الاسبوع الرابع والعشرون

الذاكرة الافتراضية Virtual Memory

التصفح حسب الصفحة المطلوبة – التقييم البطيء Demand paging- Lazy evaluation

عند القيام باستخدام برنامج ما فان هذا البرنامج يتم تقسيمه لصفحات pagesوهذه الصفحات يتم تحميلها للذاكرة لكي يتم تنفيذ هذا البرنامج , في بيئة طلب الصفحات (Demand paging) لا يتم تحميل جميع صفحات البرنامج إلى الذاكرة الرئيسية , إنما فقط الصفحات المطلوبة لتنفيذ العمليات التي يقوم بها المستخدم لهذا البرنامج , وهذه التقنية تعتبر أفضل بكثير من تحميل كامل البرنامج إلى الذاكرة (جميع صفحات البرنامج) , وذلك لان أجزاء من البرنامج قد لا تستخدم خلال تشغيله , مما يعني حجز جزء من الذاكرة لصفحات من البرنامج غير مستخدمة , هذه التقنية تمكننا من تحميل عدة برامج كبيرة في آن واحد نظراً لان كل برنامج لن يتم تحميله بالكامل , مما يوفر مساحة في الذاكرة لبرامج أخرى .

عند طلب المستخدم تنفيذ صفحة لم يتم تحميلها إلى الذاكرة , سيتم تحميل هذه الصفحة المطلوبة من خلال المبدل البطيء (الكسول) (Lazy swapper) . عند تطبيق تقنية طلب الصفحات ، قد تحتاج احد البرامج التي يعمل عليها المعالج حاليا إلى صفحة لم يتم تحميلها إلى الذاكرة مما ينتج ما يطلق عليه بخطأ الصفحة Page Fault

استبدال الصفحة Page Replacement

ذكرنا سابقاً إلى حصول حالة خطأ الصفحة ، ويمكن معالجته بأن يتم إحضار هذه الصفحة من القرص الصلب إلى الذاكرة ، وربما تظهر مشكلة أخرى في النظام ألا وهي عدم وجود إطار فارغ في الذاكرة ، لذلك يطبق النظام خوارزميات على أساسها يختار صفحة (الضحية) ليستبدلها بالصفحة المطلوبة .

خوارزميات استبدال الصفحة Page Replacement Algorithms

خوارزميات استبدال الصفحات هي التي تختار وتقرر الصفحة التي تخرج من الذاكرة عندما لا يكون هناك إطاراً (Frame) فارغاً في الذاكرة ، وتوضع الصفحة الخارجة (الضحية) في القرص الصلب ، وتهدف هذه الخوارزميات إلى تقليل نسبة حدوث خطأ الصفحة (Page Fault) ، فكلما زادت عدد الإطارات بالذاكرة كلما قلت نسبة خطأ الصفحة .

يوجد العديد من خوارزميات استبدال الصفحات ، كل خوارزمية لها طريقتها الخاصة في اختيار الضحية في حال وجود خطأ الصفحة ، ويتم تقييم واختيار واحدة منها على أساس الأقل تسببا في خطأ الصفحة . يتم اختبار كل خوارزمية بتطبيقها على سلسلة من طلبات الصفحات (Reference string) ، كل صفحة تمثل برقم ، لذا فهي سلسلة من الأرقام ومن ثم نحسب عدد مرات حدوث خطأ الصفحة لكل خوارزمية .

من هذه الخوارزميات :

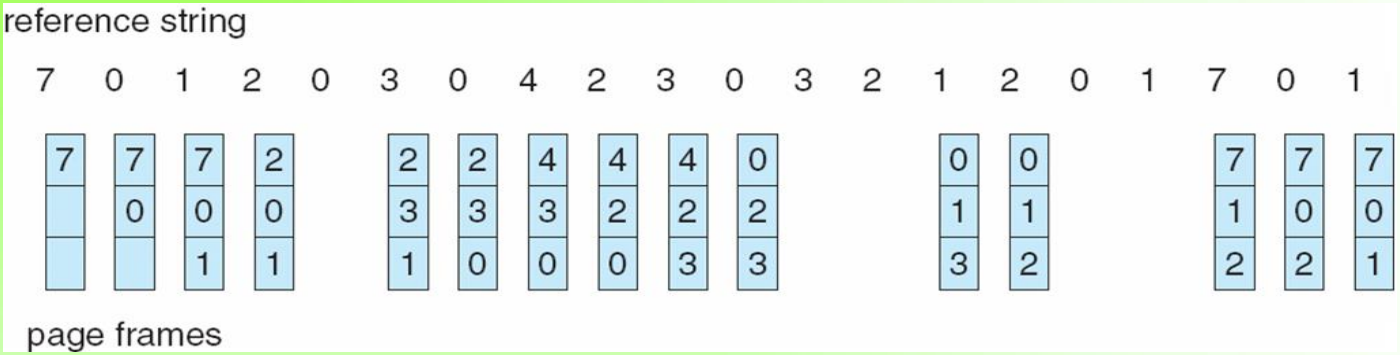
1- خوارزمية الصفحة التي تصل أولا تخرج أولا (First In First Out Algorithm) : ومختصرها (FIFO)

تعتبر هذه الخوارزمية من أبسط خوارزميات تبديل الصفحة سهلة الفهم والبرمجة ، إذ أن هذه الخوارزمية تربط كل صفحة مع الوقت الذي وصلت للذاكرة ، وعندما نختار صفحة لنستبدلها فإننا نختار أقدم صفحة وصلت ودخلت الذاكرة.

مثال:

كم خطأ صفحة يحصل لسلسلة من طلبات الصفحات (Reference string) أدناه ، علما أن عدد الإطارات الفارغة هي ثلاثة (3 Free Page Frames) .

الحل:

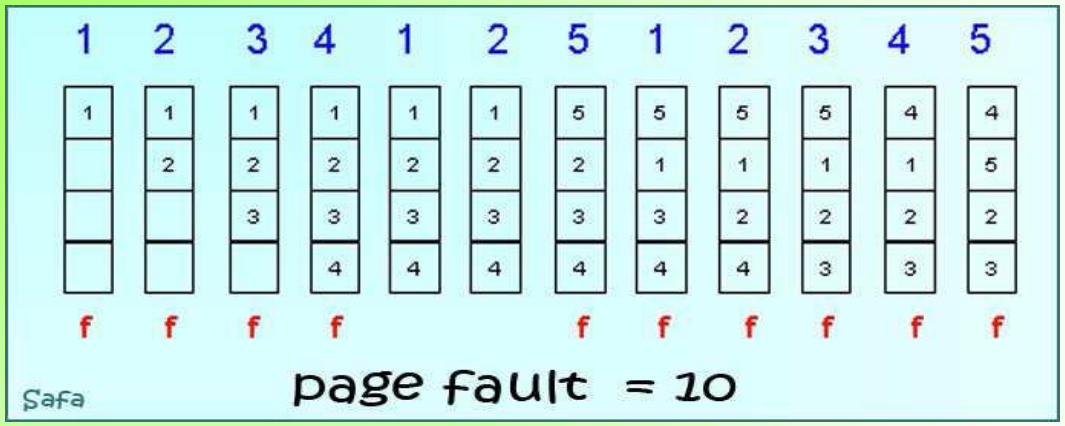


15 خطأ صفحة

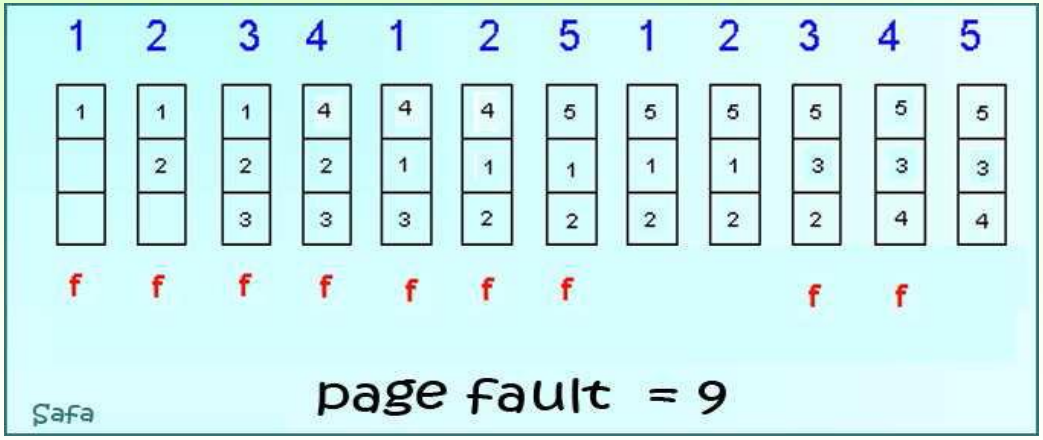
نعرف بديهياً أن زيادة عدد الإطارات الفارغة في الذاكرة المعطاة لكل برنامج ، يمكن أن يقلل من احتمال العدد الكلي لأخطاء الصفحة التي تتسبب بها تقنية استبدال الصفحات ، ولأن لكل قاعدة شذوذ ، فإن الشذوذ عن هذه القاعدة يدعى شذوذ بيلادي (Belady's Anomaly) . لا يعني بالضرورة أن العدد الكلي لأخطاء الصفحات سيقل مع زيادة في عدد الإطارات الفارغة في الذاكرة .

مثال:

المثال يوضح شذوذ بيلادي ، أولاً بتطبيق سلسلة طلبات الصفحات على أربعة إطارات فارغة ، ينتـج لنا (10) أخطاء صفحة (f) .



والآن نطبق نفس السلسلة ولكن على ثلاث إطارات فارغة بدلاً من أربعة فينتج لنا (9) أخطاء فقط .



2- خوارزمية الصفحة ذات الاستخدام الأقل (Least Recently Used Algorithm):

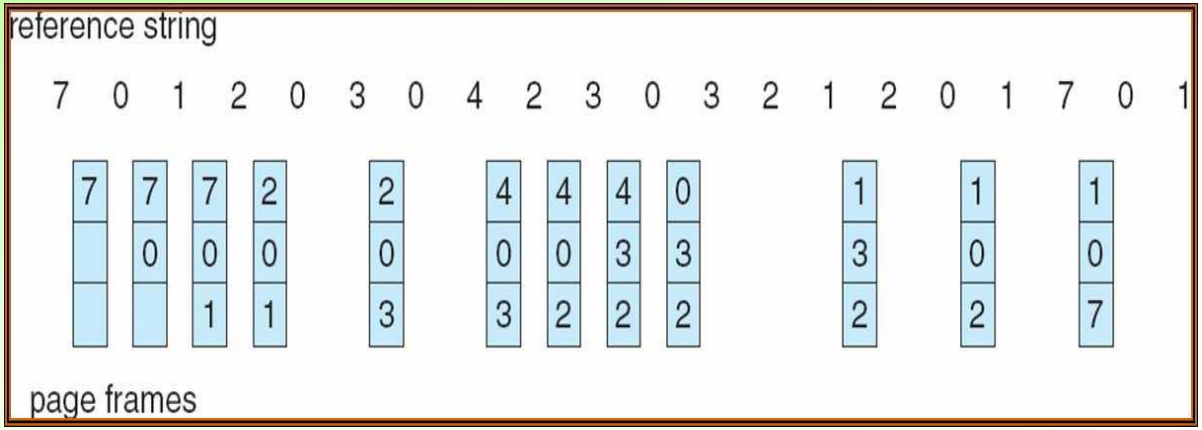
ومختصرها (LRU)

هذه الخوارزمية تستخدم الماضي القريب ، فتستبدل الصفحة التي لم يتم استخدامها لأطول مدة ، أي الأقل استخداماً.

مثال:

كم خطأ صفحة يحصل لسلسلة من طلبات الصفحات (Reference string) أدناه ، علما أن عدد الإطارات الفارغة هي ثلاثة (3 Free Page Frames):

الحل:



12 خطأ صفحة

أسئلة اختبارية :

س1: عرف خطأ الصفحة Page Fault ؟ .

س2: خوارزميات استبدال الصفحات تهدف الى 00000 نسبة حدوث خطأ الصفحة , فكلما زادت عدد 00000 بالذاكرة كلما 000000 نسبة خطأ الصفحة ؟ .

س3: ما المقصود (بشذوذ بيلادي) وفي اية خوارزمية استبدال الصفحات تحصل ؟ .