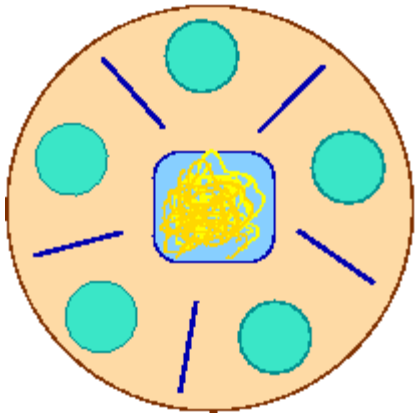
أمثلة عن الإستعصاء
Deadlock Exercises

د. فادي تركاوي

The Dining Philosophers

- المشكلة هنا أنه إذا أخذ كل فيلسوف الشوكة اليمنى فإن النظام يتعرض لإستعصاء



- الفيلسوف يفكر بشكل غير محدد زمنيا. و كل فيلسوف يقوم بالآكل سوف ينتهي من الأكل في النهاية. الفلاسفة يتناولون و يتركون الشوك بنفس الترتيب أو بدون تحديد، لكن هذه الأفعال تتم بشكل atomic .
- *any philosopher who tries to EAT eventually does .*

```
philosopher(i)
{
philosophise();
take_fork(i); // take the left fork
take_fork((i + 1) % N); // take the right
   fork , when it is already taken leave it
   and try sometime later the forkI
….
```

```
process P[i]
while true do
{
THINK;
PICKUP(CHOPSTICK[i],CHOPSTICK[i+1 mod 5]);
EAT;
PUTDOWN(CHOPSTICK[i],CHOPSTICK[i+1 mod 5])
}
```

مسألة ١عن كشف الإستعصاء

- لدينا العمليات الآتية:

P1,P2,P3 and P4 هذه العمليات تملك الموارد الآتية كما في المصفوفة Allocation Matrix أما الموارد التي تقوم العمليات بطلبها مذكورة في المصفوفة Request Matrix حيث أن الموارد المتوفرة هي الشعاع A

	R1	R2	R3
P1	٣	٢	١
P2	٢	٢	١
P3	١	١	١
P4	٠	٠	١

Request Matrix R

	R1	R2	R3
P1	١	١	١
P2	٢	١	٢
P3	١	١	١
P4	١	١	١

Allocation Matrix C

A = (١ , 0 , ٠) (R3,R2,R1) */ Available Resources المصادر المتوفرة حاليا /*

أذكر بالتفصيل كيف يتم تخصيص الطلب الموارد من قبل العمليات بحيث يتم تجنب الوقوع بالإستعصاء أو ما يسمى Deadlock Detection

مع ذكر حالة المصادر المتوفرة بعد إنتهاء كل عملية طلب للموارد من قبل العمليات

الحل

- في البداية تأتي العملية P4 ثم فيصبح المصادر المتوفرة 1 1 2
- بعدها تأتي P3 فتصبح المصادر المتوفرة هي 2 2 3
- بعدهى تأتي العملية P2 فتصبح المصادر المتوفرة 4 3 5
- بعدها يتم تنفيذ العملية P1 فنحصل على المصادرة الموجودة و هي Existing Resources = (5 , 4 , 6)

مسألة ٢ عن كشف الإستعصاء

في نظام تشغيل تقوم عدد من العمليات P1,P2,P3,P4 و التي تملك المصادر (A,B,C) Allocation Matrix و تحتاج المصادر Request Matrix. بإستخدام خوارزمية Deadlock Detection Algorithm هل هناك حالة إستعصاء؟

	A	B	C
P1	٠	٠	٠
P2	٢	٠	٣
P3	٠	١	٠
P4	١	١	١
P5	٦	٠	٢

Allocation Matrix

	A	B	C
P1	٠	١	٠
P2	٢	٠	٠
P3	٢	٠	٢
P4	٢	١	١
P5	٠	٠	٢

Request Matrix

Available Resources A = (0 ,0 ,0) /* المصادر الحالية */

Existing Resources E= (5 ,2 ,6) /* المصادر الكلية */

و أذكر المصادر المتوفرة بعد كل عملية طلب و ما هو تسلسل تنفيذ العمليات لكي يتم تجنب الإستعصاء من قبل العمليات.

الحل

- بعد كل عملية تخصيص يتم تنفيذ العمليات كالآتي بحيث يتم إكتشاف الإستعصاء و يكون بعدها المصادر المتوفرة كالآتي
- p1 و بعدها يتوفر لدينا المصادر الآتية (0, 1, 0)
- p3 و بعدها يتوفر لدينا المصادر الآتية (1,2, 2)
- p4 و بعدها يتوفر لدينا المصادر الآتية(3, 2, 4)
- p2 و بعدها يتوفر لدينا المصادر الآتية(3, 2, 6)
- p5 و بعدها يتوفر لدينا المصادر الآتية(5, 2, 6)

مثال ١ تجنب الإستعصاء

- One resource type R with 22 unit
- Three processes X, Y, and Z with initial claims 3, 11, and 19 respectively.
- Currently the processes have 1, 5, and 10 units respectively.
- Hence the manager currently has 6 units left.
- Also note that the max additional needs for the processes are 2, 6, 9 respectively.
- So the manager cannot assure (with its **current** remaining supply of 6 units) that Z can terminate. But that is **not** the question.
- This state is safe
  - Use 2 units to satisfy X; now the manager has 7 units.
  - Use 6 units to satisfy Y; now the manager has 12 units.
  - Use 9 units to satisfy Z; done!

process	initial claim	current alloc	max add'l
X	3	1	2
Y	11	5	6
Z	19	10	9
Total		16	
Available		6	

A safe state with 22 units of one resource

مثال ٢ تجنب الإستعصاء

- Currently the processes have 1, 5, and 12 units respectively.
- The manager has 4 units.
- The max additional needs are 2, 6, and 7.
- This state is unsafe
  - Use 2 unit to satisfy X; now the manager has 5 units.
  - Y needs 6 and Z needs 7 so we can't guarantee satisfying either
- Note that we were able to find a process that can terminate (X) but then we were stuck. So it is not enough to find one process. We must find a sequence of all the processes.

process	initial claim	current alloc	max add'l
X	3	1	2
Y	11	5	6
Z	19	12	7
Total		18	
Available		4	

An unsafe state with 22 units of one resource