الباب السابع
الذاكرة الظاهرية

تعريف الذاكرة الظاهرية

نظم التشغيل الحديثة تستخدم جزء من **القرص الصلب كامتداد للذاكرة الرئيسية**. حيث يحمل جزء من البرنامج في **الذاكرة الرئيسية** ، وباقي أجزاء البرنامج **تحمل في جزء من القرص الصلب** (إذا لم يكن هنالك متسع لتحميله كاملا في الذاكرة الرئيسية). في هذه الحالة يعتبر **هذا الجزء من القرص الصلب امتداداً للذاكرة الرئيسية** ويسمى **الذاكرة الظاهرية** وبهذا تكون هذه النظم قد حلت مشكلة ارتباط حجم البرنامج بحجم المساحة المتوفرة من الذاكرة الرئيسية.

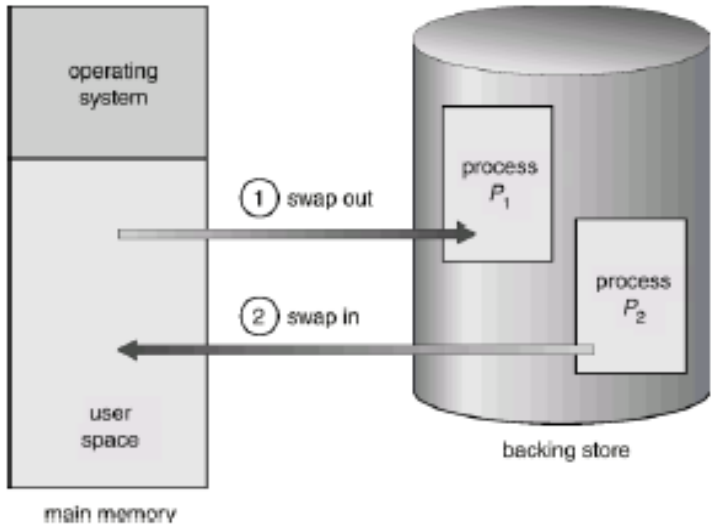
هنالك أسماء متعددة تستخدم للذاكرة الظاهرية (virtual memory)، فقد يطلق عليها الذاكرة الافتراضية أو الذاكرة الخيالية أو الذاكرة الوهمية، كلمات كثيرة لمعنى واحد.

يبدأ المعالج في تنفيذ جزء البرنامج الموجود بالذاكرة، وإذا أحتاج إلى أوامر أو بيانات من الجزء الآخر (الموجود بالقرص) سيقوم مدير الذاكرة بتبديل الجزأين (swap).

المبادلة *Swapping*

المبادلة هي تحويل البرنامج (أو جزء منه) من الذاكرة الرئيسية إلى القرص أو العكس. تتم المبادلة على مرحلتين، الشكل (7-1)، هما:

- تحويل البرنامج (أو جزء منه) من الذاكرة إلى القرص (swap out).
- تحميل البرنامج (أو جزء منه) من القرص إلى الذاكرة (swap in).



شكل رقم (7-1) / المبادلة

الذاكرة الظاهرية

تعتبر الذاكرة الظاهرية فصل بين ذاكرة المستخدم المنطقية والذاكرة الحقيقية وتختلف عن الذاكرة الحقيقية في الآتي:

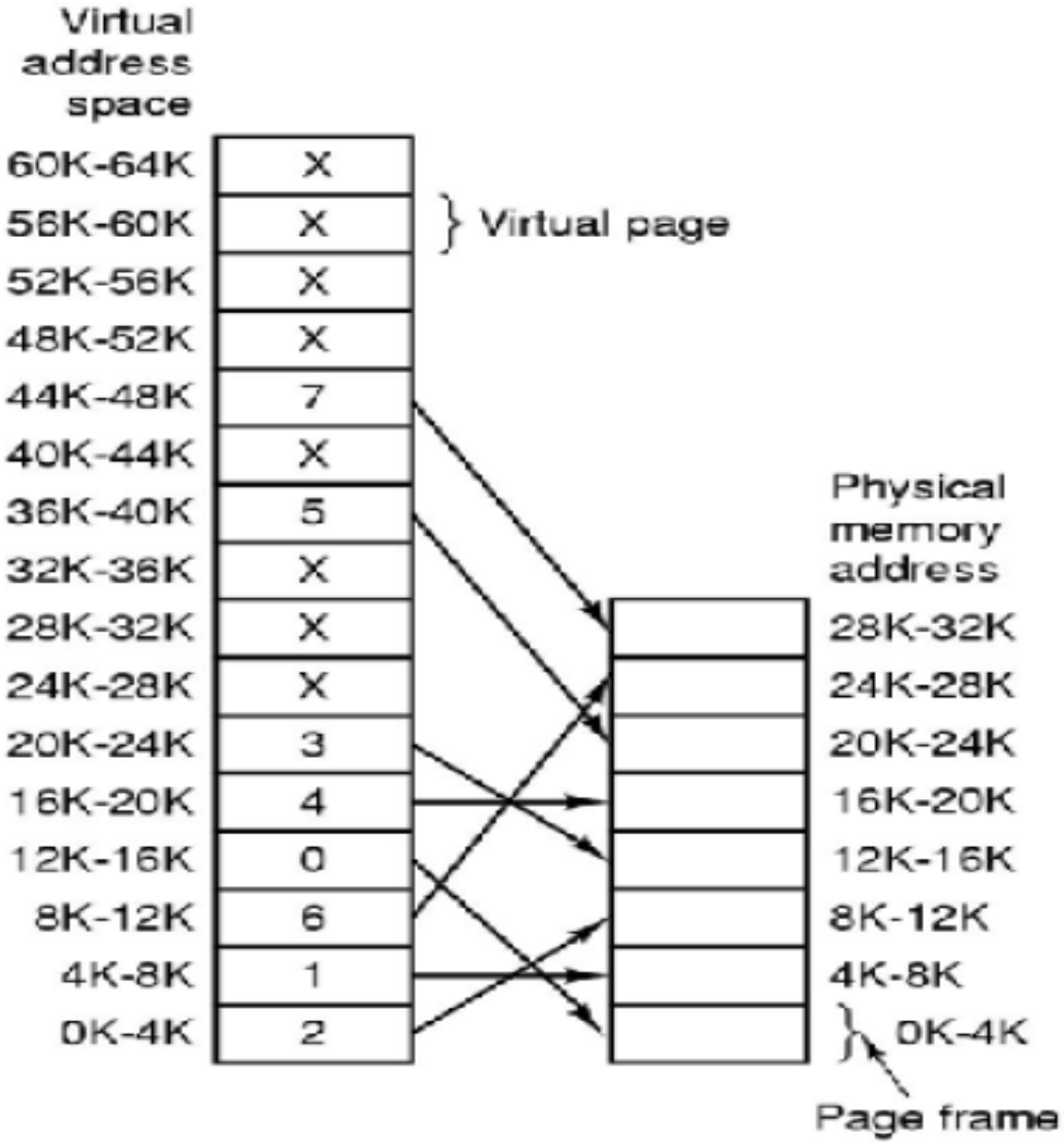
- فقط جزء من البرنامج يكون بالذاكرة.
- قد يكون مجال العنوان المنطقي أكبر من مجال العنوان الحقيقي.
- يمكن لعدة عمليات التشارك في مجال عناوين.

يمكن تطبيق الذاكرة الظاهرية في :

- إدارة الذاكرة بالصفحات.
- إدارة الذاكرة بالمقاطع.

• الذاكرة الظاهرية في الصفحات

إذا كان صفحات البرنامج أقل أو تساوي الإطارات بالذاكرة فلن نحتاج إلى ذاكرة ظاهرية. ولكن نحتاج للذاكرة الظاهرية إذا كانت صفحات البرنامج أكبر من الإطارات المتوفرة بالذاكرة. مثلا في الشكل (7-2) نجد أن الذاكرة مقسمة إلى 8إطارات، والبرنامج مقسم إلى 16صفحة. سيكون هنالك 8 صفحات بالذاكرة من البرنامج (المرقمة) والبقية ستكون في القرص الصلب (ذاكرة ظاهرية – امامها علامة X).



شكل رقم (2-7) الصفحات.

التعامل مع الذاكرة الظاهرية:

- إحضار الصفحة التي نحتاجها إلى الذاكرة.
- نبحث عن الصفحة المطلوبة المطلوبة في جدول الصفحات
  - إذا كانت الصفحة غير صحيحة ، نتوقف
  - إذا كانت ليست بالذاكرة، نحضرها إلى الذاكرة
- لا نحضر صفحة إلى الذاكرة ما لم نحتاجها

هنالك خوارزميات عديدة يستخدمها مدير الذاكرة ليحدد أي صفحة يخرج من الذاكرة عند ما نريد إطار فارغ لتخزين الصفحة المطلوبة به (فمن الصفحة الضحية)؟

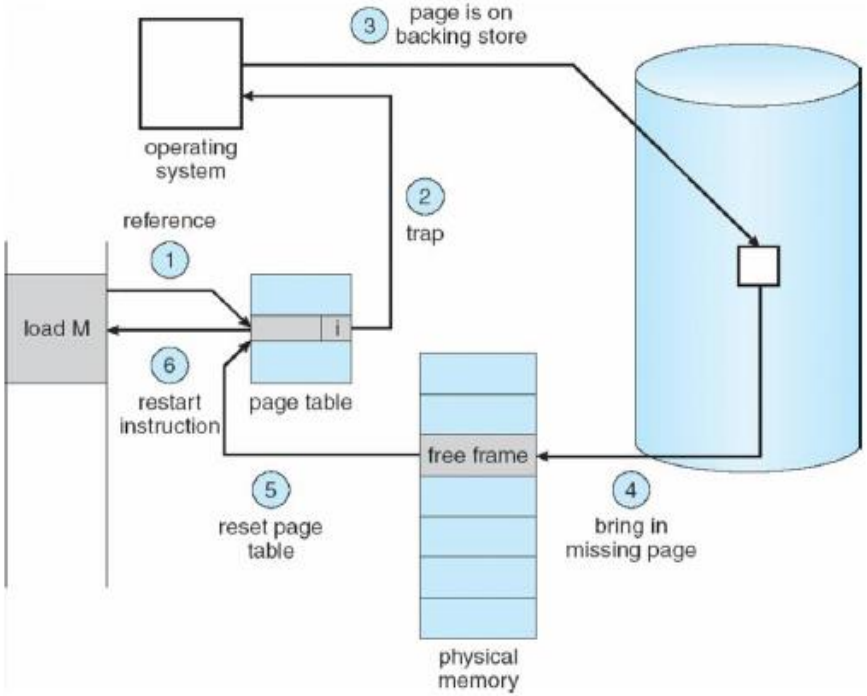
ملاحظة

قبل إخراج الصفحة (استخدام مكانها)، سيقوم مدير الذاكرة باختبار هل الصفحة المراد استخدام مكانها قد تم تعديلها أم لا ؟ إذا تم تعديلها يجب أن تحفظ بالقرص، أما إذا لم يتم تعديلها فنكتفي بنسختها القديمة الموجودة بالقرص.

خطأ صفحة (page fault)

عندما يطلب برنامج صفحة غير موجودة بالذاكرة (إفتقاد صفحة)، سيسبب ذلك قفز إلى نظام التشغيل برسالة تقول أن هنالك خطأ صفحة page fault ، أي أن البرنامج يريد صفحة ولم يجدها بالذاكرة. على نظام التشغيل توفير الصفحة المفتقدة للبرنامج الذي إفتقدها حسب الخطوات التالية:

- إذا كانت الصفحة ليست صحيحة، يتوقف abort.
- إذا كانت الصفحة صحيحة لكنها ليست بالذاكرة:
  - الحصول على إطار فارغ.
  - وضع الصفحة المطلوبة في هذا الإطار الفارغ.
  - تحديث جداول الصفحات بوضع رقم الصفحة ورقم الإطار الذي وضع بها.
  - تعديل بت التصحيح إلى **v** ، وهذا يعني أن الصفحة صحيحة.
- إعادة تنفيذ الأمر الذي سبب خطأ صفحة page fault.



ماذا يحدث إذا لم نجد إطار فارغ ؟

إذا طلبنا صفحة غير موجودة بالذاكرة ولم يجد نظام التشغيل إطار فارغ لإحضار الصفحة المطلوبة فيه، ماذا يفعل نظام التشغيل ؟

سيقوم بعملية استبدال (swap)، أي إخراج صفحة من الذاكرة (نسميها الصفحة الضحية) وإدخال الصفحة المطلوبة مكانها.

هنا يظهر سؤال هام وهو أي صفحة سيخرج من الصفحات التي بالذاكرة (من ستكون الصفحة الضحية) ؟ هنالك عدة خوارزميات تحدد أي صفحة سنخرج.

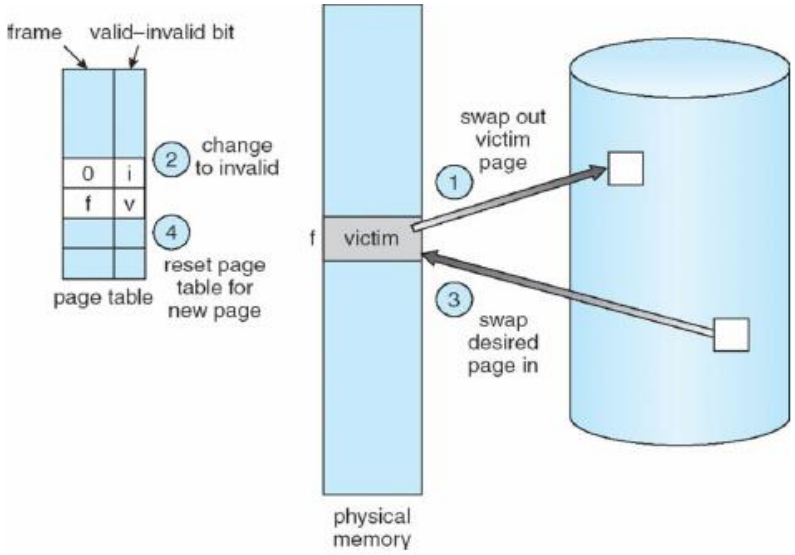
يستخدم نظام التشغيل أحد هذه الخوارزميات، مع وضع إعتبار للأداء في اختيار الخوارزمية (نريد خوارزمية تخرج صفحة لا نحتاجها في القريب العاجل وهذا يقلل عدد أخطاء الصفحات page faults).

استبدال الصفحات Page Replacement

نستفيد من استبدال الصفحات في استخدام إطارات قليلة لتنفيذ عملية بصفحات كثيرة، وهي الفكرة الرئيسية في الذاكرة الظاهرية.

خطوات استبدال الصفحة:

1. أبحث عن موقع الصفحة في القرص
2. أبحث عن إطار فارغ، إذا وجد إطار نستخدمه.
3. إذا لم يوجد إطار فارغ، نستخدم خوارزمية لإختيار الإطار الضحية **victim** frame.
4. إخراج الصفحة الضحية من الإطار الضحية.
5. إحضار الصفحة المطلوبة إلى الإطار الضحية.
6. تعديل جدول الصفحات.
7. إعادة تنفيذ الأمر الذي سبب خطأ الصفحة.



خوارزميات استبدال الصفحات *Page Replacement Algorithms*

هنالك العديد من الخوارزميات التي تستخدم في استبدال الصفحات منها:

- إخراج الصفحة الأقدم (الأول أولا تخرج أولا)، first come first out (fifo)
- إخراج الصفحة التي لا نحتاج لها قريبا (الخوارزمية المثلى optimal)
- خوارزمية الصفحة الاقل استخدام LRU
- خوارزمية LRU التقريبية approxmination
- خوارزمية الساعة.
- خوارزمية الفرصة الثانية.
- خوارزمية التعداد
- نريد أقل معدل page-fault

خوارزمية الصفحة الأقدم تحميلا (FIFO)

مثال

إذا كان لدي سلسلة طلبات الصفحات التالية:

⇨ 1,2,3,2,1,5,2,1,6,2,5,6,3,1,3,6,1,2,4,3

ما هي عدد خطأ الصفحات (page fault) إذا استخدمنا 3 إطارات فارغة.

الحل

1,2,3,2,1,5,2,1,6,2,5,6,3,1,3,6,1,2,4,3
1 1 1 2 3 5 1 6 2 5 3 1 6 2
2 2 3 5 1 6 2 5 3 1 6 2 4
3 5 1 6 2 5 3 1 6 2 4 3
---
3 1 1 1 1 1 1 1 1 1 1

خطأ الصفحة = 14 =page fault

الخوارزمية المثلى Optimal Algorithm

استبدال الصفحة التي لن نطلبها قريبا، أي الأبعد في السلسلة.

مثال

إذا كان لدي سلسلة طلبات الصفحات التالية:

⇨ 1,2,3,2,1,5,2,1,6,2,5,6,3,1,3,6,1,2,4,3

ما هي عدد أخطاء الصفحات (page fault) إذا استخدمنا 3 إطارات فارغة.

الحل

في البداية ستكون الإطارات الثلاثة فارغة، سنحضر الصفحات 1، 2، 3 في الإطارات الثلاث الفارغة مما يولد 3 خطأ صفحة. بعدها سنحتاج إلى الصفحات 2، 1 وهما بالذاكرة فنستخدمهما دون حدوث خطأ صفحة. بعدها سنحتاج الصفحة 5، وهي ليست بالذاكرة لذلك لابد من إخراج صفحة وإدخالها مكانها. سنخرج الصفحة التي لا نطلبها قريبا (لدينا الآن بالذاكرة الصفحات 2،1، 3) فإيها سنخرج:

- الصفحة 1 سنطلبها بعد صفحتين من الصفحة 5.
- الصفحة 2 سنطلبها بعد الصفحة 5 مباشرة.
- الصفحة 3 سنطلبها بعد 6 صفحات، وهي الأبعد لذلك سنخرجها ونضع 5 مكانها، الآن لدينا بالذاكرة الصفحات 2،1، 5.

سنحتاج الصفحة 2 وهي موجودة بالذاكرة، وبعدها سنحتاج الصفحة 1 وهي بالذاكرة، بعدها سنحتاج الصفحة 6 التي لا توجد بالذاكرة لذلك سنتبدلها بالصفحة 1 لأنها الأبعد. وهكذا إلى أن نصل للحل التالي:

1	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
1	1	1			1			6				6	6				2	2	
	2	2			2			2				2	1				1	4	
		3			5			5				3	3				3	3	

سيكون لدينا في النهاية 9 خطأ صفحة.

التي لم تستخدم حديثا (least recently used (LRU))

بإفتراض أن الصفحة التي استخدمت حديثا هي التي سنحتاجها وهي التي ستستخدم مرة ثانية، أما التي لم تستخدم حديثا فغالبا لن نحتاجها في القريب العاجل. لذلك سنخرج الصفحة التي لم تستخدم حديثا (الأقدم استخداما).

مثال

إذا كان لدي سلسلة طلبات الصفحات التالية:

⇨ 1,2,3,2,1,5,2,1,6,2,5,6,3,1,3,6,1,2,4,3

ما هي عدد أخطاء الصفحات (page fault) إذا استخدمنا 3 إطارات فارغة.

الحل

في البداية ستكون الإطارات الثلاثة فارغة، سنحضر الصفحات 1، 2، 3 في الإطارات الثلاث الفارغة مما يولد 3 خطأ صفحة. بعدها سنحتاج إلى الصفحات 2، 1 وهما بالذاكرة فنستخدمهما دون حدوث خطا صفحة. بعدها سنحتاج الصفحة 5، وهي ليست بالذاكرة لذلك لابد من إخراج صفحة وإدخالها مكانها. سنخرج الصفحة الأقل استخداما (لدينا الآن بالذاكرة الصفحات 1،2، 3)

- الصفحة 1 استخدمت آخر شئ.
- الصفحة 2 استخدمت قبل الصفحة 1.
- الصفحة 3 استخدمت قبل الصفحة 2،أي هي الأول استخداما بين الصفحات الثلاث التي بالذاكرة، لذلك سنخرجها ونضع 5 مكانها، الآن لدينا بالذاكرة الصفحات 1،2، 5.

سنحتاج الصفحة 2 وهي موجودة بالذاكرة، وبعدها سنحتاج الصفحة 1 وهي بالذاكرة، بعدها سنحتاج الصفحة 6 وهي ليست بالذاكرة لذلك سنتبدلها بالصفحة 5 لأنها الأقل استخداما. وهكذا إلى أن نصل للحل التالي:

1	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
1	1	1	1	3	2	1	5	2	1	6	2	5	6	6	1	3	6	1	2
	2	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4
		3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3

سيكون لدينا في النهاية 11 خطأ صفحة.

الخوارزميات المعتمدة على العدد (counting-based)

- خوارزمية الصفحة الأقل استخداما ((least frequently used (LFU)
- خوارزمية الصفحة الأكثر استخداما ((most frequently use (MFU)

خوارزمية الصفحة الأقل استخداما ((least frequently used (LFU)

بإفتراض أن الصفحات الاكثر استخدامها (قيمة عداداها أكبر) هي النشطة وهي التي ستستخدم كثيرا، فنقوم بإخراج الصفحة التي لها أقل رقم في عدادها، أي عدد مرات استخدامها أقل من بقية الصفحات.

لكن هذه الخوارزمية قد يكون بها مشكلة إذا كانت الصفحة ذات العداد الأكبر لن تعمل مرة أخرى، فتظل في الذاكرة دون فائدة منها. ويمكن حل مثل هذه المشكلة بنقص عدادها بفترات منتظمة.

خوارزمية الصفحة الأكثر استخداما (most frequently use (MFU))

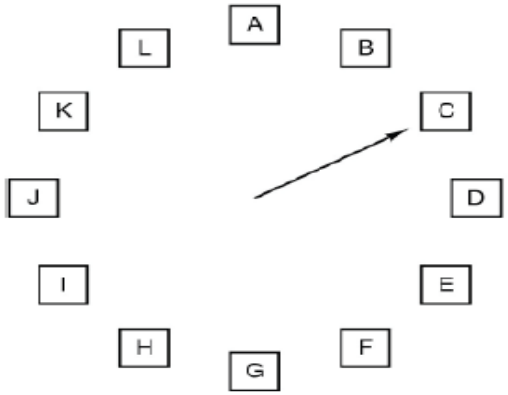
هنا نفترض أن الصفحات التي لها قيمة عداد أقل هي التي وصلت حديثا وهي التي تحتاج إلى وقت اكثر في الذاكرة، لذلك سنخرج الصفحة التي لها رقم أكبر في عدادها (التي استخدمت كثيرا).

تعتبر هذه الخوارزميات غير شائعة لأن تطبيقها مكلف.

خوارزمية الساعة (Clock)

سيتم إختبار الصفحة التي يشير إليها مؤشر الساعة هل استخدمت حديثا أم لا، فإذا استخدمت حديثا نحول مؤشر الساعة ليشير إلى الصفحة التي تليها ونختبرها هل استخدمت حديثا أم لا، فإذا استخدمت ننقل المؤشر للصفحة التي تليها، وهكذا حتى نعثر على صفحة لم تستخدم فنخرجها.

في الشكل التالي سنختبر الصفحة C فإذا استخدمت سيتحول المؤشر إلى الصفحة D ونختبرها هل استخدمت أم لا، وهكذا حتى نعثر على صفحة لم تستخدم فنستبدلها ويتحول المؤشر إلى الصفحة التي تليها.



تمرينات

1. إذا كان لدينا سلسلة الصفحات التالية (page refrence string):

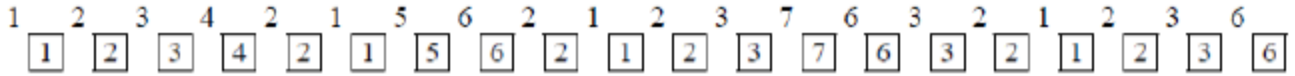
⇨ 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

أحسب عدد أخطاء الصفحات (page faults) للخوارزميات التالية بإفتراض إطار واحد، إطارين، ثلاث إطارات، أربع إطارات، خمسة إطارات، ستة إطارات، وسبعة إطارات:

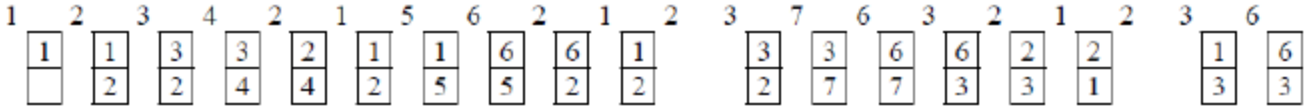
* خوارزمية FIFO
* خوارزمية LRU

الحل

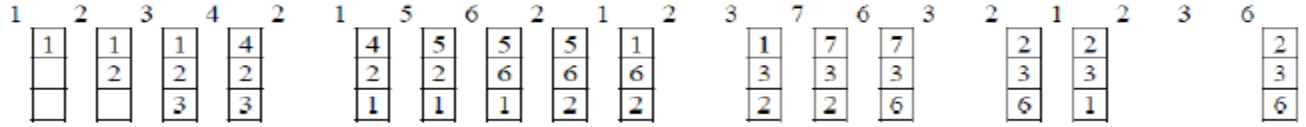
لخوارزمية LRU



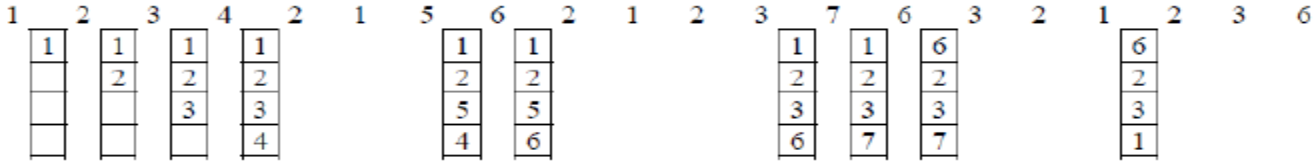
20 خطأ صفحة

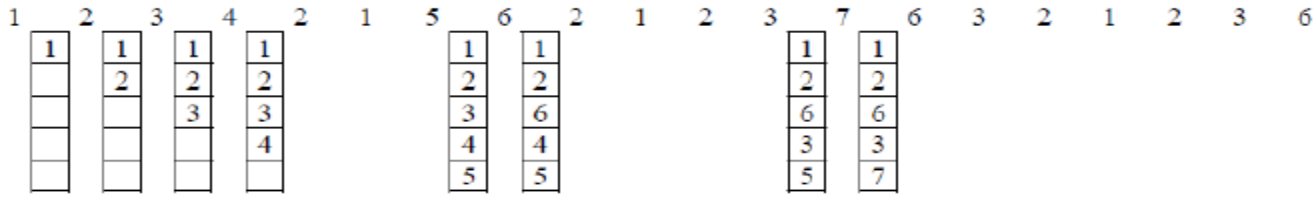


18 خطأ صفحة

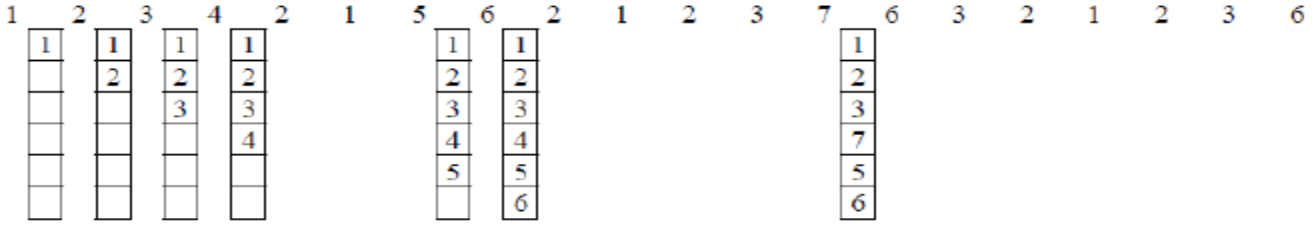


15 خطأ صفحة

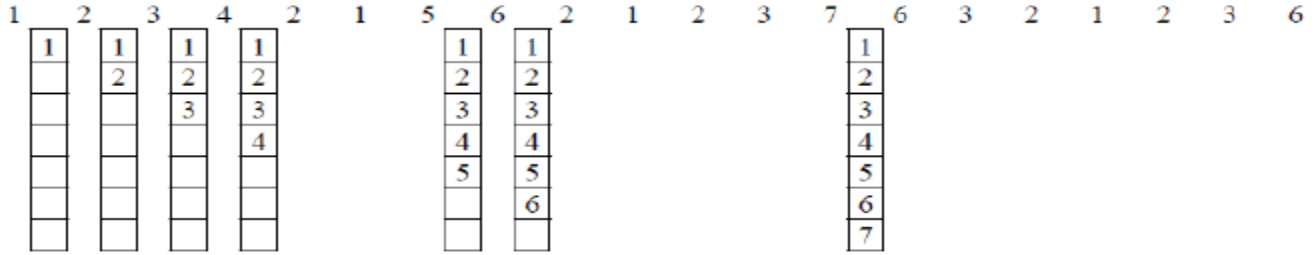




8 خطأ صفحة.



7 خطأ صفحة.



اكمل الحل بالنسبة لخوارزمية FIFO

تمارين غير محلولة

2. عرف الذاكرة الظاهرية ؟
3. تكون البرامج التي تستخدم الذاكرة الظاهرية بطيئة نوع ما، لماذا ؟
4. ما هي المبادلة (swapping) ؟
5. تنقسم المبادلة إلى نوعين ، ما هما ؟
6. اذكر ثلاث من خوارزميات تبديل الصفحات ؟
7. إذا كان لدي سلسلة طلبات الصفحات التالية:

1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

- أحسب عدد أخطاء الصفحات (page fault) إذا استخدمنا خوارزمية FIFO لعدد:
  - إطار واحد فارغ.
  - إطارين فارغين .
  - ثلاث إطارات فارغة.
  - أربع إطارات فارغة.