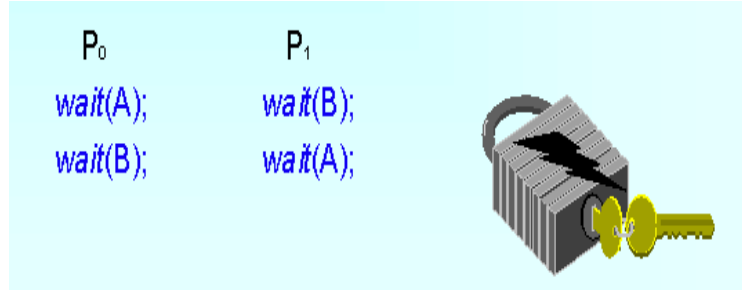


ماهو الجمود؟

هو منع العمليات من استخدام موارد النظام بشكل دائم بسبب تنافسها مع العمليات الأخرى على هذه الموارد أثناء عملية الاتصال بينهم.



سبب حدوث الجمود

عندما تكون هناك عمليات تنتظر تنفيذ مهام- قامت فيها عمليات سابقة- وهذه العمليات السابقة هي بدورها تنتظر انقضاء مهام أخرى.

لتبسيط هذه العملية نأخذ هذه الامثلة :

المثال الأول:

- لتكن هناك عمليتان س و ص كلاهما تريدان طباعه مستند موجود على محرك الأقراص
- 1- س طلبت استخدام الطابعة وأخذت الموافقة
 - 2- ص طلبت استخدام محرك الأقراص وأخذت الموافقة
 - 3- س طلبت استخدام محرك الأقراص لكن طالبها رفض ريثما ينتهي ص من استخدامه
 - 4- ص طلب استخدام الطابعة
- هنا يحدث الجمود!!

المثال الثاني:

لدينا قلم و مسطره و هناك شخصين يريدون الرسم احدهم يحمل القلم والأخر المسطرة

في حالة ان الشخص الذي يحمل القلم أراد الحصول على المسطرة من الشخص الآخر و الشخص الذي يحمل المسطرة أراد القلم فإن الجمود يظهر فيه هذه الحالة لأن كلا الطرفين لا يمكن تحقيقهم .

المثال الثالث:

الجسر...حيث انه يكون ذو اتجاه واحد..ولكن ماذا يحدث إذا خالفت سيارة ذلك وتعاكست سيارتان؟؟ سيتولد الجمود .

ونستطيع حله بأن تتراجع إحدى السيارات، أي أننا أجهضنا إحدى السيارات عن حركتها (preemption)، ولكن المشكلة في حالة العمليات أننا لا نستطيع أن نجهض العملية دائما...

المثال الرابع:

لدينا عمليتين وكل واحدة منهم تريد أن تطلب ملفين ، العملية الأولى طلبت الملف الأول والعملية الثانية طلبت الملف الثاني وبذلك تنتظر كل عملية الأخرى لتحرر الملف حتى يمكنها من طلبه..

الموارد (Resources):

هي الأشياء التي تستخدمها العملية لتنفيذ مهامها قد تكون: الأجزاء الصلبة من الحاسوب(المهاردوير) مثلا محرك الأقراص, الطابعة أو غيرهما. أو قد تكون معلومات مثلا ملف أو غيره. أي أن الموارد هي أي شي يستخدم لتنفيذ عملية مخصصة يحجز لمدة زمنية محده أي إلى انقضاء تنفيذ المهمة.

هناك نوعان من الموارد:

- 1- يمكن احتكارها (Preemptable) مثل الذاكرة .
- 2- لا يمكن احتكارها (Non-Preemptable) مثل الطابعة.

الموارد التي يمكن احتكارها:

هي الموارد التي تعين لعملية ما إلى أن تنتهي من تنفيذ مهمتها ثم تعين إلى مهمة أخرى بدون أي تصادم.

الموارد التي لا يمكن احتكارها:

هي الموارد التي لا يمكن أن تؤخذ من عملية وتعين إلى أخرى بدون أن تسبب تصادم. "عند ذكرنا لمثال الطابعة لا نقصد أن تعين إلى عملية أخرى خلال تنفيذ مهامها, بل نظام التشغيل يعامل الطابعة كمورد يمكن احتلاله – المشغل يرتب المهام التي يجب على الطابعة تنفيذها يحدث في الموارد التي لا يمكن احتلالها كما سنرى أن الجمود

التسلسل لاستخدام مورد هو:

- 1- طلب المورد: إذا كان المورد متوفر (أي لا تستخدمه عملية أخرى) فإنه يمنح على الفور أو ينتظر إذا كان يستخدم في عملية أخرى.
- 2- استخدام المورد: في حال الحصول عليه.
- 3- تحرير المصدر: في حال الانتهاء من تنفيذ المهمة.

شروط حدوث الجمود

1- Mutual exclusion (منع التبادل)

فقط عملية واحده يمكنها أن تستخدم المورد. إذا كان المورد ممكن استخدامه بنفس الوقت لعدة عمليات فإن هذا لا يسبب Deadlock.

2- الاستخدام و الانتظار (Hold and Wait)

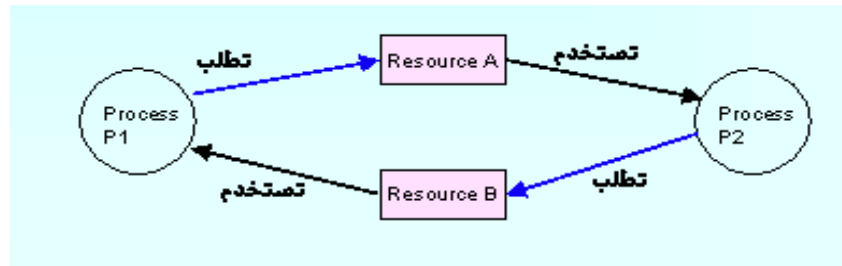
- أن تكون العمليات حازمة مورد وتطلب آخر

3- عدم الإجهاض (no preemption):

أي أن العملية التي تحمل المورد لا يمكن إجهاضها و لا بد أن تقوم هي بتحرير الموارد التي تحملها.

4- الانتظار الدائري (Circular wait):

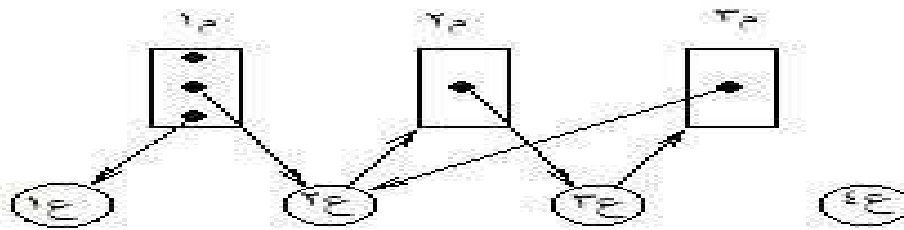
معنى أن تتكون دائرة مغلقة من الإنتظارات تنشأ من احتكار عملية لمورد أو أكثر تنتظره عملية أخرى، وهذه العملية بدورها تنتظر عملية أخرى.... وهكذا.



** علماً أن توافر هذه الشروط لا يضمن حدوث الجمود فهي ضرورية، ولكنها ليست كافية، لحدوث حالة الجمود.

تمثيل الجمود:

(Resource-allocation graph) يمكننا استخدام رسم بياني يسمى تعيين الموارد



القيم تمثل الموارد و العمليات

مجموعة القيم تقسم إلى صنفين مختلفين :

1- $E = \{1, 2, 3, 4\}$ وهذه المجموعة تتكون من كل العمليات الفعالة في النظام.

2- م = {1م، 2م، 3م} وهذه المجموعة تتكون من أنواع الموارد في النظام .

سوف نستخدم القمم المستطيلة لتمثيل المورد والنقاط الداخلية لتمثيل الطلب على المورد وسوف نستخدم القمم الدائرية لتمثيل العملية.

الخطوط من العملية إلى المورد تمثل الطلب وتسمى بخطوط الطلب.

الخطوط من النقاط الداخلية في المورد إلى العملية تعني انه تم تعيين المورد لهذه العملية وتسمى بخطوط التعيين.

في الرسم البياني أعلاه نجد أن: **

هناك ثلاث طلبات على المورد م1 , ونجد أن العمليتان ع1 وع2 تم تعيين المورد لهما.

ع2 يطلب مورد م2 لكن ينتظر الحصول عليه بعد أن ينتهي منه ع3.

Deadlock ع3 يطلب م3 الذي يستخدمه ع2 هنا يحدث

الانتظار الدائري هنا يمكن تمثيله كالتالي:

ع2 < - م2 < - ع3 < - م3 < - ع2

طرق التعامل مع الجمود:

1) تجاهل المشكلة (Ignore the problem):

هذه الإستراتيجية معقولة، الجمود من غير المحتمل حدوثه في أغلب الأحيان، فإن نظام التشغيل يمكن

أن يعمل لسنوات دون حدوث جمود. نظام التشغيل إذا له نظام منع الجمود أو نظام كشف

الجمود فإن هذا سيؤثر سلبيا على نظام التشغيل (بيطئ الجهاز).

2) كشف الجمود (Deadlock Detection):

دائما يلبي طلب المصدر إذا كان ممكن، ويراقب الجمود بشكل دوري، وإذا حصل جمود يقوم

بمعالجته. إذا كان هناك instance لكل مصدر فإنه من المحتمل اكتشاف الجمود بواسطة

الرسم البياني ونبحث عن وجود cycle .

3) منع الجمود (Deadlock Prevention):

يرفض أحد الشروط الأربعة للجمود.

4) تجنب الجمود (Deadlock Avoidance):

لا يبلي طلب المصدر إذا كان يمكن أن يؤدي إلى الجمود. فهذا يتفادى الجمود الذي قد يحدث للمصدر لاحقاً.

** الفرق بين منع الجمود deadlock prevention و تجنب الجمود deadlock

avoidance غير ملحوظ إلى حد ما . تشير إستراتيجية تجنب الجمود إلى أنه عندما يطلب المصدر فهو يمنح للعملية فقط إذا كان لا يؤدي إلى الجمود.

تتضمن إستراتيجية منع الجمود إلى تغيير القواعد بالتالي فإن العملية لن تتمكن من طلب المصدر الذي يمكن أن يؤدي إلى الجمود.

الحالة الآمنة:

هي الحالة الوحيدة التي تضمن أن كل العمليات باستطاعتها أن تقوم بالمهام التي يتوجب عليها القيام بها. أما الحالة غير الآمنة لا تعطي مثل هذا الضمان.

نستطيع تفادي الجمود Banker باستخدام خوارزمية

تنص هذه الخوارزمية انه لا تنفذ العملية إذا كانت ستضع النظام في حالة غير آمنة-أي أن ليس كل العمليات سوف تقول بمهامها.

يعلم مسبقاً فيما يتعلق بالموارد التي سوف يتطلب أن يكون نظام التشغيل تفادي الجمود تطلبها العمليات وتستخدم لفترة زمنية محدودة

شفاء الجمود:

توجد حلول عديدة لمعالجة الجمود بعد اكتشافه منها: أعلام المستخدم بوجود حالة الجمود وترك الخيار له لمعالجة الجمود أو ترك نظام التشغيل يعالج الجمود ويشفى منه أوتوماتيكياً. وعندما يترك الخيار لنظام التشغيل الشفاء من الجمود فهناك خيارات لكسر الجمود:

1/ إجهاض أحد المهام العالقة في الانتظار المسبب للجمود.

2/ منع المهام العالقة في الجمود من بعض المصادر أو كلها.

أ/ إجهاض المهمة:

لكسر الجمود عن طريق الإجهاض يسلك نظام التشغيل إحدى الطريقتين إما إجهاض جميع المهام العالقة في الجمود أو إجهاض أحد المهام العالقة في الجمود. وفي جميع الحالات يسترد نظام التشغيل جميع المصادر التي لدى المهمة المجهضة.

1/ إجهاض جميع المهام العالقة في الجمود:

وفي هذه الحالة سوف يكسر الجمود بشكل أكيد ولكن هذه العملية تعتبر مكلفة نظرا لان المهام المجهضة تضطر لإعادة العمل وحجز المصادر وإضاعة الكثير من الوقت.

2/ إجهاض أحد المهام التي تكسر حالة الجمود:

عندما يتم إجهاض أحد المهام يقوم نظام التشغيل باستدعاء خوارزمية اكتشاف الجمود التأكد من انتهاءه أو وجوده وإذا كان الجمود لا يزال موجودا فلا بد من إجهاض مهمة أخرى وإعادة العمل حتى ينتهي الجمود.

أن إجهاض المهمة ليس بالعمل السهل. حيث أن المهمة قد تكون في منتصف التعديل على ملف وإجهاضها يترك الملف في حالة غير صحيحة وكذلك أيضا إجهاض المهام التي تقوم بعملية طباعة فلا بد من إعادة الطباعة قبل انجاز الطباعة التالية.

فلذلك عندما نقوم بإجهاض أحد المهام فلا بد من اختيار المهمة ذات التكلفة الأقل. وتحديد المهمة ذات التكلفة الأقل ليس بالأمر السهل وهي تعتمد على سياسة نظام التشغيل وعلى الطريقة التي يتبعها لتحديد التكلفة.

ويعتمد تحديد تكلفة المهمة على:

1/ أولوية المهمة. 2/ مدى العمل الذي أنجزته. 3/ كم عدد وما نوع المصادر التي لدى المهمة. 4/ وكم عدد المهام التي أحتاج أن ينتهي تنفيذها. 5/ كم سيحتاج من المصادر الإضافية. 6/ أي المهام متفاعلة حاليا.

ب/التسابق على المصادر:

حيث تقوم المهمة بطلب المصادر التي تحتاجها فتحصل على البعض والبعض الآخر لا تستطيع الحصول عليه حتى لا تعلق المهمة في الجمود.

أما إذا كانت المصادر مع المهام العاقة في الجمود. فهناك 3 حالات:

1/اختيار الضحية: يعني أي المهمة سوف يتم إنهاؤها وتمتلك مصادرها. ولاختار المهمة الضحية يكون لدى نظام التشغيل أولويات للامتلاك حتى يحقق تقليل التكلفة. ويحدد عامل التكلفة بعدد المصادر وكمية العمل المنجزة لدى كل مهمة

2/إعادة تنفيذ المهمة: بعد أخذ المصادر من المهمة الضحية لا تستطيع المهمة المضي في التنفيذ بل يعاد التنفيذ إلى نقطة آمنة تستطيع المهمة التنفيذ منها بشكل صحيح.

3/التجويج: عندما نختار مهمة ضحية فكيف نتأكد انه لن يحصل لها تجويج أي لن تكون هي لضحية في كل مره يكون اختيار الضحية على أساس التكلفة القليلة وحتى لا تكون المهمة هي الضحية في كل مرة يقوم نظام التشغيل بإضافة عدد المرات التي كانت المهمة فيها هي الضحية إلى التكلفة وفي كل مرة يبحث فيها نظام التشغيل عن الضحية يقارن بين المهام العالقة في الجمود بعدد المرات التي حصل فيها للمهمة إعادة تنفيذ وهذا يعتبر من أكثر الحلول شيوعا.

***أيضا من حلول الجمود

1- الربط السابق لمصادر الحاسب قبل وقوع حالة الجمود, بحيث يتم جدولة العمليات المختلفة حسب حاجتها من مصدر الحاسب, وشرط تجنب حالة الجمود, ولهذا الحل العيوب التالية:

أ- يحدث للمستخدم أن لا يعلم أوقات وفترات ارتباط كل مصدر من مصادر الحاسوب بالعمليات المختلفة, وذلك قبل بدء التنفيذ, وبالتالي يصعب تحديد عملية الربط المسبق لمصدر الحاسوب.

ب- عند ربط مصدر الحاسوب بعمل ما من الأعمال مسبقا, قد يتم حجز هذا المصدر فترة طويلة دون أن يكون لهذا المصدر أي عمل وظيفي يؤديه, وبذلك يؤخر تنفيذ الأعمال الأخرى.

ج- من العيب أن يتم ربط مصدر من مصادر الحاسوب مسبقا, خاصة إذا صغر احتمال استخدام ذلك المصدر بذلك العمل المرتبط به, كأن يتم ربط الطابعة مثلا مسبقا بعمل قد لا يحتاج أكثر من سطر واحد طباعة يتم طباعته في نهاية العمل.

2- ربط المصادر بالأعمال تحت شروط **Constrained Allocation** ويتفرع من هذا الربط مايلي:

1- الربط القياسي **Standard Allocation**

2 - الربط المتحكم به **Controlled Allocation**

أي لا يتم ربط المصادر بالأعمال إلا وفق قيود محددة, تمنع حالة الجمود. أما في الربط القياسي فيعني وضع الأعمال في جدول مرتبة حسب تسلسل أرقام مصادر الحاسوب التي تدل على أولويات الربط.

3- التحقق من حدوث حالة العناق الميت وحل تلك الحالة لحظة حدوثها **Detect and Recover**

وتعني هذه الطريقة ترك العمليات تحت التنفيذ, ومراقبة حالة الجمود, حتى إذا وقعت تم معالجة تلك الحالة في حينه ويلزم لذلك جدولان هما:

أ- جدول تعيين المصادر (**Resource assignment table**):

وظيفة هذا الجدول تحديد رقم المصدر ورقم العملية المرتبطة بذلك المصدر

ب- جدول العمليات المنتظرة (**process wait table**):

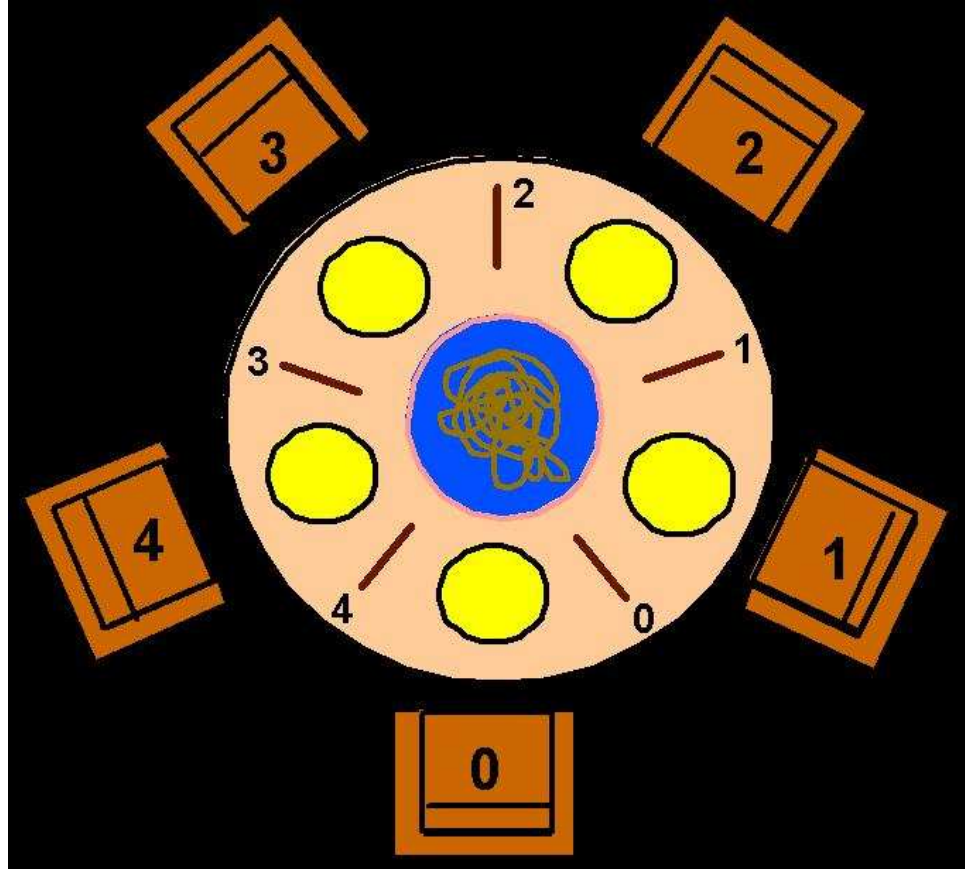
وظيفة هذا الجدول تحديد أرقام العمليات الواقعة في حالة الانتظار وأرقام المصادر المرتبطة بتلك العمليات المنتظرة

يتعاون الاثنان على تحديد حالة الجمود وحلها.

***طعام الفلاسفة:**

وقد مثل على مشكلة الجمود ادغار ديكسترا في 1968 بأمر مشهور وهو مشكلة طعام

الفلاسفة (**Dining Philosophers**):



فلو كان هناك خمسة فلاسفة يجلسون حول طاولة مستديرة وكل فيلسوف يقضي وقته ما بين التفكير والأكل وفي وسط الطاولة صحن كبير من السباغيتي وسيحتاج الفيلسوف إلى شوكتين ليتناول حصته من السباغيتي..ولسبب ما..لن يأخذ الفلاسفة سوى خمسة شوكة طعام ولذا ستوضع شوكة واحدة بين كل زوج من الفلاسفة وقد اتفقوا على أن كل منهما سيستخدمها عند حاجته لها بيده اليمين أو الشمال حسب موقع الشوكة منه

وفي هذا الموقع
(http://www.doc.ic.ac.uk/~jnm/book/book_applets/Diners.html)

ستجدين برنامج يوضح الوقت الذي يمضيه الفيلسوف في الأكل أو التفكير. لاحظي أن:

اللون الأصفر يعني أن الفيلسوف يفكر.

اللون الأزرق يعني أن الفيلسوف جائع

اللون الأخضر يعني أن الفيلسوف يأكل.

استخدمي زر التجميد "Freeze" و زر المتابعة "Restart" للتحكم بالمحاكاة. وجري تحريك الشريط إلى أقصى اليسار ولا حظي ماذا سيحدث؟؟

بعد بضع ثوان ، سيصبح كل من الفلاسفة جائعاً وسيمسك كل منهم شوكة واحدة معه وسينتظر توفر شوكة ثانية له وبما أنه لا يوجد المزيد من الشوك وكل منهم يحتاج لشوكتي طعام حتى يتمكن من الأكل فإن أياً منهم لن يتمكن من الأكل وسيصلون لمأزق الجمود وعدم القدرة على المتابعة.

يقضي الفلاسفة حياتهم بين التناوب على التفكير والأكل ، الفيلسوف بحاجة لالتقاط الشوك التي عن يمينه وعن شماله ليقوم بعملية الأكل، عندما يبدأ الفيلسوف بالتفكير ، يضع الشوك على الطاولة مرة أخرى.

وخلال عملية الأكل يكون أحد الفلاسفة يفكر أو جوعان أو...

وسنلاحظ أن مشكلة طعام الفلاسفة توفرت فيها شروط حدوث الجمود.

-فيلسوف واحد فقط يستطيع استخدام الشوكة في الوقت الواحد، ما يعني أن الشوكة مورد مقصور على فيلسوف واحد فقط.

-الفيلسوف الجائع والذي معه شوكة واحدة فقط سوف يحتفظ بها ويحتجزها وسينتظر الحصول على شوكة أخرى.

-وبما أن الفلاسفة مسالمون ومهذبون فلا أحد منهم سيرغب أو يحاول انتزاع الشوكة من جاره بالإجبار.

اتبعي [الرابط السابق](#) لرؤية البرنامج الآخر الذي حلّ المشكلة السابقة. حيث تجنب احتمال حدوث الجمود بأن جعل عدداً زوجياً من الفلاسفة يأخذون الشوك من البقية بترتيب مختلف، وهو أول اليسار