

الاسبوع الثالث والعشرون

الذاكرة الافتراضية Virtual Memory

الذاكرة الافتراضية Virtual Memory

الفكرة الأساسية وراء الذاكرة الافتراضية أن الحجم الكامل للبرنامج والبيانات قد يزيد عن كمية الذاكرة الفيزيائية المتوفرة له . يحتفظ نظام التشغيل بأجزاء البرنامج المستخدمة حالياً في الذاكرة الرئيسية ، ويترك الباقي في القرص . مثلاً، يمكن تشغيل برنامج بحجم 16MB على جهاز فيه 4MB باختيار 4MB اللازمة كي تبقى في الذاكرة عند كل لحظة ومبادلة أجزاء البرنامج بين القرص الصلب والذاكرة حسب الحاجة .

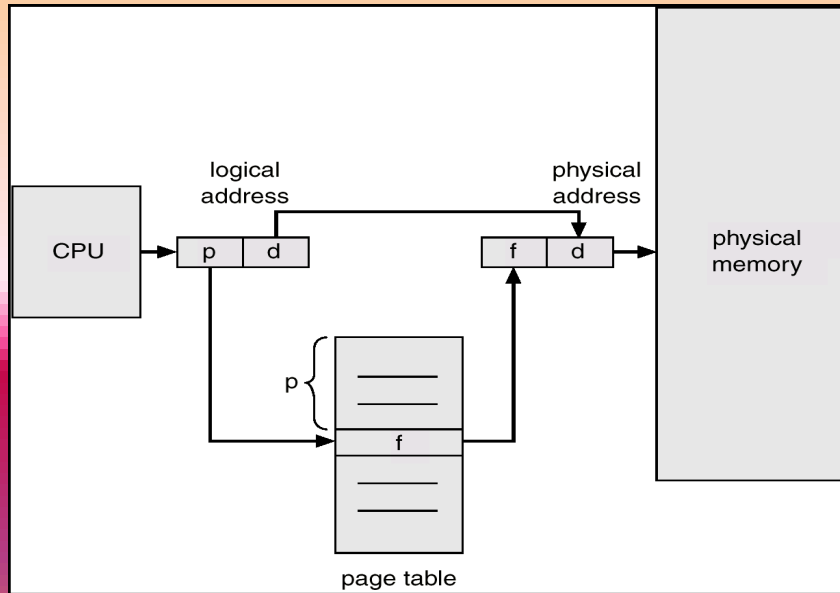
التصفح Paging

الشكل (43) يوضح أسلوب آخر لتخصيص الذاكرة ولكن من نوع غير المتجاور ، مبدأ التصفح هو أن العملية process منطقياً يقسم من قبل نظام التشغيل إلى أقسام متساوية بالحجم يطلق عليها بالصفحة (Page) . بالمقابل يتم تقسيم الذاكرة الفيزيائية إلى أقسام متساوية بالحجم أيضاً مساوية لحجم الصفحة ويطلق عليها بالإطار (Frame) . مما تقدم نستنتج أن :

حجم الصفحة = حجم الإطار

Page size = Frame size

قام المصممون بدراسة أكفاً حجم للصفحة بحيث لا يحدث فقدان أو ضياع كبير لمساحات الذاكرة عند ظهور حالة التفسخ الداخلي (Internal Fragmentation) وكان حجم كل صفحة هو (4 بايت) .



الشكل (43) هيكلية ذاكرة في بيئة الصفحات

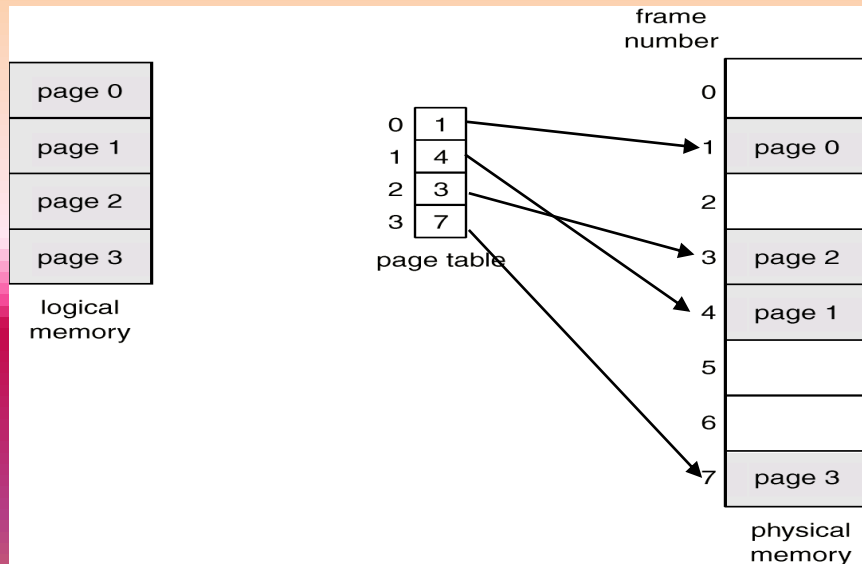
جدول خارطة الصفحات Page Map Table

رقم الصفحة	رقم الإطار
0	5
1	6
2	1
3	2

لكي يسيطر نظام التشغيل على مسألة تحميل الصفحات إلى الذاكرة الرئيسية لغرض التنفيذ يتم توفير جدول يدير هذا الإجراء يطلق عليه جدول خارطة الصفحات (Page Map Table) ومختصره (PMT) .
يوضع فيه رقم الصفحة (Page) ورقم الإطار Frame. الشكل (44)
يمثل نموذج لجدول خارطة الصفحات لعملية معينة .

الشكل (44) جدول خارطة الصفحات

مثال :



عملية (process) تتكون من (4 صفحات) ، معطاة جدول خارطة الصفحات ، كيف يتم تخصيص موقع لكل صفحة في الذاكرة بالاعتماد على الخارطة ؟.

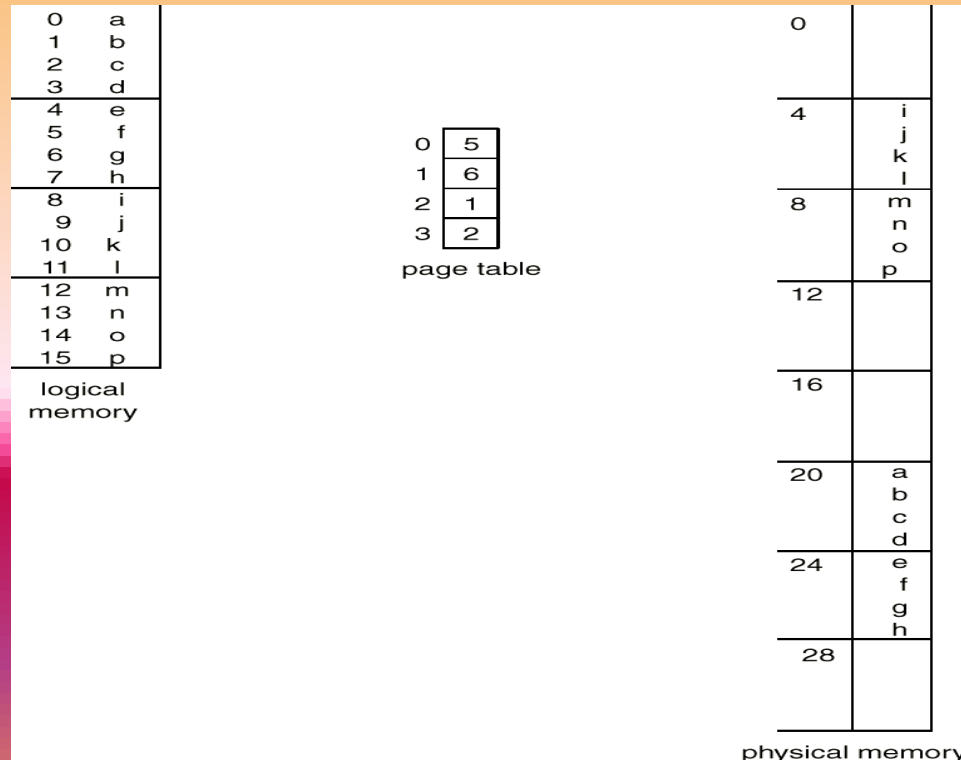
تمثيل البرنامج في الذاكرة Paging Implementation Memory

مثال:

لديك العملية (process) بحجم (16 bytes) , حجم الذاكرة الرئيسية (32 bytes) باستخدام صفحة بحجم (4 bytes) , كيف يتم تمثيل العملية التي محتوياتها (a b c d e f g h I j k l m n o p) في الذاكرة الرئيسية ؟ علماً أن جدول خارطة الصفحات كالآتي :

الحل:

0	5
1	6
2	1
3	2



لمعرفة العنوان الفيزيائي لأي حرف من محتويات العملية نستخدم القانون الآتي :
العنوان الفيزيائي = (رقم الإطار * حجم الصفحة) + الإزاحة

$$\text{Physical address} = (\text{Frame no.} * \text{Page size}) + \text{offset}$$

على افتراض نرغب في حساب العنوان الفيزيائي للحرف (n) لذا نطبق القانون أعلاه :

$$\text{Physical address} = (2 * 4) + 1 = 9$$

ملاحظة:

كلما يزيد حجم الصفحة كلما يقل حجم جدول خارطة الصفحات والعكس صحيح أيضا :
فلو كان لدينا برنامج حجمه (24 بايت) وحجم الصفحة الواحدة (4 بايت) فان الجدول يتكون من (6) مداخل :
قسمة حجم البرنامج على حجم الصفحة الواحدة (24 بايت / 4 بايت) .
أما لو كان حجم الصفحة الواحدة (8 بايت) فان الجدول يتكون من (3) مداخل :

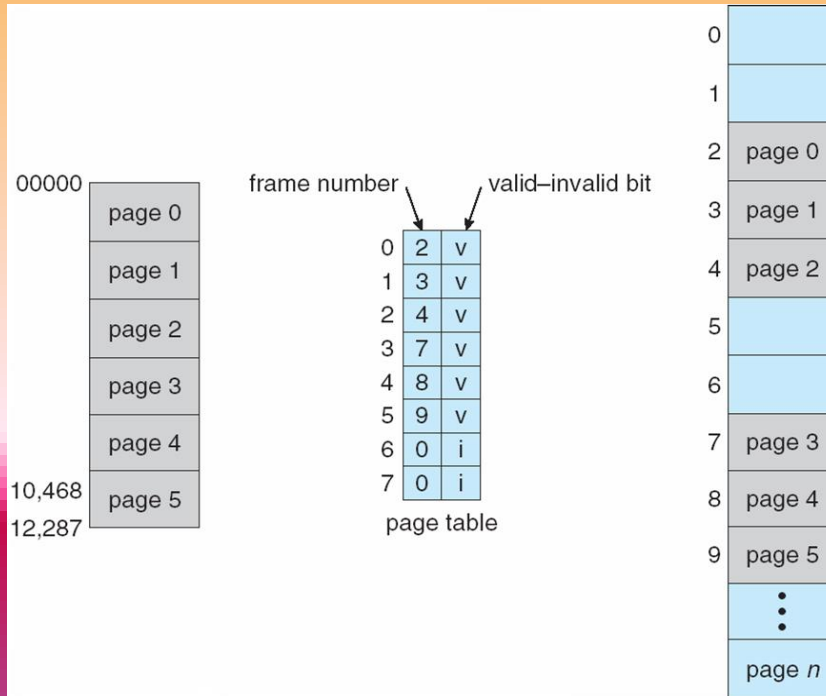
مثال:

برنامج حجمه (5000 بايت) , احسب عدد الصفحات ومقدار التفسخ الداخلي عندما :
1- حجم الصفحة = 512 بايت .
2- حجم الصفحة = 4096 بايت .

حماية الذاكرة Memory Protection

حماية الذاكرة في بيئة الصفحات تنجز من خلال (bits) توضع مع كل صفحة في جدول خارطة الصفحات كما في الشكل (45)، منها (Protection bits) ويطلق عليه (valid/invalid bit) ، الحرف (v) يشير أن الصفحة موجودة في الذاكرة الرئيسية ، أما الحرف (i) يشير إلى احتمالين :

- 1- أن الصفحة لا يحتويها البرنامج أساساً .
- 2- الصفحة موجودة لكن لم يتم تحميلها إلى الذاكرة الرئيسية بعد (لا تزال في القرص) .



كما يمكن إضافة (bits) أخرى للجدول أعلاه مثل (R) للقراءة و (W) للكتابة لمعرفة ما يمكن إجراؤه على تلك الصفحة خلال حساب العنوان الفيزيائي .

الشكل (45) حماية الذاكرة في بيئة الصفحات من خلال وضع بتات

اسئلة اختبارية :

- س1: ما المقصود بالذاكرة الافتراضية Virtual Memory ؟.
- س2: مبدأ 00000 هو أن يتم تقسيم العملية Process منطقيا الى اقسام 0000 بالحجم يطلق عليها ب 0000 بالمقابل يتم تقسيم الذاكرة الفيزيائية الى اقسام ويطلق عليها 0000000 ؟.
- س3: جدول خارطة الصفحات يوضع فيه 00000 و 0000000 ؟.
- س4: كلما 00000 حجم الصفحة كلما 0000000 حجم خارطة الصفحات والعكس صحيح ايضا ؟.