الباب الخامس

الايصـــــــاد (Deadlock)

- تعريف المورد
- مفهوم الايصاد
- مخططات تخصيص المورد (Resource Allocation Graphs)
- تمثيل الانتظار الدائري
- صيغة النظام
- مواصفات الايصاد
- التعامل مع الايصاد
  - الوقاية
  - العلاج

مقدمة

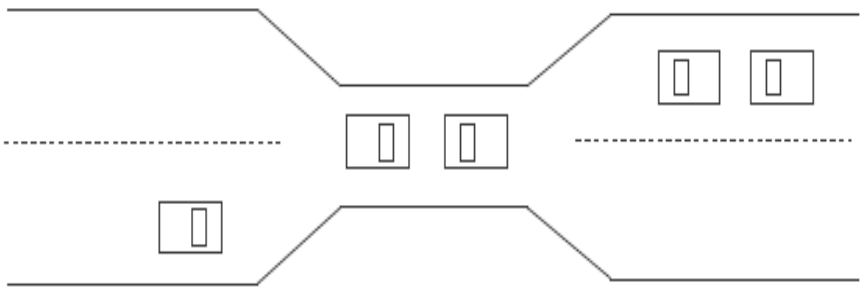
في الانظمة متعددة البرامج تتنافس العمليات على مجموعة من **الموارد**، بحيث تطلب عملية ما مورد ما، فاذا لم يكن متوفر ستضطر **للانتظار** (أي تتحول الى حالة الانتظار Blocked state) وقد تبقى على هذه الحالة باستمرار وذلك لاحتمال وجود عملية اخرى تحجز هذا المورد وهي ايضا في حالة **انتظار**.

تعريف المورد (Resource)

المورد هو كل ما يستخدمه البرنامج من **عتاد** (مثل الطابعة، القرص الصلب، والذاكرة...الخ) **وبرامج وملفات** (مثل جدول قاعدة بيانات، برنامج ما، صفحة إنترنت.)

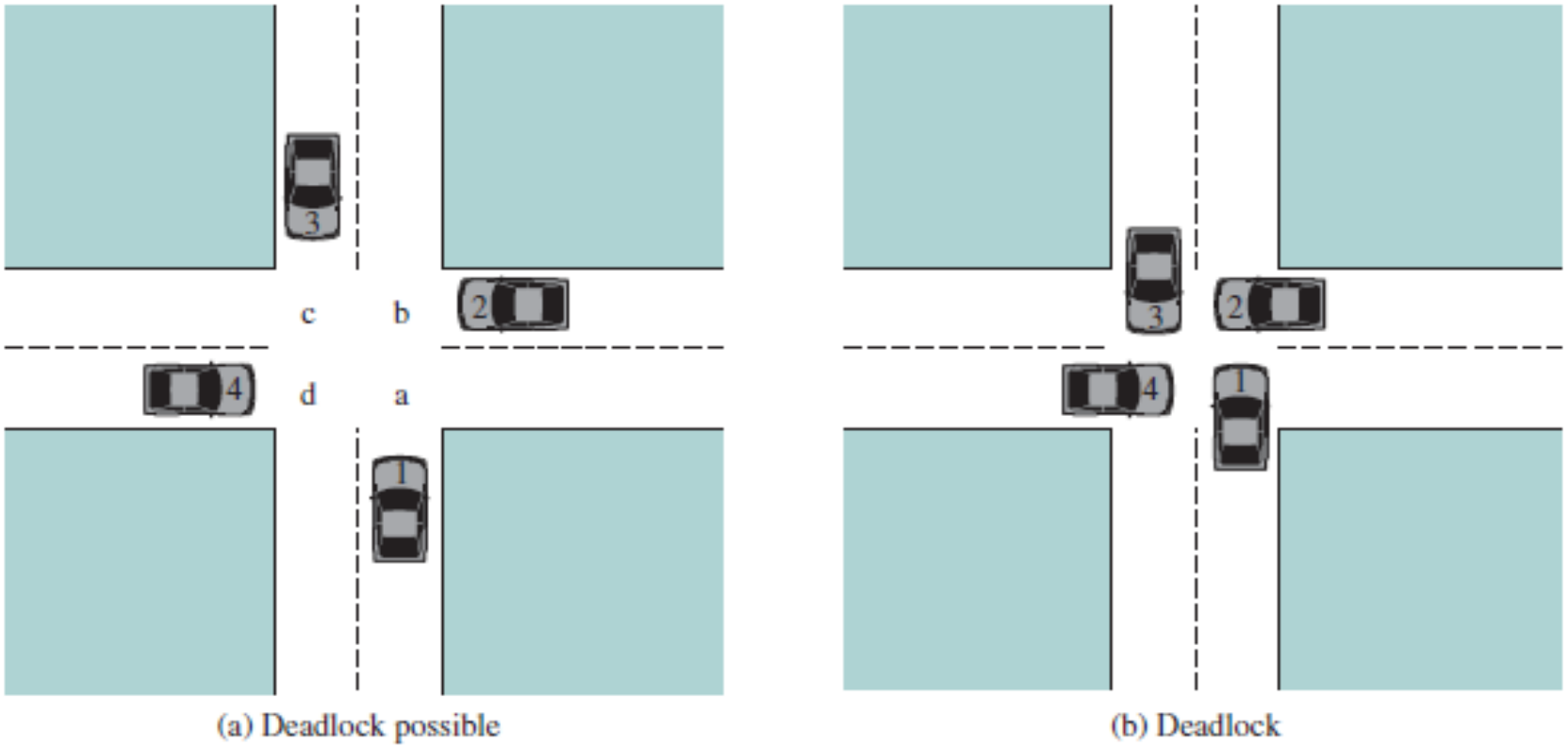
مفهوم الايصاد (تشابهي)

إذا كنت تقود سيارة (**عملية**) ثم دخلت في ممر ضيق (**مورد**)يتسع لسيارة واحدة فقط، وجاءت سيارة من الاتجاه المعاكس (**عملية أخرى**)ودخلت نفس الممر ، فالتقت السيارتان في منتصف الممر وكل سيارة تريد المرور عبر الممر، هنا نقول أن هنالك حالة ايصاد او اختناق في الممر، شكل 5-1



شكل 5-1 : الايصاد او الاختناق

مثال آخر، إذا كان هنالك تقاطعات طرق كما في **الشكل 5-2** ، فإن أي سيارة لا يمكنها التحرك ما لم نخرج بعض السيارات خارج الطريق (شكل 5-2).



شكل 5-2 : الاشارة الضوئية

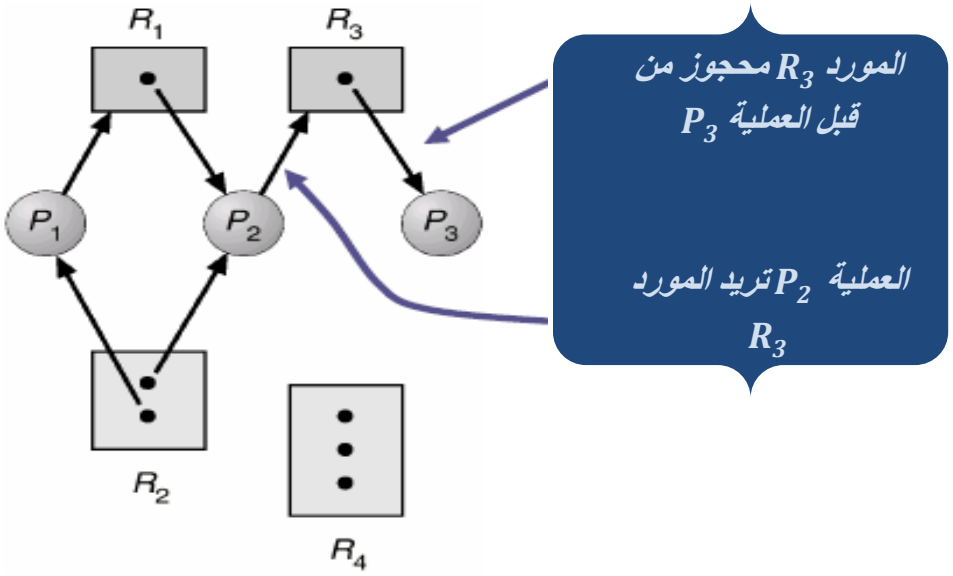
مخططات تخصيص المورد (Resource Allocation Graphs)

يمكننا استخدام بعض الرسومات لتوضيح مسببات الاختناق .مثلا يمكن استخدام:

- **المربع** ليدل على **مورد**، في الشكل 5.3، يعتبر كل من **R و S موارد**.
- **الدائرة** لتدل على **عملية**، في الشكل 5.3، يعتبر كل من **A و B عمليات**.
- **السهم** الذي يخرج من **دائرة** (**عملية**) ويشير إلى **مربع** (**مورد**) يعني أن العملية تريد (تطلب هذا المورد)، مثلا في **الشكلb5.3**، نجد أن **العملية B تريد المورد S**.
- **السهم** الذي يخرج من **مربع** (**مورد**) ويتجه الى **دائرة** (**عملية**) يعني أن هذا المورد يستخدم بواسطة العملية **(محجوز)**، في **الشكل a5.3**، **المورد R مستخدم بواسطة العملية A.**

شكل رقم (' 5.3): رسومات توضح العلاقة بين الموارد والعمليات.

- في حالة **وجود اكثر من مورد من نفس النوع** فالسهم الذي يخرج من داخل المربع (**مورد**) ويتجه الى دائرة (**عملية**) يعني أن هذا المورد يستخدم بواسطة العملية **(محجوز)**، في الشكل c5.3 نلاحظ مثلا ان المورد **R2** مستخدم من قبل العلميتين **P1,P2**

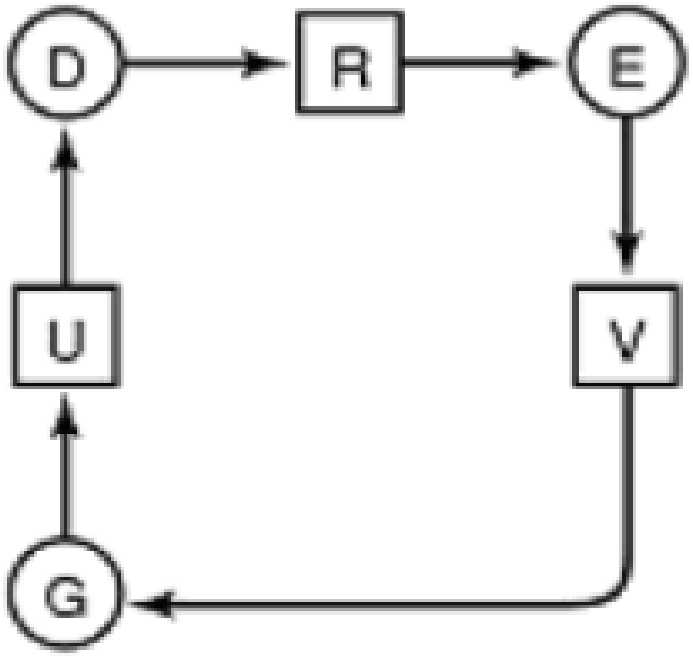


الشكل c5.3

تمثيل الانتظار الدائري بالرسم

نجد أن هنالك **انتظار دائري** حيث نجد أن العملية - مثلا في **الشكل 5.4**، **D** تستخدم المورد **U** وتحتاج إلى المورد **R** ، ونجد أن المورد **R** تستخدمه العملية **E** التي تريد المورد **V** الذي يستخدم بواسطة العملية **G** ، ونجد أن **G** تريد المورد **U** الذي تستخدمه العملية **D** وهكذا نجد أنفسنا في **دائرة انتظار**.

إذا ابتدأت من أي **عملية** أو **مورد** في أي شكل ، وتتبعت اتجاه الأسهم فقادتك إلى نفس النقطة التي بدأت منها فأنت في انتظار دائري بدون شك، مثلا في الشكل 5.4، إذا بدأت من أي نقطة **R** – مثلا وتحركت مع اتجاه الاسم ستصل مرة أخرى إلى **R** مما يدل على وجود انتظار دائري.



شكل رقم (' 5.4): انتظار دائري.

صيغة النظام :- System Model

النظام يتضمن **عدد محدود من المصادر** موزعة علي مجموعة عمليات تتنافس عليها.

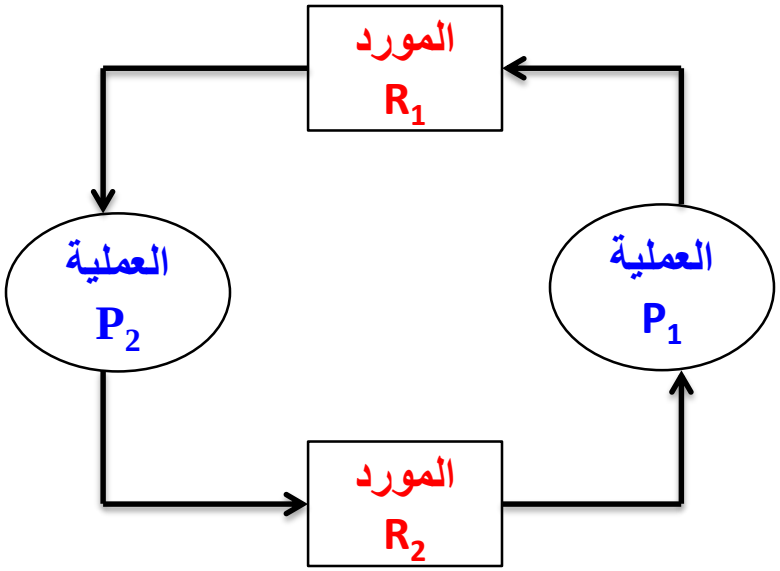
لاستخدام الموارد فان **الصيغة** تعني ان **التعامل مع الموارد يتم وفقا للتسلسل الاتي:**

- **الطلب (Request)** : لو الطلب **غير مضمون فوريا** بمعني أن المصدر **تستخدمه عملية أخرى** في الوقت الحالي اذاً العملية يجب أن تنتظر حتى يمكن أن تتحصل علي المصدر .
- **الاستخدام (Utilization) :** العملية يمكن أن تستغل المصدر بمعني لو كان المصدر طابعة مثلا إذاً العملية يمكنها أن تطبع على الطابعة .
- **التسريح (Release)** : العملية تسرح المصدر.

مفهوم الايصاد

في **البرمجة المتعددة (Multiprogramming)**، يقال لمجموعة من العمليات انها في حالة **ايصاد (Deadlock)** لو أن **كل عملية من المجموعة تنتظر مورد ما محجوز من قبل عملية اخرى في المجموعة**.

فإذا كان لدينا عمليتين P_1, P_2 ، وكانت العملية P_1 تستخدم عدة موارد (من بينها المورد R_1) وتحتاج إلى مورد إضافي هو المورد R_2، في نفس الوقت لدينا العملية P_2 التي تستخدم المورد R_2 وتحتاج إلى مورد آخر لتكمل تنفيذها هو المورد R_1 الذي بحوزة العملية P_1، هنا ستظل العمليتان في هذا الوضع إلى ما لا نهاية، فكل عملية لن تكتمل لأنها تحتاج مورد العملية الأخرى، هنا يحدث الايصاد لأن كل عملية تحتكر مواردها التي تستخدمها ولا تتركها أبدا.



مواصفات الايصاد (Deadlock Characteristics)

من الطبيعي ان تكون **حالة الايصاد حالة غير مرغوب فيها** ولذلك ينبغي معرفة الخصائص التي تودي الى حالة الايصاد، وهي تتمثل في **تحقق الاربعة شروط التالية دفعة واحدة** وفي نفس الوقت (أي مع بعضها البعض).

1. **الاقصاء المتبادل (Mutually exclusive) :** عملية واحدة فقط هي التي تستخدم مورد معين في اللحظة المعينة فلا يمكن لعمليتين استخدام نفس المورد في نفس الوقت.
2. **الاحتفاظ والانتظار (Hold and wait) :** تحتفظ العملية بمواردها التي تستخدمها وتريد موارد أخرى لتكمل عملها.
3. **الاحتكار أو عدم التخلي عن المورد (no preemption) :** لا يمكن انتزاع المورد من العملية التي تستخدمه ما لم يكتمل عملها طوعيا.
4. **الانتظار الدائري (Circular wait) :** يحدث الانتظار الدائري في سلسلة من العمليات إذا كانت مجموعة من العمليات $P_1, P_2, P_3, \ldots, P_n$، تنتظر مورد او اكثر وفقا للاتي – العملية P_1 تمتلك المورد R_1، و تنتظر المورد R_2 من العملية P_2، والعملية P_2 تمتلك المورد R_2، و تنتظر المورد R_3 المملوك من قبل العملية P_3، وهكذا حتي نصل الى اخر عملية P_n التي تمتلك المورد R_n، وتنتظر المورد R_1 المملوك من قبل العملية P_1، وبذلك بصبح لدينا حلقة من الانتظار (Circular wait)

التعامل مع الايصاد او الاختناق (Deadlock)

يمكن التعامل مع الاختناق بطريقتين:

- **الوقاية** وذلك بمنع حدوثه أو تجنبه.
- **العلاج** وذلك **بتجاهله** أو **إصلاحه**، طبعا التجاهل لا يحتاج كثير من العناء ، فكل ما على نظام التشغيل هو التظاهر بعدم حدوثه (وكأن شيء لم يحدث)أما الإصلاح فيحتاج مرحلة قبله هي **اكتشاف الاختناق أولا، ثم إصلاحه ثانيا.**

تجاهل الايصاد : خوارزمية النعامة (*The Ostrich Algorithm*)

أبسط خوارزمية تستخدم في تجاهل الاختناق هي **خوارزمية النعامة** (أدفن رأسك في الرمل وتظاهر بأنه لا توجد مشكلة)

التظاهر بعدم حدوث الاختناق أو الهروب من المشكلة نلجأ إليها لسببين هما:

- اذا كان الايصاد نادر الحدوث.
- إذا كان علاجه مكلفا.

اكتشاف الايصاد : هنالك طرق عديدة لاكتشاف الايصاد.

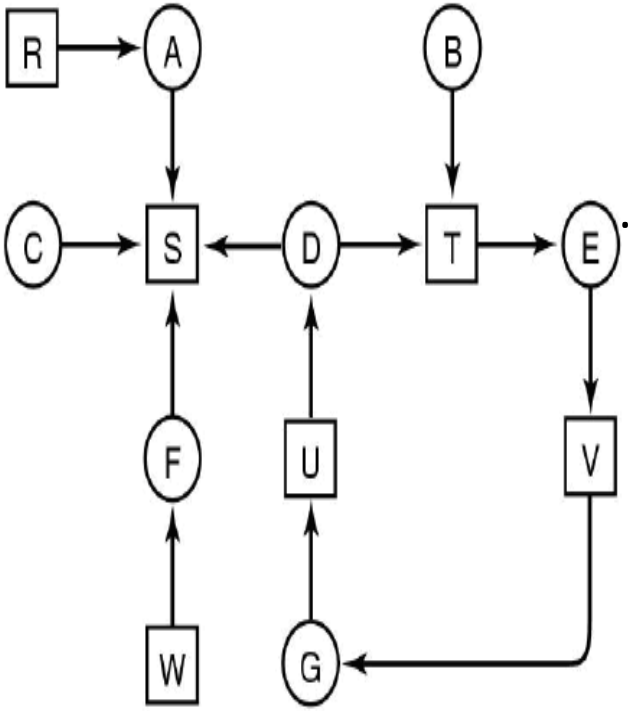
اكتشاف الايصاد **لمورد واحد من كل نوع** هنا سنستخدم **الرسم** لاكتشاف الايصاد في نظام به مورد من كل نوع (مثلا طابعة واحدة، ماسحة واحدة،.. الخ.)

مثال :إذا كان لدينا **7 عمليات** من **A** إلى **G** و **مورد** واحد من كل نوع، وكانت طلبات العمليات كما يلي:

- العملية **A** تستخدم المورد **R** وتريد المورد **S** لإتمام عملها.
- العملية **B** لا تستخدم أي مورد وتريد المورد **T** لإتمام عملها.
- العملية **C** لا تستخدم أي مورد وتريد المورد **S** لإتمام عملها.
- العملية **D** تستخدم المورد **U** وتريد المورد **S** و المورد **T** لإتمام عملها.
- العملية **E** تستخدم المورد **T** وتريد المورد **V** لإتمام عملها.
- العملية **F** تستخدم المورد **W** وتريد المورد **S** لإتمام عملها.
- العملية **G** تستخدم المورد **V** وتريد المورد **U** لإتمام عملها.

نريد أن نعرف هل سيكون هنالك اختناق أم لا(اكتشاف الايصاد)؟

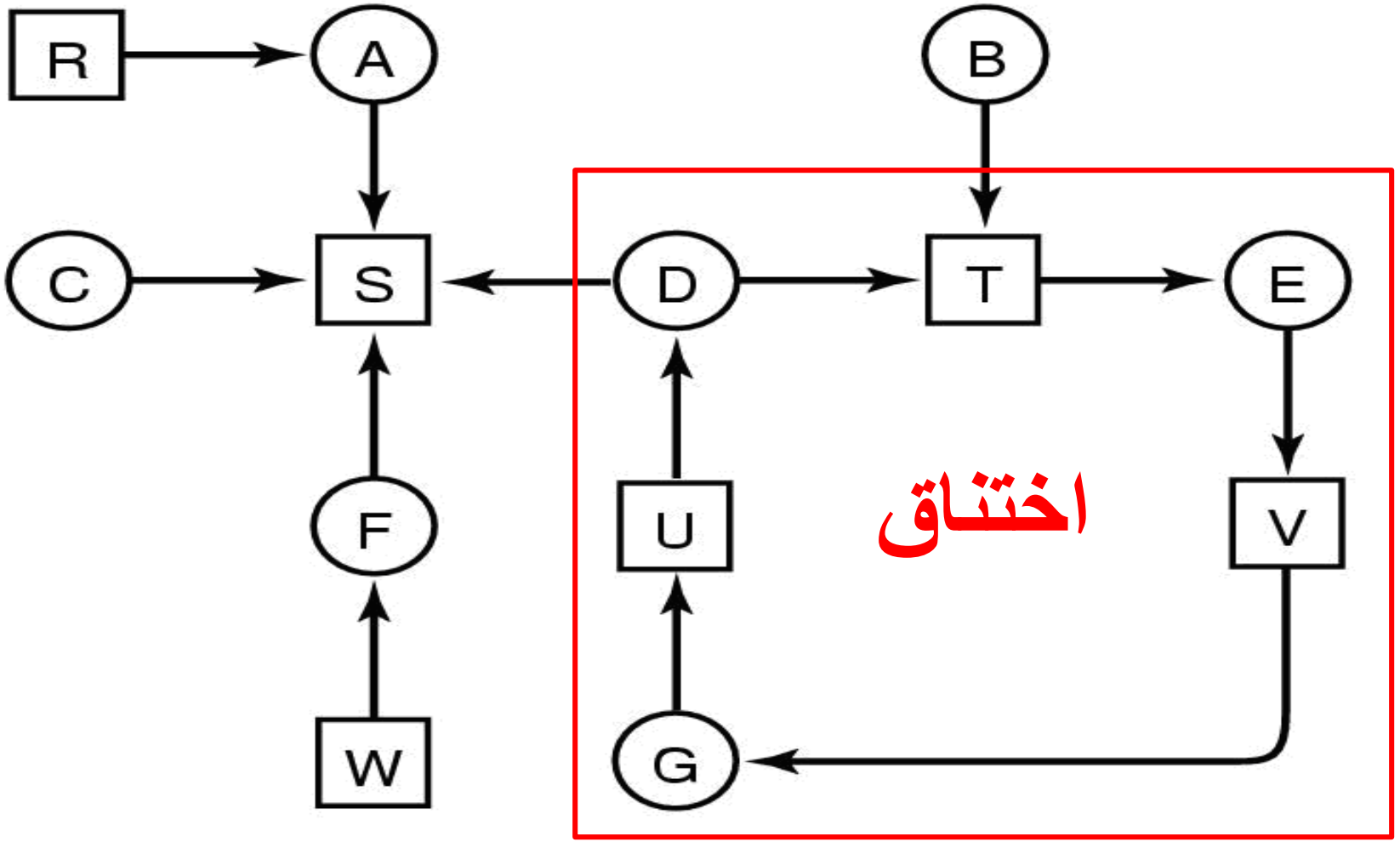
الحل : سنقوم بإنشاء رسم لكل الموارد مع العمليات التي تستخدمها والتي تريدها، فيكون الرسم كما في **الشكل 5.7**



5.5

شكل رقم (7-5): رسم يبين الموارد.

نلاحظ من الشكل (7-5)، ان **هناك اختناق**، لأنني مثلا اذا بدأت من المورد **U** ثم تتبعت اتجاه الأسهم سأجد نفسي في **U** مرة أخرى وهذا يعني وجود انتظار دائري ، مع وجود مسببات الاختناق الأخرى.



اكتشاف الاختناق لعدة موارد

قد يكون لدينا عدد من الموارد من كل نوع (مثلا **5 طابعات، 4 ماسحات**، ...الخ)، في هذه الحالة يصعب استخدام الرسومات للتعبير عن الموارد المتعددة لذلك سنستخدم **طريقة المحددات (Matrix)** لاكتشاف الاختناق.

فاذا كان لدينا عدد من العمليات P_n وعدد من الموارد R_n $(R_1\ R_2 \dots R_n)$ حيث R_i تمثل مجموعة من الموارد ذات النوع الواحد.

مثـــال

إذا كان لدينا **7 طابعات، 6 ماسحات، 5 أقراص**، فإن المتجه الذي يمثل هذه **الموارد الكلية** سيكون كالتالي :

طابعات	ماسحات	اقراص
[5	6	7]

فاذا استخدمت **العمليات** عدد **5 طابعات، 4 ماسحات، و 3 أقراص**، فإن **المتجه** الذي يمثل **الموارد المستخدمة** سيكون:

طابعات	ماسحات	اقراص
[3	4	5]

فتكون الموارد الحرة او المتاحة

[2	2	2]

محددة الاستخدام (الموارد التي تستخدمها العمليات حاليا)

يمكننا تكوين محددة للموارد المستخدم بواسطة عدة عمليات، حيث يمثل كل صف في المحددة موارد عملية معينة. مثلا لو أردنا إنشاء محددة لأربع عمليات كانت تستخدم الموارد أعلاه، فإن المحددة ستكون كما يلي:

	نوع المورد		
	الطابعات	الماسحات	الأقراص
الموارد التي تستخدمها العملية الأولى	5	1	1
الموارد التي تستخدمها العملية الثانية	0	1	1
الموارد التي تستخدمها العملية الثالثة	0	1	0
الموارد التي تستخدمها العملية الرابعة	0	1	1

P1	1
P2	1
P3	0
P4	1

من الشكل نجد أن:

- العملية الأولى تستخدم 5 طابعات، ماسحة، وقرص.
- العملية الثانية تستخدم ماسحة وقرص.
- العملية الثالثة تستخدم ماسحة واحدة فقط.
- العملية الرابعة ماسحة وقرص.

محددة الطلبات (الموارد التي تحتاجها العمليات)

بنفس الطريقة يمكن إنشاء محددة للموارد التي تحتاجها كل عملية من كل مورد، كما في المثال التالي:

	الموارد التي تحتاجها العملية	الطابعات	الماسحات	الأقراص
P1	الموارد التي تحتاجها العملية الأولى	2	2	2
P2	الموارد التي تحتاجها العملية الثانية	5	6	7
P3	الموارد التي تحتاجها العملية الثالثة	4	3	3
P4	الموارد التي تحتاجها العملية الرابعة	4	4	4

من الشكل نجد أن:

- العملية الأولى تحتاج إلى طابعتين وماسحتين وقرصين.
- العملية الثانية تحتاج إلى 5 طابعات، 6 ماسحات و7 أقراص.
- العملية الثالثة تحتاج 4 طابعات، 3 ماسحات، و 3 أقراص.
- العملية الرابعة تحتاج إلى 4 طابعات، 4 ماسحات، و 4 أقراص.

اكتشاف الايصاد باستخدام المحددات او المصفوفات

في المثال السابق فان

الشكل 1 →

5	6	7

الموارد الكلية

الشكل 2 →

P1	1	1	5
P2	1	1	0
P3	0	1	0
P4	1	1	0

الموارد المستخدمة

الشكل 3 →

	2	2	2

الموارد الحرة او المتاحة

الشكل 4 →

P1	2	2	2
P2	7	6	5
P3	3	3	4
P4	4	4	4

الموارد المطلوبة

طريقة الحل

سنقوم أولا بمقارنة الموارد الحرة مع صفوف محددة الموارد المطلوبة، ثم نختار الصف (العملية) الذي تكون موارده المطلوبة أقل أو تساوي الموارد الحرة المتوفرة. بمقارنة صفوف المحددة في الشكل **4** مع الموارد الحرة في الشكل **3** سنجد أن العملية الأولى يمكنها إكمال عملها باستخدام الموارد الحرة المتاحة. سنعطي العملية الأولى ما تحتاج من الموارد المتاحة ونخصم ذلك من الموارد الحرة كما يلي:

(2 2 2) = الموارد الحرة

(2 2 2) = الموارد التي تحتاجها العملية الأولى

(0 0 0) = المتبقي من الموارد الحرة.

الشكل 5

إذا كانت الموارد الحرة تكفي لواحد من عمليتين، فالأفضل اختيار العملية التي تستخدم موارد أكثر ذلك لأني سأحصل على موارد حرة أكثر بعد أن تكمل العملية عملها وتحرر ما تستخدم من موارد.

الآن ستكمل العملية الأولى عملها ثم تحرر ما لديها من موارد (التي كانت تستخدم مسبقا والتي إعطيناها لها في الخطوة السابقة)، بالتالي سيكون لدي من الموارد الحرة ما يلي:

(0 0 0) = الموارد الحرة المتبقية بعد إعطاء العملية الأولى ما تحتاج من موارد

(5 1 1) = الموارد التي كانت تستخدمها العملية الأولى من قبل (من محددة الاستخدام)

(2 2 2)= الموارد التي استخدمتها العملية الأولى مؤخرا (من محددة المطلوب)

(7 3 3) = الموارد الحرة بعد انتهاء العملية الأولى وتحريرها لمواردها

الشكل 6

الأن سنقارن الموارد الحرة التي حصلنا عليها بعد انتهاء العملية الأولى وتحرير مواردها مع المتبقي من صفوف محددة المطلوب (شكل 6). سنجد أن العملية التي يمكن أن تكفيها الموارد الحرة هي العملية الثالثة، فنخصم الموارد التي نريد من الموارد الحرة التي لدينا كما في الشكل (7)

(7 3 3) = الموارد الحرة

(4 3 3) = الموارد التي تحتاجها العملية الثالثة

(3 0 0) = المتبقي من الموارد الحرة.

شكل رقم 7

الآن ستكمل العملية الثالثة عملها ثم تحرر ما لديها من موارد (القديم والجديد)، بالتالي سيتوفر لدي من الموارد التالي (بعد انتهاء العملية الثالثة):

(3 0 0) = الموارد الحرة المتبقية بعد إعطاء العملية الأولى ما تحتاج من موارد

(0 1 0) = الموارد التي كانت تستخدمها العملية الثالثة من قبل (من محددة الاستخدام)

(4 3 3)= الموارد التي استخدمتها العملية الثالثة مؤخرا (من محددة المطلوب)

(7 4 3) = الموارد الحرة بعد انتهاء الثالثة وتحريرها لمواردها

شكل رقم **8**

الآن سنقارن الموارد الحرة التي حصلنا عليها بعد انتهاء العملية الثالثة وتحرير مواردها كما في الشكل (**8**)، مع المتبقي من العمليات التي بمحددة المطلوب (شكل **4**). سنجد أن الموارد المتوفرة لدينا لا تكفي لأي عملية من العمليات المتبقية. وبالتالي يمكننا القول أن هنالك اختناق حدث في هذه النقطة.

P1	2	2	2
P2	7	6	5
P3	3	3	4
P4	4	4	4

مثال (2)

الآن سنعرض مثالا آخر بدون شرح مفصل يوضح حالة لا يحدث فيها اختناق.

المعطيات:

الموارد الموجودة بالنظام =(4 2 3 1)

المستخدم منها =

$$\begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 \end{vmatrix}$$

الموارد المطلوبة =

$$\begin{vmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \end{vmatrix}$$

المطلوب: هل يوجد اختناق أم لا ؟

الحل

أولا نحسب الموارد المتوفرة وذلك بطرح الموارد المستخدمة (مجموع أعمدة محددة الموارد المستخدمة) من الموارد كما يلي:

موارد النظام	(4	2	3	1)
الموارد المستخدمة	(2	1	3	1)
الموارد المتوفرة	(2	1	0	0)

ثم نقارن المتاح مع صفوف المطلوب (في محددة المطلوب)، فنجد أن الصف الثالث يمكنه العمل ، فنعطيه ما يريد من موارد كما يلي:

الموارد المتوفرة =	(2	1	0	0)
الموارد المطلوبة (صف 3)	(2	1	0	0)
الموارد المتوفرة بعد إعطاء صف 3 موارده	(0	0	0	0)

بعد انتهاء العملية صف 3، سنستفيد من موارده، فيصبح لدينا الموارد المتاحة التالية:

الموارد المتوفرة بعد إعطاء صف 3 موارده	(0	0	0	0)
موارد صف3 في محددة المستخدم	(0	1	2	0)
موارد صف3 في محددة المطلوب	(2	1	0	0)
الموارد المتوفرة بعد انتهاء صف 3 من عمله	(2	2	2	0)

الموارد المطلوبة =

1	0	0	2
0	1	0	1
0	0	1	2

الآن سنقارن المتاح مع صفوف المطلوب (في محددة المطلوب)، فنجد أن الصف الثاني يمكنه العمل ، فنعطيه ما يريد من موارد كما يلي:

الموارد المتوفرة =	(0	2	2	2)
الموارد المطلوبة (صف 2)	(0	1	0	1)
الموارد المتوفرة بعد إعطاء صف 2 موارده	(0	1	2	1)

بعد انتهاء العملية صف 2، سنستفيد من موارده، فيصبح لدينا الموارد المتاحة التالية:

الموارد المتوفرة بعد إعطاء صف 2 موارده	(0	1	2	1)
موارد صف2 في محددة المستخدم	(1	0	0	2)
موارد صف2 في محددة المطلوب	(0	1	0	1)
الموارد المتوفرة بعد انتهاء صف 2 من عمله	(1	1	2	4)

الموارد المطلوبة =

1	0	0	2
0	1	0	1
0	0	1	2

نجد أن الصف المتبقي في محددة المطلوب تكفيه الموارد المتوفرة بعد انتهاء صف2، وهذا يعني أنه لا يوجد اختناق.

معالجة الايصاد

الخطوة السابقة وضحت كيف يمكننا اكتشاف الايصاد ، هنا سنقوم بمعالجة الايصاد بعد إكتشافه، ويتم ذلك بعدة طرق منها:

- انتزاع بعض الموارد من أحد عمليات الايصاد : يجب التأكد من أن الموارد قابل للنزع، بحيث لا يسبب نزعه مشكلة للعملية، مثلا إذا كانت العملية تقوم بتسجيل بيانات على اسطوانة ضوئية (تستخدم مسجل الأسطوانات الضوئية (CD writer) فإن انتزاع هذا المورد من العملية في هذه اللحظة قد يتسبب في تعطل الاسطوانة. كذلك انتزاع طابعة من عملية وهي تطبع قد يجعل مخرج الطابعة مبتور وغير واضح. لكن إذا كان الاختناق على الذاكرة فيمكن تحويل أحد العمليات المتسببة في الاختناق إلى القرص الصلب (انتزاع الذاكرة من العملية) ، وإرجاعها فيما بعد دون حدوث مشكلة.

- إيقاف بعض العمليات التي تشارك في الاختناق: يمكن توقيف عملية (قتلها كما يقال في لينكس (kill)) والاستفادة من مواردها لتشغيل بقية العمليات. في هذه الحالة سنحاول قتل العمليات التي يمكن تشغيلها من جديد بسهولة ودون أن تسبب مشاكل.

الوقاية

يمكننا أن نقي النظام من حدوث الاختناق وذلك بطريقتين:

- تجنبه
- منعه

تجنب الاختناق باستخدام خوارزمية المصرف Banker's Algorithms

هنا يحاول النظام التأكد من أن الموارد يمكن حجزها بصورة آمنة، وذلك قبل الشروع في حجزها. هنالك العديد من الخوارزميات التي يمكن استخدامها للتأكد من أن الموارد يمكن حجزها بدون أن يحدث اختناق (صورة آمنة safe state).

مثال: اذا كان لدينا موارد ، منها المستخدم ومنها غير المستخدم (الحر) ، وهنالك عمليات تستخدم موارد وتريد المزيد ، فيمكننا استخدام الطريقة التالية من حجز الموارد الحرة (**Free**) للعمليات التي تريد موارد لا يسبب اختناق، مثلا الشكل (7.15) ثلاث عمليات (**C, B, A**)، وكل عملية تستخدم عدد معين من الموارد وتحتاج أن تصل إلى عدد معين من الموارد لتعمل.

Max	Has	Process
9	3	A
4	2	B
7	2	C

Free=3

الشكل 7.15

العملية **A** تستخدم **3** موارد وتحتاج أن تصل إلى **9** موارد، العملية **B** لديها **موردين** وتريد أن تصل إلى **4** موارد، العملية **C** لديها **موردين** وتريد أن تكمل مواردها إلى **7** موارد لتعمل .و لدينا **ثلاث موارد متاحة** .(**Free=3**) نريد التأكد من أننا لو استخدمنا هذه الموارد المتوفرة يمكن أن نصل إلى حالة آمنة (لا يحدث ايصاد.)

الحل

- بمقارنة **المتوفر (Free=3)** مع **المطلوب (Max-Has)**، نجد أن العملية **B** يمكن أن تعمل وذلك بإعطائها **موردين** من المتوفر فيكون لدينا مورد واحد متوفر (**Free=1**) كما في الخطوة رقم 1.
- بعد انتهاء العملية **B** ستترك الموارد التي كانت تستخدم 4 موارد، فيصبح المتوفر هو (**Free=5**)، كما في الخطوة رقم 2.
- بمقارنة المتوفر مع المطلوب، نجد أن العملية **C** يمكن أن تعمل وذلك بإعطائها **5 موارد** وهي كل ما يتوفر لدينا كما في الخطوة 3.
- بعد انتهاء العملية **C** ستترك الموارد التي كانت تستخدم 7 موارد، فيصبح المتوفر هو (**Free=7**)، كما في الخطوة 4.
- بمقارنة المتوفر مع المطلوب، نجد أن العملية **A** يمكن أن تعمل وذلك بإعطائها **6 موارد** ويبقى مورد واحد (**Free=1**) غير مستخدم كما في الخطوة 5.
- بعد انتهاء العملية **A** ستترك الموارد التي كانت تستخدم **9 موارد**، فيصبح المتوفر هو (**Free=10**)، كما في الخطوة 6.

هذا يدل أن الموارد يمكن استخدامها دون حدوث ايصاد (**حالة آمنة.**)

Max	Has	Process
9	3	A
4	4	B
7	2	C

Free=1
الخطوة رقم 1

Max	Has	Process
9	3	A
-	-	B
7	2	C

Free=5
الخطوة رقم 2

Max	Has	Process
9	3	A
-	-	B
7	7	C

Free=0
الخطوة رقم 3

Max	Has	Process
9	3	A
-	-	B
-	-	C

Free=7
الخطوة رقم 4

Max	Has	Process
9	9	A
-	-	B
-	-	C

Free=1
الخطوة رقم 5

Max	Has	Process
-	-	A
-	-	B
-	-	C

Free=10
الخطوة رقم 6

مثال

إذا لم نستخدم الموارد بحذر لتجنب الايصاد ، فقد نصل إلى حالة **غير آمنة (unsafe)** مما يتسبب في ايصاد .المثال السابق يمكن استخدامه لتوضيح كيف يمكن أن يحدث ايصاد (باعتبار ان المتوفر حاليا هو **Free=2**).

Max	Has	Process
9	3	A
4	2	B
7	2	C

Free=2

الحل

- بمقارنة **المتوفر (Free)** مع **المطلوب (Max-Has)**، نجد أن العملية **B** يمكن أن تعمل وذلك بإعطائها **موردين** من المتوفر فيصبح المتوفر **(Free=0)** كما في الخطوة رقم 1.
- بعد انتهاء العملية **B** ستترك الموارد التي كانت تستخدم 4 موارد، فيصبح المتوفر هو **(Free=4)**، كما في الخطوة رقم 2.
- بمقارنة المتوفر مع المطلوب، نجد أنه لا يوجد أي عملية اخرى يمكنها اكمال عملها و بالتالي نصل الى **حالة ايصاد.**

Max	Has	Process
9	3	A
4	4	B
7	2	C

Free=0
الخطوة رقم 1

Max	Has	Process
9	3	A
-	-	B
7	2	C

Free=4
الخطوة رقم 2

منع الاختناق

تجنب الايصاد قد يكون غير ممكن لأنه يتطلب دراية مسبقة بالطلبات المستقبلية والتي لا يمكن معرفتها في غالبا الأحيان، لذلك سنجد أن الوقاية من الايصاد يمكن أن **تتحقق بنفي واحد أو أكثر من مسببات الايصاد.**

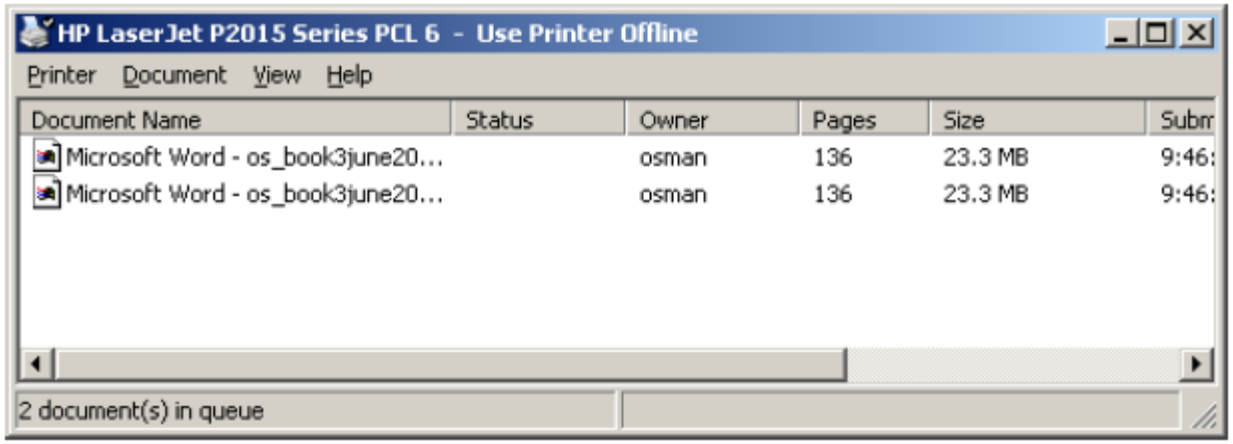
فمن المعلوم أن مسببات الايصاد لابد من توافرها جميعها **(أربعة مسببات)** حتى يحدث الايصاد، بالتالي إذا استطعنا نفي سبب واحد من هذه المسببات الأربع سنكون وقينا أنفسنا من الاختناق.

- **نفي الاقصاء المتبادل (Mutually Exclusion)**

إذا كان لدينا مورد لا يقبل المشاركة ولا يمكن استخدامه بأكثر من عملية في نفس الوقت، **يمكنني تجنب الاحتكار وذلك بتخصيص عملية تكون مسئولة عن هذا المورد**، بحيث تمنع بقية العمليات من استخدام المورد مباشرة وإنما تتعامل هذه العمليات مع العملية المسئولة من المورد، وتقوم الأخيرة بالتعامل مع المورد.

مثلا مدير الطباعة في ويندوز يمثل عملية مسئولة عن الطابعة، حيث عندما يرسل أي برنامج (عملية) أمر طباعة، فإن الملف المراد طباعته سيرسل إلى مدير الطباعة ، ويتعامل مدير الطباعة مع الطابعة حيث يرسل لها البيانات كلما فرغت من مهمة (نجد أن - عملها حسب ورود الملفات إليه) القادم أولا يخدم أولا (. مثلا الشكل) 7

هنالك ملفين أرسلا للطابعة ولكن قام باستلام هذه الملفات مدير الطباعة ثم ينظم هو التعامل مع الطابعة.



- **نفي الاحتفاظ والانتظار (Hold and wait)**

لا تبدأ العملية في العمل ما لم تتأكد من أن كل الموارد التي تحتاجها متوفرة وغير مستخدمة من قبل عمليات أخرى .ولكن هنا تظهر مشكلة هي أن معظم العمليات لا تستطيع حجز ما تريد من الموارد ما لم تبدأ العمل .أيضا هنالك مشكلة أخرى هي أنه حتى ولو توفرت كل الموارد للعملية وبدأت عملها فهذا يجعل الموارد محتكرة لأن العملية قد تحتاج مورد لعمل ما وبعد نصف ساعة ستحتاج المورد الثاني، فتكون العملية قد حجزت هذا المورد الأخير لمدة نصف ساعة دون أن تستفيد منه بقية الموارد.

- **الاحتكار أو عدم التخلي عن المورد (no preemption)**

غير قابل للتطبيق :لان انتزاع مورد من عملية وهي تستخدمه أحيانا قد يتسبب في فشل العملية. مثلا إيقاف الطابعة وهي تطبع في ملف قد ينتج عنه طباعة غير مكتملة، وإيقاف مسجل الاسطوانات الضوئي (CD writer وهو ينسخ بيانات في اسطوانة CD قد يتسبب في تلف الاسطوانة وفشل عملية النسخ.

- **نفي الانتظار الدائري**

نرقم كل الموارد، ثم نسمح للعمليات بطلب الموارد بصورة تصاعدية ولا نسمح لعملية بأن تطلب مورد رقمه أصغر من المورد الذي طلبته من قبل، فمثلا إذا طلبت عملية ما المورد رقم2 ، فلا نسمح لها بطلب مورد رقمه أقل من 2 ويمكنه طلب مورد رقمه أكبر من 3.

مثال

في الشكل التالي ، إذا طلبت العملية A **الماسحة** ورقمها هو 2، فلا يمكن لهذه العملية أن تطلب **الطابعة** لأن رقمها أقل من رقم **الماسحة**. افترض أن العملية B قامت بطلب **الشريط الممغنط** (الرقم 4)، فلا يمكن لهذه العملية طلب أي مورد رقمه أقل من 4، يعني لا يمكنها طلب **الماسحة** التي تستخدمها العملية A بهذه الطريقة لا يمكن أن يحدث انتظار دائري.

المورد	التسلسل
الطابعة	1
الماسحة	2
سواقة الاقراص	3
الشريط الممغنط	4
الراسم	5

تمارين محلولة

1- اذكر ثلاث أمثلة لموارد ؟
الطابعة ، القرص الصلب ، جدول قاعدة بيانات

2- تنقسم الموارد إلى نوعين ما هما ؟
قابلة للنزع وغير قابلة للنزع

3- اذكر مثال لمورد قابل للنزع ومثال لمورد غير قابل للنزع ؟
الطابعة (غير قابلة)، الذاكرة (قابل للنزع)

4- هنالك أربع طرق للتعامل مع الايصاد ، اذكرها ؟
- تجاهله
- اكتشافه وعلاجه
- تجنبه
- منعه

5- ما الفرق بين التجنب والمنع ؟
التجنب هو حجز الموارد بصورة آمنة (safe) ، المنع هو نفي احد مسببات الايصاد

6- كيف يتم منع الايصاد؟

- نفي المنع المتبادل
- نفي الاحتفاظ والانتظار
- نفي الاحتكار
- نفي الانتظار الدائري

7- ما هي مسببات الايصاد الأربعة ؟

- المنع المتبادل
- الاحتفاظ والانتظار
- الاحتكار
- الانتظار الدائري

8- ما هي خوارزمية النعامة ؟ ومتى نستخدمها ؟

- هي التظاهر بأنه لا يوجد ايصاد ، ونستخدمها إذا كان :
- الاختناق يحدث نادرا
- تكلفة الوقاية من الاختناق عالية جدا