaùp giaûi quyéát toát baøi toaùn ñoàng boä hoaù caàn thaôu maûn 4 ñieàu kieän sau:

- **Mutual Exclusion** : Khoâng coù hai tieán trình cuøng ôû trong mieàn gaêng cuøng luùc.
- **Progess** : Moät tieán trình taïm döøng beân ngoaøi mieàn gaêng khoâng ñööïc ngaên caûn caùc tieán trình khaùc vaøo mieàn gaêng
- **Bounded Waiting** : Khoâng coù tieán trình naøo phaûi chôø voâ haïn ñeå ñööïc vaøo mieàn gaêng.
- Khoâng coù giaûi thieát naøo ñaët ra cho söi lieân heä veà toác ñoä cuûa caùc tieán trình, cuõng nhö veà soá löôïng boä xöû lyù trong heä thoáng.

Caùc giaûi phaùp ñoàng boä hoaù

- Nhoùm giaûi phaùp Busy Waiting
 - Phaàn meàm
 - Söû duïng caùc bieán côø hieäu
 - Söû duïng vieäc kieäm tra luaân phieân
 - Giaûi phaùp cuâa Peterson
 - Phaàn cöùng
 - Caám ngaét
 - Chæ thò TSL
- Nhoùm giaûi phaùp Sleep & Wakeup
 - Semaphore
 - Monitor
 - Message

Cáùc giatíi phaùp “Busy waiting”

While (chöa coù quyèàn) donothing() ;

CS;

Töø boû quyèàn söû duïng CS

- Tieáp tuïc tieâu thuï CPU trong khi chôø ñôïi vaøo mieàn gaêng
- Khoâng ñoøi hoûi söi trôï giuùp cuâa Heä ñieàu haønh

Nhóùm giaûi phaùp Busy-Waiting

- Caùc giaûi phaùp Busy Waiting
 - Caùc giaûi phaùp phaàn meàm
 - Giaûi phaùp bieán côø hieäu
 - Giaûi phaùp kieâm tra luaân phieân
 - Giaûi phaùp Peterson
 - Phaàn cöùng
 - Caám ngaét
 - Chæ thò TSL

Giaûi phaùp phaàn meàm 1: Söù duïng bieán côø hieäu

int lock = 0

P0

NonCS;

```
while (lock == 1); // wait  
lock = 1;
```

CS;

```
lock = 0;
```

NonCS;

P1

NonCS;

```
while (lock == 1); // wait  
lock = 1;
```

CS;

```
lock = 0;
```

NonCS;

Giaiđi phâup phâan meàm 1: Tình huoáng

int lock = 0

P0

NonCS;

```
while (lock == 1); // wait  
lock = 1;
```

CS;

```
lock = 0;
```

NonCS;

P1

NonCS;

```
while (lock == 1); // wait  
lock = 1;
```

CS;

```
lock = 0;
```

NonCS;

Nhaän xeùt Giaûi phaùp phaàn meàm 1: Bieán côø hieäu

- Coù theå môû roäng cho N tieán trình
- Khoâng baûo ñaûm Mutual Exclusion
 - Nguyêân nhaân ?

Bò ngaét xöû lyù

```
while ( lock == 1); // wait  
lock = 1;
```

CS !

Taøi nguyeân duøng chung

- Baûn thaân ñoaïn code kieäm tra vaø daønh quyèan cuõng laø CS !

Giaûi phaùp phaàn meàm 2 : Kieâm tra luaân phieân

int turn = 1

PO

NonCS;

while (turn !=0); // wait

CS;

turn = 1;

NonCS;

P1

NonCS;

while (turn != 1); // wait

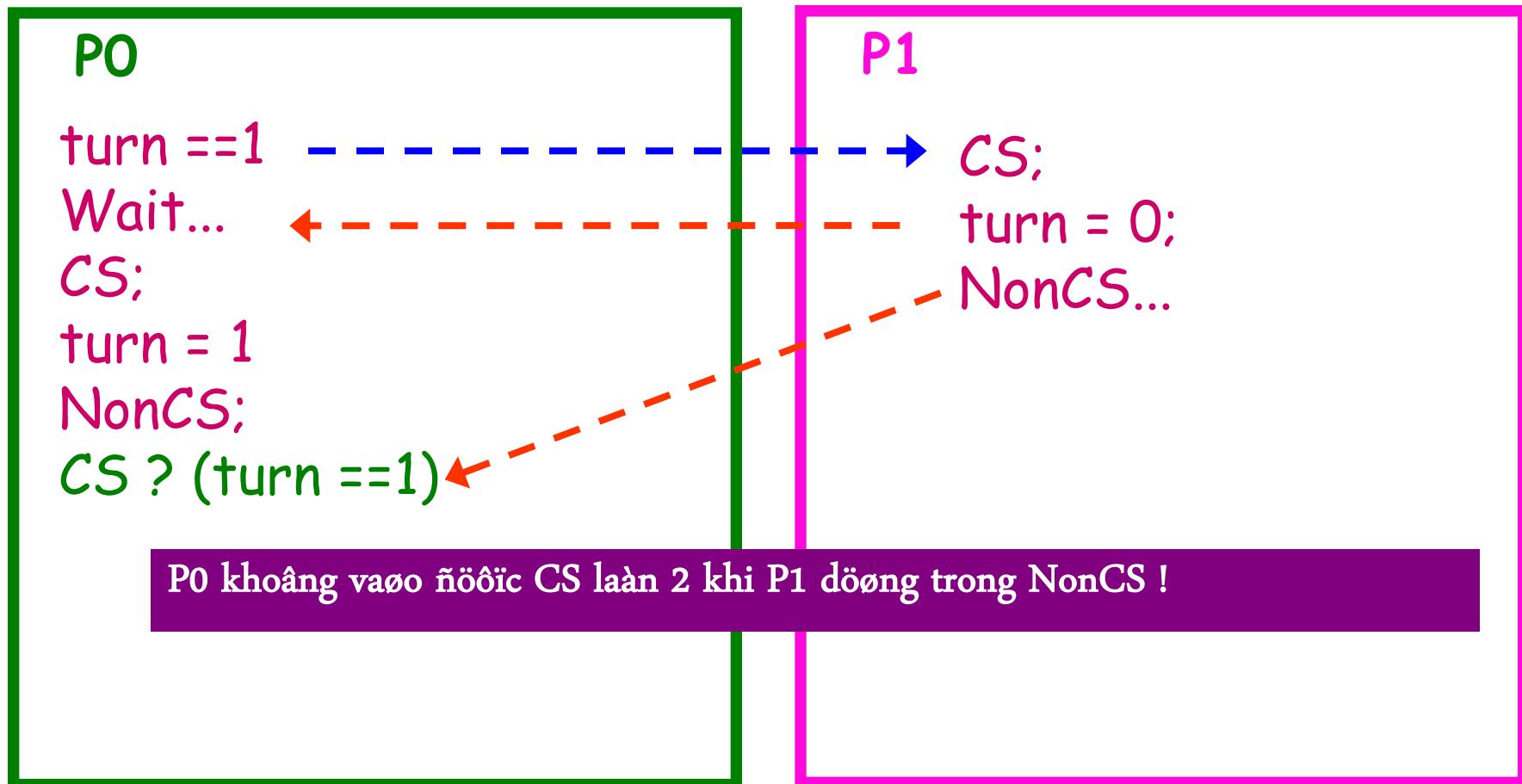
CS;

turn = 0;

NonCS;

Giaiđi phaüp phaân meàm 2 : Tình huoáng

int turn = 1



Nhaän xeùt Giaûi phaùp 2: Kieâm tra luaân phieân

- Chæ daønh cho 2 tieán trìnħ
- Baûo ñaûm Mutual Exclusion
 - Chæ coù 1 bieán *turn*, taïi 1 thôøi ñieäm chæ cho 1 tieán trìnħ *turn* vaøo CS
- Khoâng baûo ñaûm Progress
 - Guyeân nhaân ?
 - “Môø cuûa” cho ngöôøi = “Ñoùng cöûa” chính mình !

Giaûi phaùp phaàn meàm 3 : Peterson's Solution

- Keát hôïp yù töôûng cuâa 1 & 2, caùc tieán trình chia seû:
 - int turn; *// ñeán phieân ai*
 - int interest[2] = FALSE; *// interest[i] = T : Pi muoán vaøo CS*

Pi

NonCS;

```
j = 1 - i;  
interest[i] = TRUE;  
turn = j;  
while (turn==j && interest[j]==TRUE);
```

CS;

```
interest[i] = FALSE;
```

NonCS;

Giai phap phan meam 3 : Peterson

Pj

NonCS;

```
i = 1 - j;  
interest[j] = TRUE;  
turn = i;  
while (turn==i && interest[i]==TRUE);
```

CS;

```
interest[j] = FALSE;
```

NonCS;

Nhaän xeüt giaûi phaùp phaàn meàm 3: Peterson

- Laø giaûi phaùp phaàn meàm ñaùp öùng ñööïc caû 3 ñieàu kieän
 - Mutual Exclusion :
 - Pi chæ coù theå vaøo CS khi: *interest[j] == F* hay *turn == i*
 - Neáu caû 2 muoán veà thi do *turn* chæ coù theå nhaän giaù trò 0 hay 1 neân chæ coù 1 tieán trình vaøo CS
 - Progress
 - Söû duïng 2 bieán *interest[i]* rieâng bieät => traïng thaùi ñoái phööong khoâng khoaù mìnhanh ñööïc
 - Bounded Wait : *interest[i]* vaø *turn* ñeàu coù thay ñoái giaù trò
- Khoâng theå môû roäng cho N tieán trình

Nhaän xeùt chung veà caùc giaûi phaùp phaàn meàm trong nhoùm Busy-Waiting

- Khoâng caàn söi hoã trôï cuûa heä thoáng
- Deã...sai, Khouù môû roäng
- Giaûi phaùp 1 neáu coù theå ñööïc hoã trôï **atomicity** thì seõ toát...
 - Nhôø ñeán phaàn cöùng ?



Nhòum Busy-Waiting - Caùc giaûi phaùp phaàn cõùng

- Caùc giaûi phaùp Busy Waiting
 - Caùc giaûi phaùp phaàn meàm
 - Giaûi phaùp bieán côø hieäu
 - Giaûi phaùp kieäm tra luaân phieân
 - Giaûi phaùp Peterson
 - Caùc giaûi phaùp phaàn cõùng
 - Caám ngaét
 - Test&Set lock Instruction

Nhóm Busy-Waiting - Giaiđi phaùp phaàn cõùng 1: Caám ngaét

NonCS;

Disable Interrupt;

CS;

Enable Interrupt;

NonCS;

- **Disable Interrupt** : Caám moïi ngaét, keâ caû ngaét ñoàng hoà
- **Enable Interrupt** : Cho pheùp ngaét

Giaûi phaùp phaàn cõùng 1: Caám ngaét

- Thieáu thaän troïng
 - Neáu tieán trình bò khoaù trong CS ?
 - System Halt
 - Cho pheùp tieán trình söû duïng moät leanh ñaëc quyèan
 - Quaù ...lieàu !
- Maùy coù N CPUs ?
 - Khoâng baûo ñaûm ñöôïc Mutual Exclusion

Nhóm Busy-Waiting - Giải pháp phàn công 2: chẽ thò TSL()

- CPU hoã tröi primitive Test and Set Lock
 - Traù veà giaù trò hieän haønh cuâa 1 bieán, vaø ñaët laïi giaù trò True cho bieán
 - Thöíc hieän moät caùch khoâng theå phaân chia

TSL (boolean &target)

{

TSL = target;
target = TRUE;

}

Aùp dùng TSL

int lock = 0

Pi

NonCS;

while (TSL(lock)); // wait

CS;

lock = 0;

NonCS;

Nhaän xeùt chung caùc giaûi phaüp phaàn cöùng trong nhoùm Busy-Waiting

- Caàn nööïc söïi hoã trôïi cuâa cô cheá phaàn cöùng
 - Khoâng deã, nhaát laø treân caùc maùy coù nhieàu boä xöû lyù
- Deã môû roäng cho N tieán trình

Nhaän xeùt chung cho caùc giaûi phaùp trong nhoùm Busy

Waiting

- Söû duïng CPU khoâng hieäu quaû
 - Lieân tuïc kieäm tra ñieàu kieän khi chôø vaøo CS
- Khaéc phuïc
 - Khoaù caùc tieán trình chöa ñuû ñieàu kieän vaøo CS, nhöôøng CPU cho tieán trình khaùc
 - Phaûi nhôø ñeán Scheduler
 - Wait and See...

Caùc giaûi phaùp ñoàng boä hoaù

- Nhoùm giaûi phaùp Busy Waiting
 - Phaàn meàm
 - Söû duïng caùc bieán côø hieäu
 - Söû duïng vieäc kieäm tra luaân phieân
 - Giaûi phaùp cuâa Peterson
 - Phaàn cöùng
 - Caám ngaét
 - Chæ thò TSL
- Nhoùm giaûi phaùp Sleep & Wakeup
 - Semaphore
 - Monitor
 - Message

Cáùc giatri phaùp “Sleep & Wake up”

if (chöa coù quyèan) Sleep() ;

CS;

Wakeup(somebody);

- Töø boû CPU khi chöa ñöôïc vaøo CS
- Khi CS troáng, seõ ñöôïc ñaùnh thöùc ñeå vaøo CS
- Caàn ñöôïc Heä ñieàu haønh hoã tröï
 - Vì phaùi thay ñoái traëng thaùi tieán trình

YÙ töôüng

- Heä Ñieàu haønh hoã trôï 2 primitive :
 - Sleep() : Tieán trình goïi seõ nhaän traïng thaùi Blocked
 - WakeUp(P): Tieán trình P nhaän traïng thaùi Ready
- AÙp duïng
 - Sau khi kieäm tra ñieàu kieän seõ vaøo CS hay goïi Sleep() tuøy vaøo keát quaû kieäm tra
 - Tieán trình vöøa söû duïng xong CS seõ ñaùnh thöùc caùc tieán trình bò Blocked tröôùc ñoù

AUp during Sleep() and Wakeup()

- int busy; // busy ==0 : CS troáng
- int blocked; // ñeám soá tieán trình bò Blocked chôø vaøo CS

```
if (busy) {  
    blocked = blocked + 1;  
    Sleep();  
}  
else busy = 1;
```

CS:

```
busy = 0;  
if(blocked) {  
    WakeUp(P);  
    blocked = blocked - 1;  
}
```

Vaán ñeà vòùi Sleep & WakeUp

P1

```
if (busy) {  
    blocked = blocked + 1;  
    Sleep();  
}  
else busy = 1;
```

P2

```
if (busy) {  
    blocked = blocked + 1;  
    Sleep();  
}  
else busy = 1;
```

CS;

WakeUp
bò “laïc”

CS;

```
busy = 0;  
if(blocked) {  
    WakeUp(P);  
    blocked = blocked - 1;  
}
```

```
busy = 0;  
if(blocked) {  
    WakeUp(P);  
    blocked = blocked - 1;  
}
```

P1 blocked
vónh vieñ

■ Nguyêân nhaân :

- Vieäc kieäm tra ñieàu kieän vaø ñoäng taùc töø boû CPU coù theå bò ngaét quaõng
- Baûn thaân caùc bieán côø hieäu khoâng ñööïc baûo veä

Caøi ñaët caùc giaùi phaùp Sleep & WakeUp ?

- Heä ñieàu haønh caàn hoã trôi caùc cô cheá cao hôñ
 - Döïa treân Sleep&WakeUp
 - Keát hôïp caùc yeáu toá kieâm tra
 - Thi haønh khoâng theå phaân chia
- Nhoùm giaùi phaùp Sleep & Wakeup
 - Semaphore
 - Monitor
 - Message

Giaûi phaùp Sleep & Wakeup 1: Semaphore

- Nööïc ñeà nghò bôûi Dijkstra naêm 1965
- Caùc ñaëc tính : **Semaphore s;**
 - Coù 1 giaù trò
 - Chæ ñööïc thao taùc bôûi 2 primitives :
 - Down(s)
 - Up(s)
 - Caùc primitive Down vaø Up ñööïc thöïc hieän khoâng theå phaân chia

Semaphore s; // s >=0

Caøi ñaët Semaphore (Sleep & Wakeup)

```
typedef struct  
{  
    int value;  
    struct process* L;  
} Semaphore ;
```

Giaù trò beân trong cuûa semaphore

Danh saùch caùc tieán trình ñang bò
block ñôïi semaphore nhaän giaù trò
döông

- Semaphore ñöôïc xem nhö laø moät resource
 - Caùc tieán trình “yeâu caàu” semaphore : goïi Down(s)
 - Neáu khoâng hoaøn taát ñöôïc Down(s) : chöa ñöôïc caáp resource
 - Blocked, ñöôïc ñöa vaøo s.L
- Caàn coù söï hoã trôï cuûa HÑH
 - Sleep() & Wakeup()

Caøi ñaët Semaphore (Sleep & Wakeup)

Down (S)

```
{  
    S.value --;  
    if S.value < 0  
    {  
        Add(P, S.L);  
        Sleep();  
    }  
}
```

Up(S)

```
{  
    S.value ++;  
    if S.value ≤ 0  
    {  
        Remove(P, S.L);  
        Wakeup(P);  
    }  
}
```

Söù duïng Semaphore

- Toå chöùc “ñoäc quyèan truy xuaát”

Semaphore $s = 1$

P_i

Down (s)
CS;
Up(s)

- Toå chöùc “hoø heïn”

Semaphore $s = 0$

$P_1 :$
Job1;
Up(s)



$P_2 :$
Down (s);
Job2;

Nhaän xeüt Semaphores

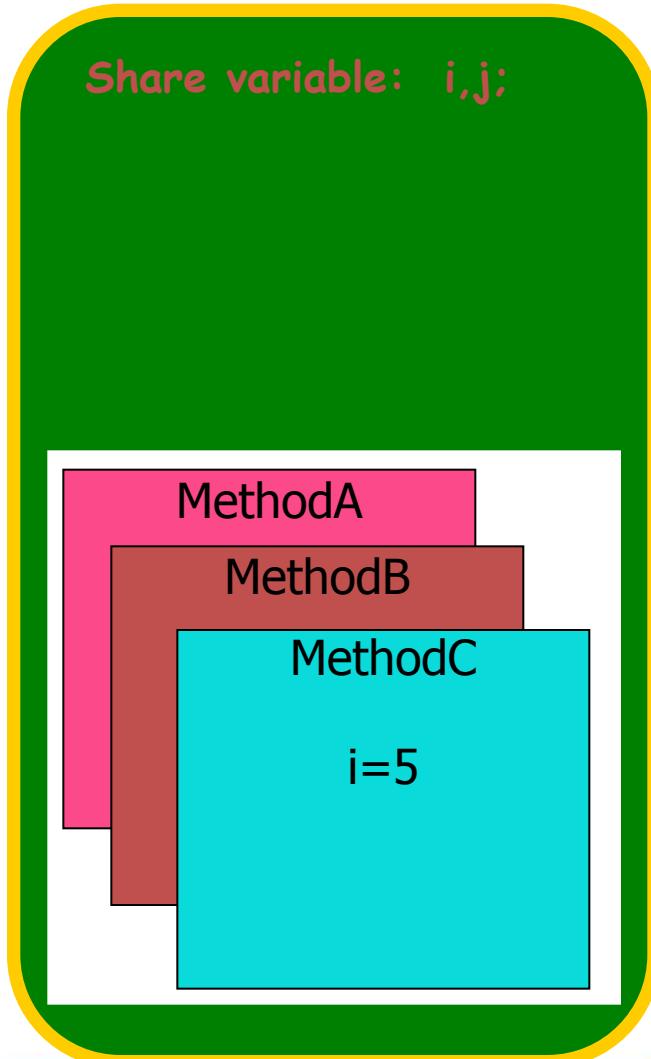
- Laø moät cô cheá toát ñeå thöïc hieän ñoàng boä
 - Deã duøng cho N tieán trình
- Nhöng yù nghóa söû duïng khoâng roõ raøng
 - MutualExclusion : Down & Up
 - Rendez-vous : Down & Up
 - Chæ phaân bieät qua moâ hình
- Khoù söû duïng ñuùng
 - Nhaàm lañn

Giaûi phaùp Sleep & Wakeup 2: Monitor

- Ñeà xuaát bôûi Hoare(1974) & Brinch (1975)
- Laø cô cheá ñoàng boä hoaù do NNLT cung caáp
 - Hoã trôïi cuøng caùc chöùc naêng nhö Semaphore
 - Deã söû duïng vaø kieåm soaùt hôn Semaphore
 - Baûo ñaûm Mutual Exclusion moät caùch töï ñoäng
 - Söû duïng bieán ñieàu kieän ñeå thöïc hieän Synchronization

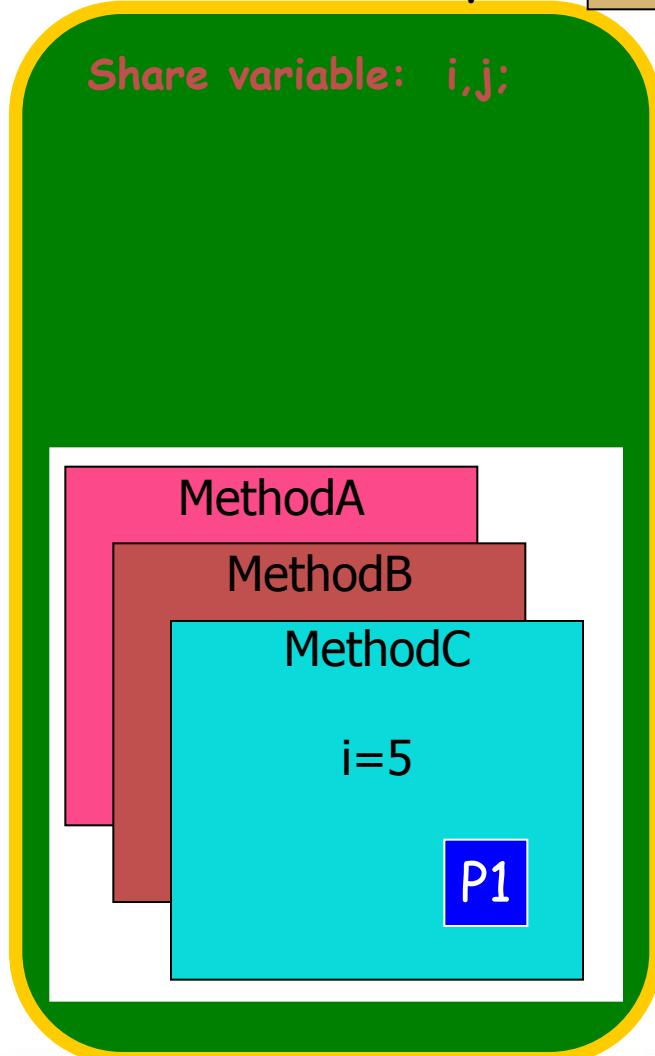
Monitor : Ngõõ nghóá vaø tính chaát(1)

Monitor M



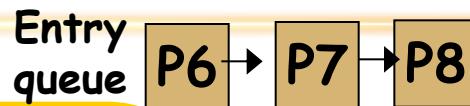
- Làø moät module chöông trình nòngh nghóá
 - Caùc CTDL, ñoái tööïng duøng chung
 - Caùc phöông thöùc xöû lyù caùc ñoái tööïng naøy
 - Baûo ñaûm tính encapsulation
- Caùc tieán trình muoán truy xuaát döõ lieäu beân trong monitor phaûi duøng caùc phöông thöùc cuûa monitor :
 - P1 : M.C() // i=5
 - P2: M.B() // printf(j)

Monitor : Ngõõ nghóá vaø tính chaát(2)



- Töi ñoäng baûo ñaûm Mutual Exclusion
 - Taïi 1 thôøi ñieåm chæ coù 1 tieán trình ñööic thöic hieän caùc phöông thöùc cuâa Monitor
 - Caùc tieán trình khoâng theå vaøo Monitor seõ ñööic ñõa vaøo Entry queue cuâa Monitor
- Ví duï
 - P1 : M.A();
 - P6 : M.B();
 - P7 : M.A();
 - P8 : M.C();

Monitor : Ngõõ nghóá vaø tính chaát(3)



Share variable: i,j;

Condition variable:



MethodA

MethodB

P1

MethodC

wait(C1);
i=5

signal(C2);

- Hoã trôï Synchronization vòùi caùc **condition variables**
 - Wait(c) : Tieán trình goïi haøm seõ bò blocked
 - Signal(c): Giaûi phoùng 1 tieán trình ñang bò blocked treân bieán ñieàu kieän c
 - C.queue : danh saùch caùc tieán trình blocked treân c
- Traïng thaùi tieán trình sau khi goïi Signal?
 - Blocked. Nhöôøng quyèàn vaøo monitor cho tieán trình ñöôïc ñaùnh thöùc
 - Tieáp tuïc xöû lyù heát chu kyø, roài blocked

Söù duïng Monitor

- Toå chöùc “ñoäc quyèan truy xuaát”

Monitor M
<resource type> RC ;
Function $AccessMutual$
 $CS; // access RC$

P_i
 $M.AccessMutual(); //CS$

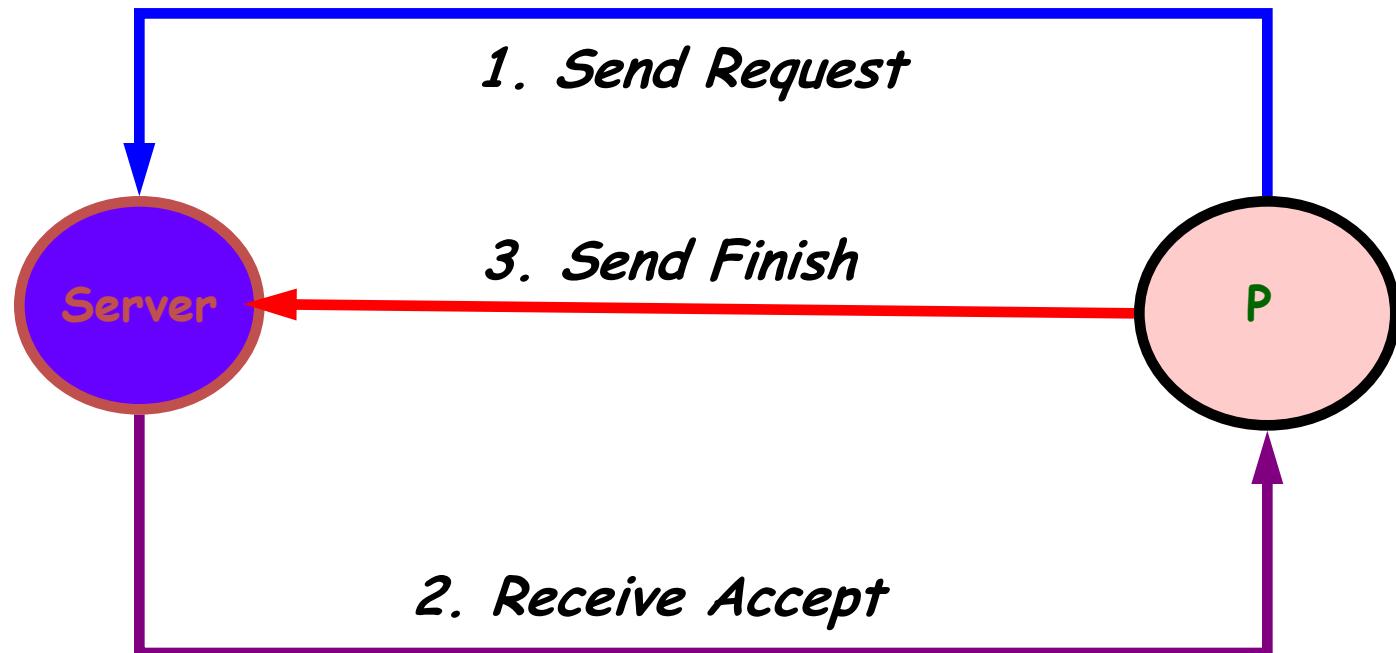
- Toå chöùc “hoø heïn”

Monitor M
Condition c ;
Function $F1$
 Job1;
 $Signal(c);$
Function $F2$
 $Wait(c);$
 Job2;

$P_1 : M.F1(); \rightarrow P_2 : M.F2();$

Giai phaüp Sleep & Wakeup 3: Message

- Nööic hoã trôi bôûi HÑH
- Nøång boä hoùa treân moâi tröôøng phaân taùn
- 2 primitive Send & Receive
 - Caøi ñaët theo mode blocking



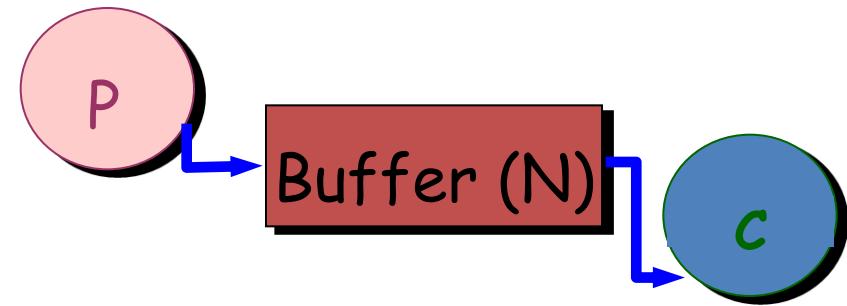
Noái dung baøi giaûng

- Xöû lyù ñoàng haønh vaø caùc vaán ñeà:
 - Vaán ñeà tranh ñoaít ñieàu khieân (Race Condition)
 - Vaán ñeà phoái hôïp xöû lyù
- Baøi toaùn ñoàng boä hoùa
 - Yeâu caàu ñoäc quyèàn truy xuaát (Mutual Exclusion)
 - Yeâu caàu phoái hôïp xöû lyù (Synchronization)
- Caùc giaûi phaùp ñoàng boä hoaù
 - Busy waiting
 - Sleep & Wakeup
- Caùc baøi toaùn ñoàng boä hoaù kinh ñieân
 - Producer – Consumer
 - Readers – Writers
 - Dinning Philosophers

Baøi toaùn ñoàng boä kinh ñieån 1:

Producer - Consumer (Bounded-Buffer Problem)

- Moâ taû : 2 tieán trình P vaø C hoaït ñoàng ñoàng haønh
 - P saûn xuaát haøng vaø ñaët vaø Buffer
 - C laáy haøng töø Buffer nîi tieâu thuï
 - Buffer coù kích thöôùc giôùi haïn
- Tình huoáng
 - P vaø C ñoàng thôøi truy caäp Buffer ?
 - P theâm haøng vaø Buffer ñaày ?
 - C laáy haøng töø Buffer troång ?



- P khoâng ñöôïc ghi döõ lieäu vaøo buffer ñaõ ñaày (Rendez-vous)
- C khoâng ñöôïc ñoïc döõ lieäu töø buffer ñang troång (Rendez-vous)
- P vaø C khoâng ñöôïc thao taùc treân buffer cuøng luùc (Mutual Exclusion)

Producer – Consummer : Giaûi phaùp Semaphore

- Caùc bieán duøng chung giöõa P vaø C
 - bufferSize = N; // soá choã trong boä ñeäm
 - semaphore mutex = 1 ; // kieäm soaùt truy xuaát ñoäc quyèan
 - semaphore empty = bufferSize; // soá choã troång
 - semaphore full = 0; // soá choã ñaày
 - int Buffer[BufferSize]; // boä ñeäm duøng chung

Producer – Consumer : Giải pháp Semaphore

Producer()

```
{  
    int item;  
    while (TRUE)  
    {  
        produce_item(&item);  
        down(&empty);  
        down(&mutex)  
        enter_item(item,Buffer);  
        up(&mutex);  
        up(&full);  
    }  
}
```

Consumer()

```
{  
    int item;  
    while (TRUE)  
    {  
        down(&full);  
        down(&mutex);  
        remove_item(&item,Buffer);  
        up(&mutex);  
        up(&empty);  
        consume_item(item);  
    }  
}
```

P&C - Giải pháp Semaphore: Thinking...

Producer()

```
{  
    int item;  
    while (TRUE)  
    {  
        produce_item(&item);  
        down(&mutex)  
        down(&empty);  
        enter_item(item,Buffer);  
        up(&mutex);  
        up(&full);  
    }  
}
```

Consumer()

```
{  
    int item;  
    while (TRUE)  
    {  
        down(&mutex);  
        down(&full);  
        remove_item(&item,Buffer);  
        up(&mutex);  
        up(&empty);  
        consume_item(item);  
    }  
}
```

Producer – Consumer : Giải pháp Monitor

```
monitor ProducerConsumer  
condition full, empty;  
int Buffer[N], count;  
procedure enter();  
{  
    if (count == N)  
        wait(full);  
    enter_item(item,Buffer);  
    count ++;  
    if (count == 1)  
        signal(empty);  
}
```

```
procedure remove();  
{  
    if (count == 0)  
        wait(empty);  
    remove_item(&item,Buffer);  
    count --;  
    if (count == N-1)  
        signal(full);  
}  
count = 0;  
end monitor;
```

Producer – Consumer : Giải pháp Monitor

Producer()

```
{  
    int item;  
    while (TRUE)  
    {  
        produce_item(&item);  
        ProducerConsumer.enter;  
    }  
}
```

Consumer():

```
{  
    int item;  
    while (TRUE)  
    {  
        ProducerConsumer.remove;  
        consume_item(item);  
    }  
}
```

Producer – Consumer : Giải pháp Message

Producer()

```
{  
    int item;  
    message m;  
  
    while (TRUE)  
    {  
        produce_item(&item);  
        receive(consumer, Request);  
        create_message(&m, item);  
        send(consumer,&m);  
    }  
}
```

Coi chöøng
Deadlock

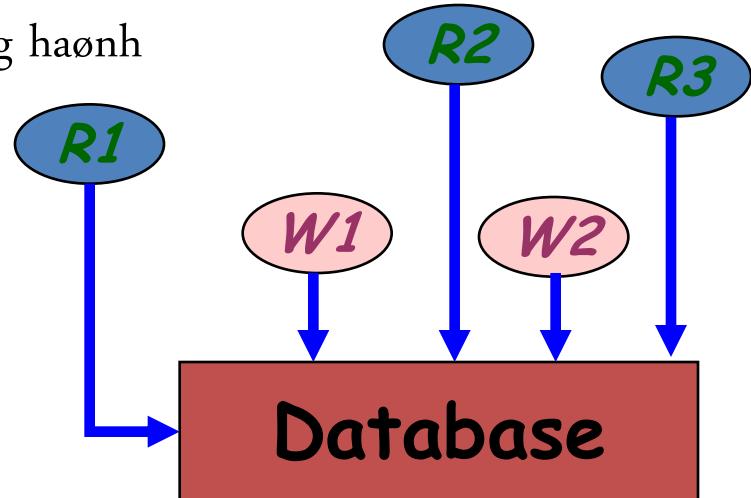
Consumer();

```
{  
    int item;  
    message m;  
    for(0 to N)  
        send(producer, Request);  
  
    while (TRUE)  
    {  
        receive(producer, &m);  
        remove_item(&m,&item);  
        send(producer, Request);  
        consumer_item(item);  
    }  
}
```

Baøi toaùn ñoàng boä hoaù kinh ñieån 2:

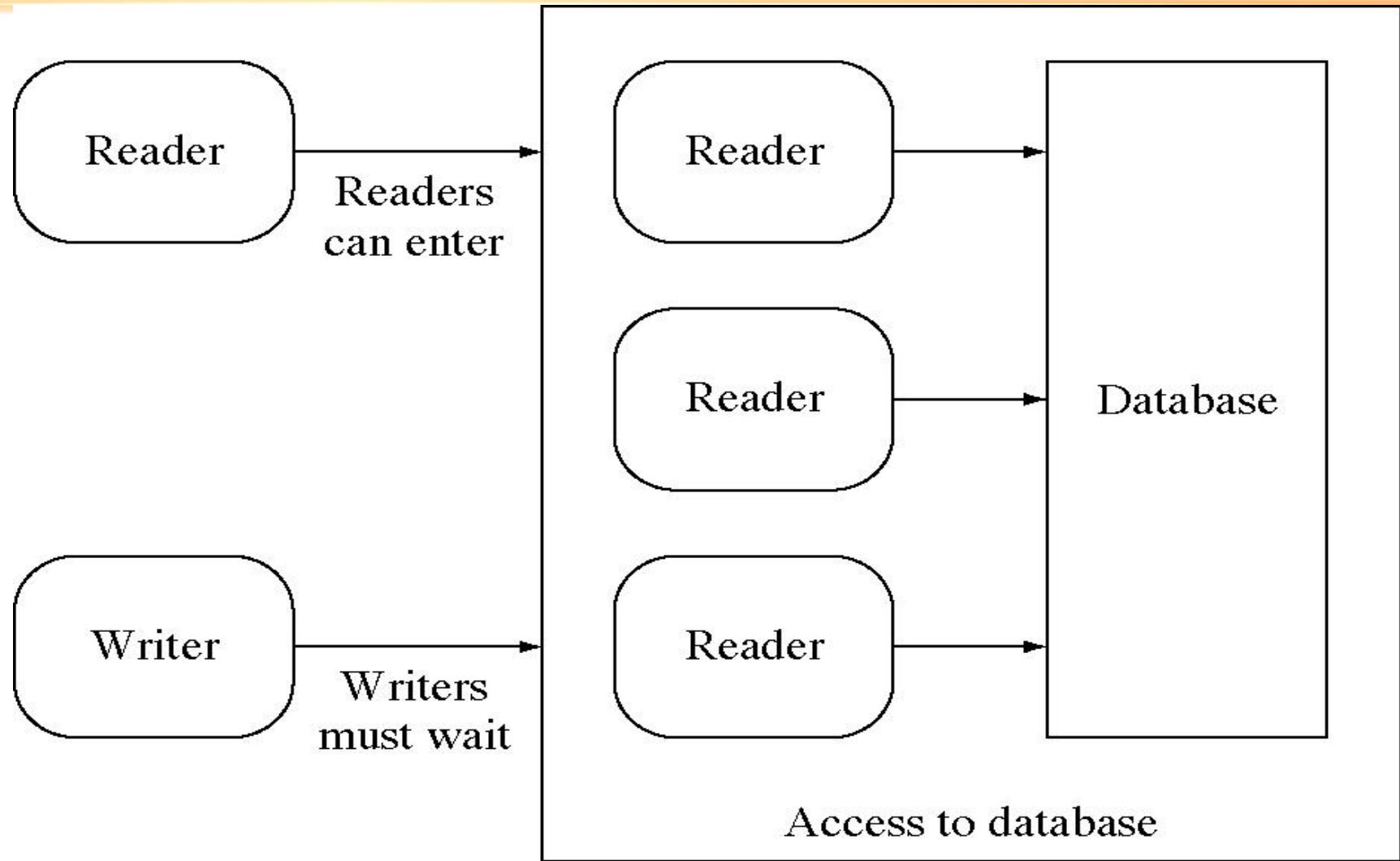
Readers & Writers

- Moâ taû : N tieán trình Ws vaø Rs hoaït ñoàng ñoàng haønh
 - Rs vaø Ws chia seû CSDL
 - W caäp nhaät noäi dung CSDL
 - Rs truy caäp noäi dung CSDL
- Tình huoáng
 - Caùc Rs cuøng truy caäp CSDL ?
 - W ñang caäp nhaät CSDL thì caùc Rs truy caäp CSDL ?
 - Caùc Rs ñang truy caäp CSDL thì W muoán caäp nhaät CSDL ?

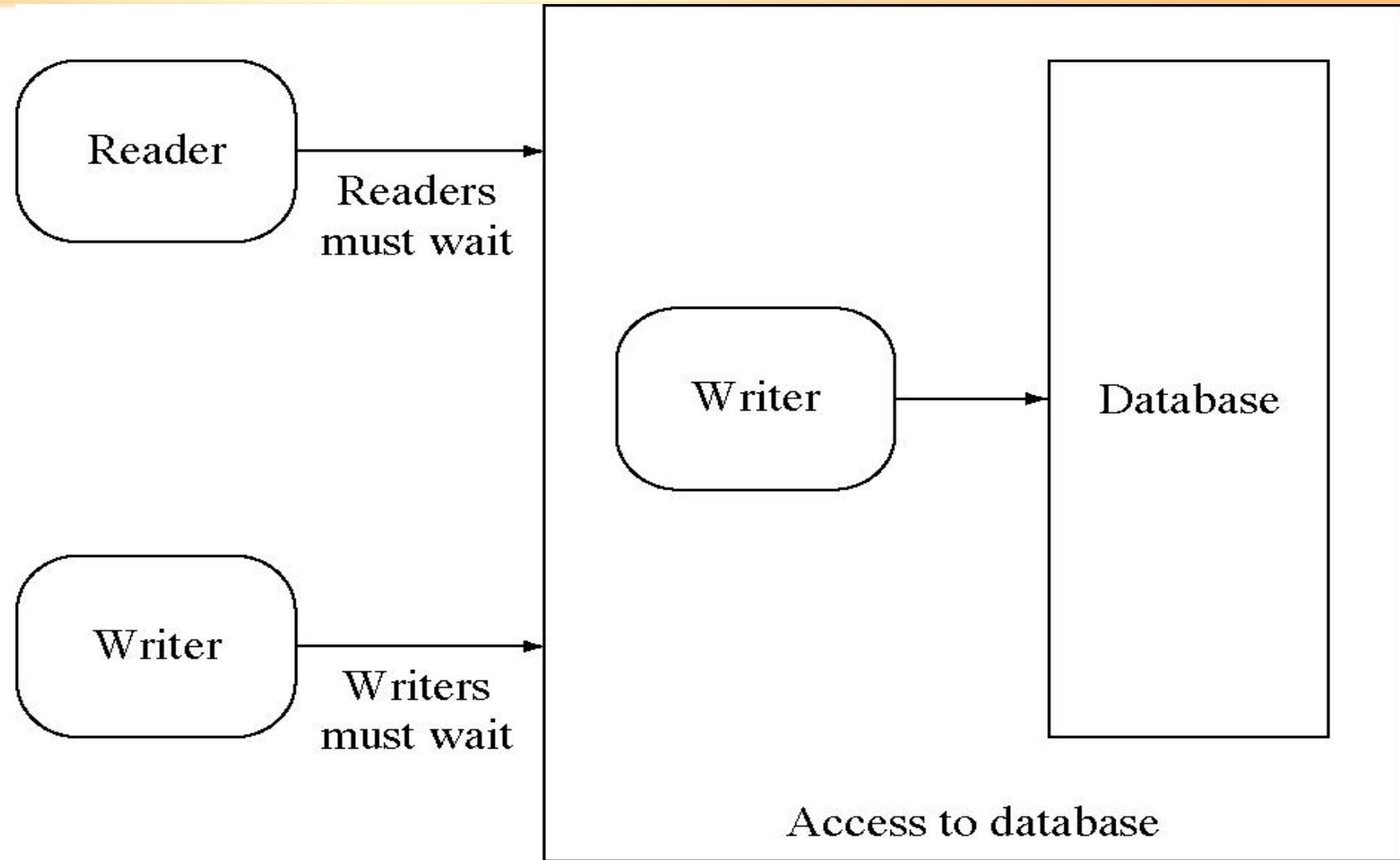


- W khoâng ñöôïc caäp nhaät döõ lieäu khi coù ít nhaát moät R ñang truy xuaát CSDL (ME)
- Rs khoâng ñöôïc truy caäp CSDL khi moät W ñang caäp nhaät noäi dung CSDL (ME)
- Taïi moät thôøi ñieäm , chæ cho pheùp moät W ñöôïc söûa ñoái noäi dung CSDL (ME)

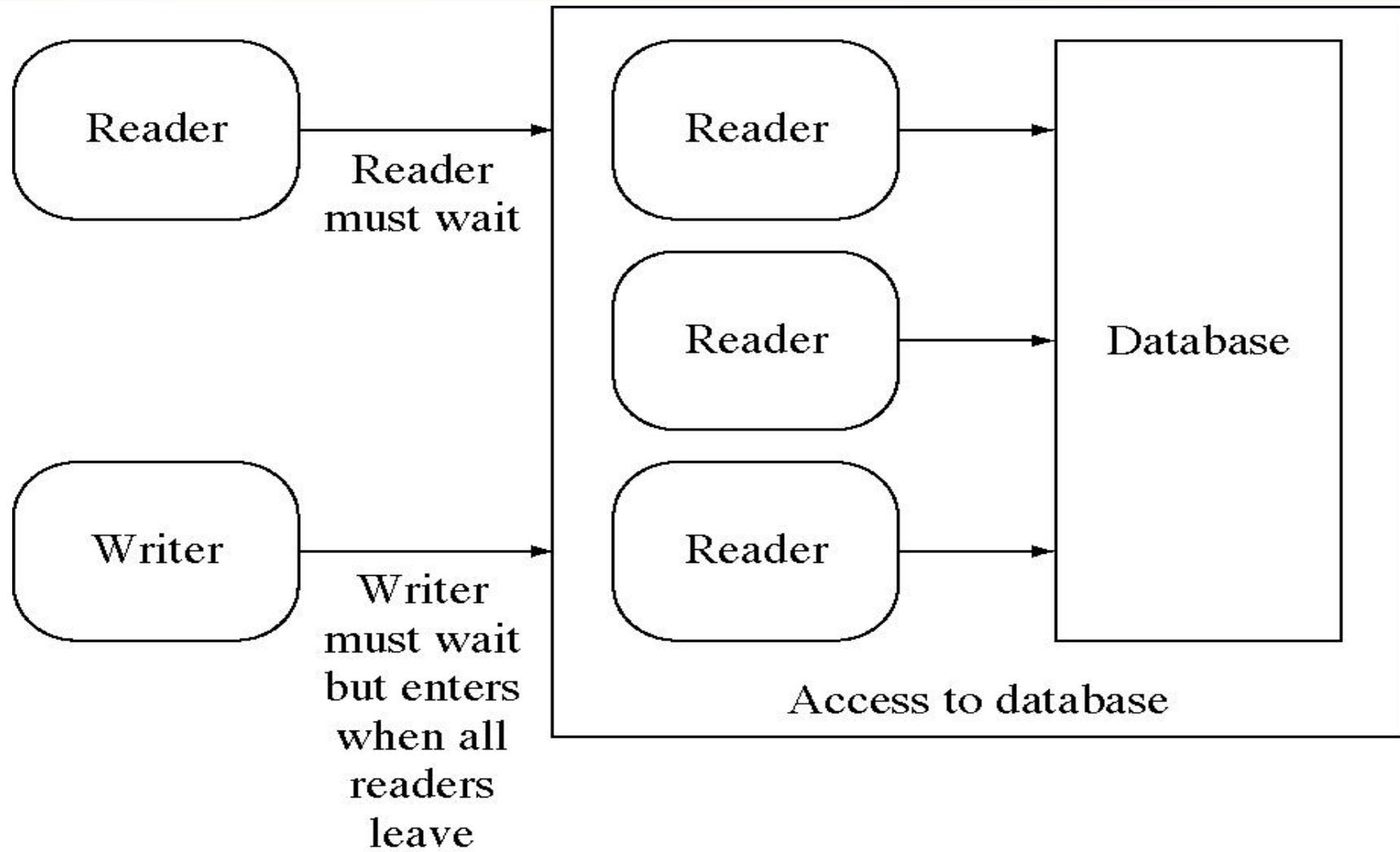
Readers-Writers vôÙi “active readers”



Readers-writers vòùi moät “active writer”



Öu tieân ai hôñ ñaây ?



Readers & Writers

- W ñoäc quyeän truy xuaát CSDL
- W hieän taïi keát thuùc caäp nhaät CSDL : ai vaøo ?
 - Cho W khaùc vaøo, caùc Rs phaûi ñôïi
 - Öu tieân Writer, Reader coù theå starvation
 - Cho caùc Rs vaøo, Ws khaùc phaûi ñôïi
 - Öu tieân Reader, Writer coù theå starvation

Readers & Writers : Giaûi phaùp Semaphore

- Caùc bieán duøng chung giöõa Rs vaø Ws
 - semaphore db = 1; // Kieåm tra truy xuaát CSDL

R&W : Giaûi phaùp Semaphore (1)

```
Reader()
```

```
{  
    down(&db);  
    read-db(Database);  
    up(&db);  
}
```

```
Writer()
```

```
{  
    down(&db);  
    write-db(Database);  
    up(&db);  
}
```

■ Chuyeän g̃i xaûy ra ?

■ Chæ coù 1 Reader ñöôïc ñoïc CSDL taïi 1 thôøi ñieåm !

R&W : Giaûi phaùp Semaphore (2)

Reader()

```
{  
    if (rc == 0)  
        down(&db);  
  
    rc = rc +1;  
  
    read-db(Database);  
  
    rc = rc - 1;  
  
    if (rc == 0)  
        up(&db);  
}
```

Writer()

```
{  
    down(&db);  
    write-db(Database);  
    up(&db);  
}
```

■ Nuùng chöa ?

- rc laø bieán duøng chung giöõa caùc Reader...
 - CS ñoù 😞

Readers & Writers : Giaûi phaùp Semaphore

- Caùc bieán duøng chung giöõa Rs vaø Ws
 - semaphore db = 1; // Kieäm tra truy xuaát CSDL
- Caùc bieán duøng chung giöõa Rs
 - int rc; // Soá löôïng tieán trình Reader
 - semaphore mutex = 1; // Kieäm tra truy xuaát rc

R&W : Giaûi phaùp Semaphore (3)

Reader()

```
{  
    down(&mutex);  
    if (rc == 0)  
        down(&db);  
    rc = rc +1;  
    up(mutex);  
    read-db(Database);  
    down(mutex);  
    rc = rc - 1;  
    if (rc == 0)  
        up(&db);  
    up(mutex);  
}
```

Writer()

```
{  
    down(&db);  
    write-db(Database);  
    up(&db);  
}
```

Ai nööïc öu tieân ?

R&W : Giải pháp Semaphore (Thinking...)

Reader()

```
{  
    down(&mutex);  
    rc = rc +1;  
    up(mutex);  
    if (rc ==1)  
        down(&db);  
    read-db(Database);  
    down(mutex);  
    rc = rc - 1;  
    up(mutex);  
    if (rc == 0)  
        up(&db);  
}
```

Writer()

```
{  
    down(&db);  
    write-db(Database);  
    up(&db);  
}
```

??? heâ, heâ, heâ ☺

R&W: Giảm phaüp Monitor

monitor ReaderWriter

? Database;

procedure R1();

{

}

procedure R...();

{

}

procedure W1();

{

}

procedure W...();

{

}

```
monitor ReaderWriter
```

```
condition OKWrite, OKRead;  
int rc = 0;  
Boolean busy = false;
```

```
procedure BeginRead()
```

```
{  
    if (busy)  
        wait(OKRead);  
    rc++;  
    signal(OKRead);  
}
```

```
procedure FinishRead()  
{
```

```
    rc--;  
    if (rc == 0)  
        signal(OKWrite);  
}
```

```
procedure BeginWrite()
```

```
{  
    if (busy || rc != 0)  
        wait(OKWrite);  
    busy = true;  
}
```

```
procedure FinishWrite()
```

```
{  
    busy = false;  
    if (OKRead.Queue)  
        signal(OKRead);  
    else  
        signal(OKWrite);  
}
```

```
end monitor;
```

Reader&Writer : Giảm thiểu phâùp Monitor

Reader()

{

RW.BeginRead();

Read-db(Database);

RW.FinishRead();

}

Writer():

{

RW.BeginWrite();

Write-db(Database);

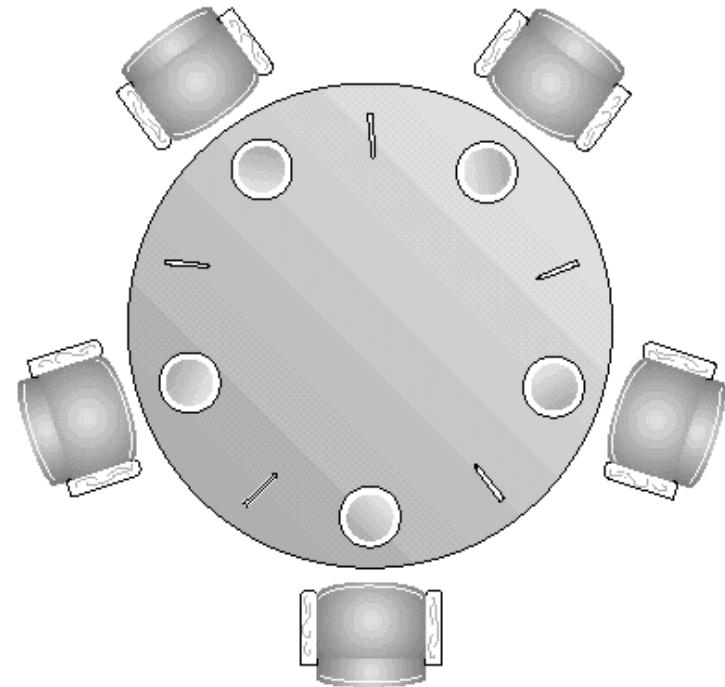
RW.FinishWrite();

}

Baøi toaùn ñoàng boä hoaù kinh ñieån 3:

Böûa aên cuâa caùc Trieát gia (Dining Philosophers)

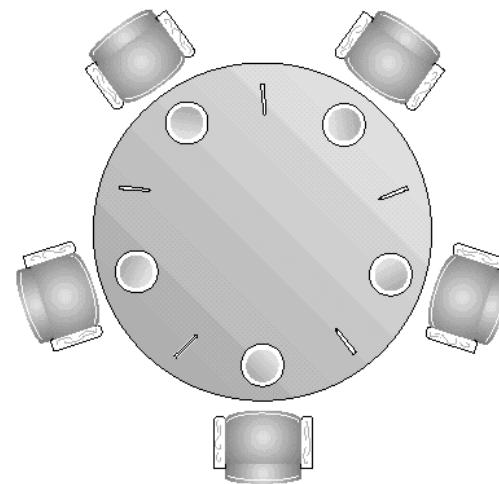
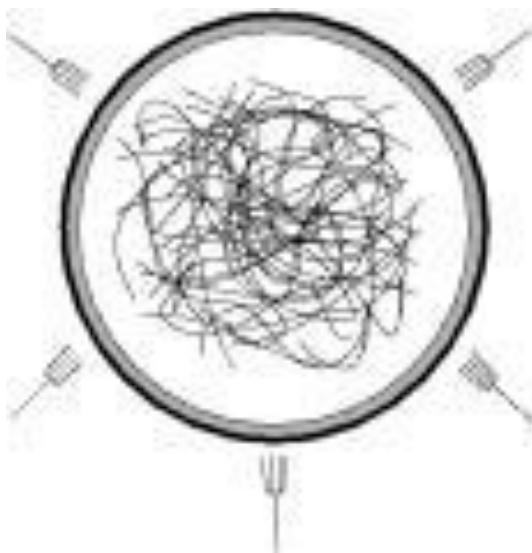
- Naêm trieát gia ngoài chung quanh baøn aên moùn spaghetti (yum..yum)
 - Treân baøn coù 5 caùi nóá ñööïc ñaët giöõa 5 caùi nóá (xem hình)
 - Ñeå aên moùn spaghetti moãi ngöôøi caàn coù 2 caùi nóá
- Trieát gia thöù i:
 - Thinking...
 - Eating...



Chuyeän gì coù theå xaûy ra ?

Dining Philosophers : Tình huống nguy hiểm

- 2 triết gia “giaónh giaät” cuøng 1 caùi nóá
 - Tranh chaáp
- Caàn ñoàng boä hoaù hoaít ñoäng cuûa caùc triết gia



Dining Philosophers : Giai tí phaùp ñoàng boää

semaphore fork[5] = 1;

Philosopher (i)

{

while(true)

{

down(fork[i]);

down(fork[i+1 mod 5])

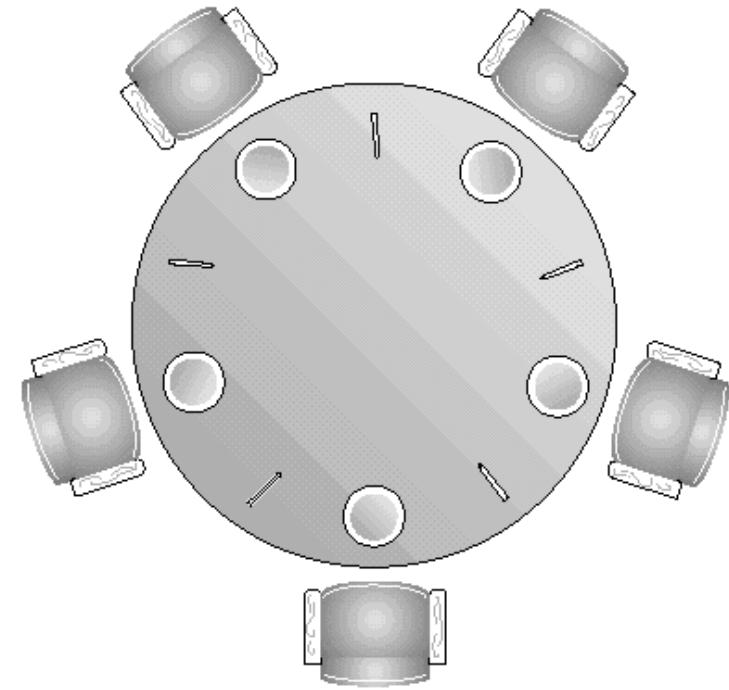
eat;

up(fork[i]);

up(fork[i+1 mod 5]);

think;

}



Deadlock

Dining Philosophers : Thàùch thöùc

- Caàn ñoàng boä sao cho:
 - Khoâng coù deadlock
 - Khoâng coù starvation