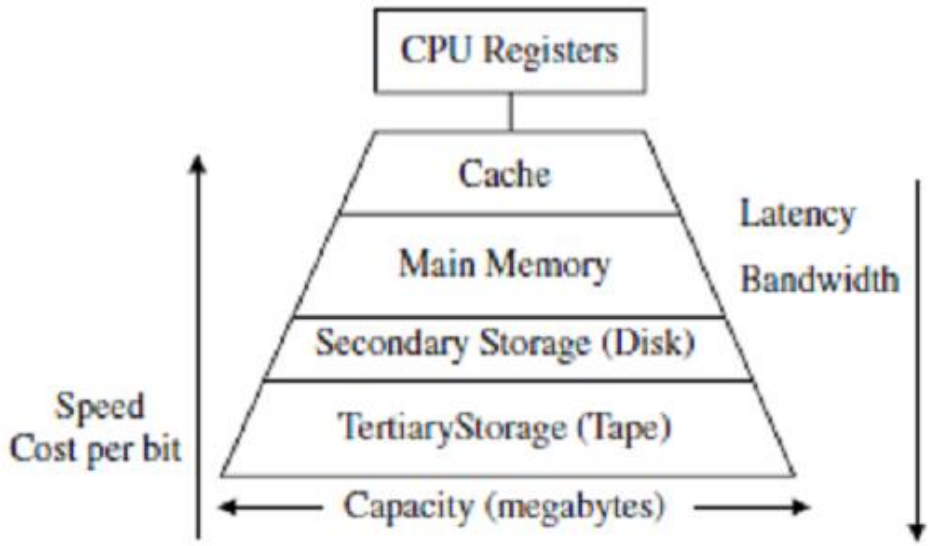
Introduction مقدمة

A computer memory has to be organized in a hierarchy. In such a hierarchy, larger and slower memories are used to supplement smaller and faster ones. if we aside register of CPU, typical memory hierarchy starts with a small, expensive, and relatively fast unit, called the cache, followed by a larger, less expensive, and relatively slow main memory unit. Cache and main memory is called primary memory that followed by larger, less expensive, and far slower magnetic memories that consist typically of the (hard) disk and the tape. The disk is called the secondary memory, while the tape is conventionally called the tertiary memory. Figure 4.1 depicts a typical memory hierarchy

يجب تنظيم ذاكرة الكمبيوتر في تسلسل هرمي. في مثل هذا التسلسل الهرمي ، تُستخدم الذكريات الأكبر والأبطأ لتكملة الذكريات الأصغر والأسرع. إذا وضعنا سجل وحدة المعالجة المركزية جانباً ، فإن التسلسل الهرمي للذاكرة النموذجي يبدأ بوحدة صغيرة ومكلفة وسريعة نسبيًا ، تسمى ذاكرة التخزين المؤقت ، تليها وحدة ذاكرة رئيسية أكبر وأقل تكلفة وبطيئة نسبيًا. تسمى ذاكرة التخزين المؤقت والذاكرة الرئيسية الذاكرة الأولية التي تليها ذاكرة مغناطيسية أكبر وأقل تكلفة وأبطأ بكثير والتي تتكون عادةً من القرص (الصلب) والشريط. يُطلق على القرص اسم الذاكرة الثانوية ، بينما يُطلق على الشريط تقليديًا اسم الذاكرة الثالثة. الشكل 4.1 يصور تسلسل هرمي نموذجي للذاكرة

Figure 4-1: memory hierarchy



The memory hierarchy can be characterized by a number of parameters. Among these parameters are:

يمكن وصف التسلسل الهرمي للذاكرة بعدد من المعلمات. من بين هذه المعلمات:

- The access type refers to the action that physically takes place during a read or write operation.
- The capacity of a memory level is usually measured in bytes.
- Cycle time is defined as the time elapsed from the start of a read operation to the start of a subsequent read .
- Latency is defined as the time interval between the request for information and the access to the first bit of that information.
- Bandwidth provides a measure of the number of bits per second that can be accessed.
- The cost of a memory level is usually specified as dollars per megabytes.

- يشير نوع الوصول إلى الإجراء الذي يحدث فعليًا أثناء عملية القراءة أو الكتابة.
- تُقاس سعة مستوى الذاكرة عادةً بالبايت.
- يتم تعريف وقت الدورة على أنه الوقت المنقضي من بداية عملية القراءة إلى بداية القراءة اللاحقة.
- يتم تعريف الكمون على أنه الفاصل الزمني بين طلب المعلومات والوصول إلى الجزء الأول من تلك المعلومات.
- يوفر النطاق الترددي مقياسًا لعدد البتات التي يمكن الوصول إليها في الثانية.
- عادة ما يتم تحديد تكلفة مستوى الذاكرة بالدولار لكل ميغا بايت.

x86 Memory Management

x86 processors manage memory according to the basic modes of operation. Protected mode is the most robust and powerful, but it does restrict application programs from directly accessing system hardware.

x86 إدارة الذاكرة

تدير معالجات x86 الذاكرة وفقًا لأنماط التشغيل الأساسية. الوضع المحمي هو الأقوى والأقوى ، لكنه يقيد برامج التطبيقات من الوصول المباشر إلى أجهزة النظام.

Real-Address Mode

In real-address mode, an x86 processor can access 1,048,576 bytes of memory (1 MByte) using 20-bit addresses in the range 0 to FFFFF hexadecimal. Intel engineers had to solve a basic problem: The 16-bit registers in the Intel 8086 processor could not hold 20-bit addresses. They came up with a scheme known as segmented memory. All of memory is divided into 64-kilobyte (64-KByte) units called segments, shown in Figure 3–2. An analogy is a large building, in which Segment represent the building's floors. A person can ride the elevator to a particular floor, get off, and begin following the room numbers to locate a room. The offset of a room can be thought of as the distance from the elevator to the room.

في وضع العنوان الحقيقي ، يمكن لمعالج x86 الوصول إلى 1،048،576 بايت من الذاكرة (1 ميغا بايت) باستخدام عناوين 20 بت في النطاق 0 إلى FFFFF الست عشري. كان على مهندسي Intel حل مشكلة أساسية: لا يمكن أن تحتوي سجلات 16 بت في معالج Intel 8086 على عناوين 20 بت. لقد توصلوا إلى مخطط يعرف باسم الذاكرة المجزأة. كل الذاكرة مقسمة إلى 64 كيلو بايت (64 كيلو بايت) وحدات تسمى المقاطع ، موضحة في الشكل 2-3. التشبيه هو مبنى كبير ، حيث يمثل الجزء أرضيات المبنى. يمكن لأي شخص ركوب المصعد إلى طابق معين والنزول منه والبدء في تتبع أرقام الغرفة لتحديد موقع الغرفة. يمكن اعتبار إزاحة الغرفة على أنها المسافة من المصعد إلى الغرفة.

Again in Figure 3–2, each segment begins at an address having a zero for its last hexadecimal digit. Because the last digit is always zero, it is omitted when representing segment values. A segment value of C000, for example, refers to the segment at address C0000. The same figure shows an expansion of the segment at 80000. To reach a byte in this segment, add a 16-bit offset (0 to FFFF) to the segment's base location. The address 8000:0250, for example, represents an offset of 250 inside the segment beginning at address 80000. The linear address is 80250h.

مرة أخرى في الشكل 2-3 ، يبدأ كل مقطع من عنوان يحتوي على صفر لأخر رقم سداسي عشري. نظرًا لأن الرقم الأخير يكون دائمًا صفرًا ، يتم حذفه عند تمثيل قيم المقطع. تشير قيمة القطعة C000 ، على سبيل المثال ، إلى المقطع الموجود في العنوان C0000. يوضح نفس الشكل توسيع المقطع عند 80000. للوصول إلى بايت في هذا المقطع ، أضف إزاحة 16 بت (0 إلى FFFF) إلى الموقع الأساسي للمقطع. العنوان 8000: 0250 ، على سبيل المثال ، يمثل إزاحة 250 داخل المقطع الذي يبدأ من العنوان 80000. العنوان الخطي هو h80250.

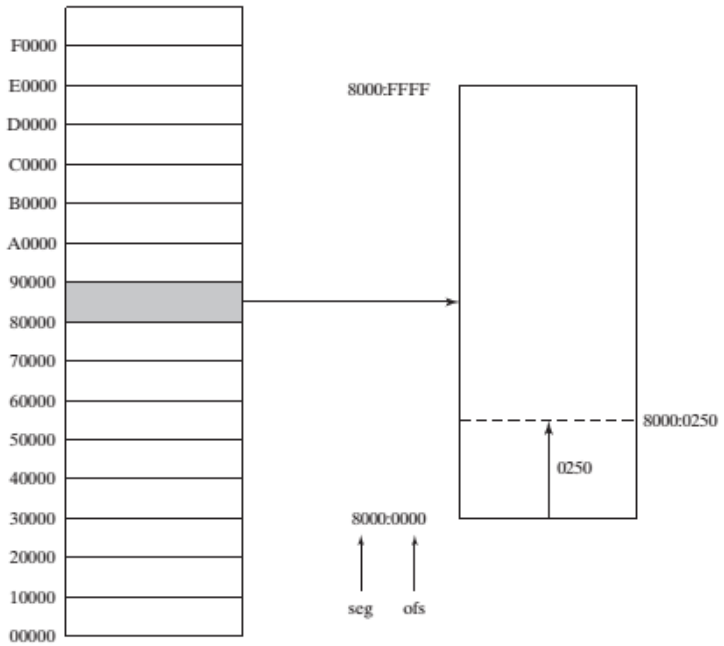


Figure 3–2: Segmented Memory Map, Real-Address Mode

20-Bit Linear Address Calculation

An address refers to a single location in memory, and x86 processors permit each byte location to have a separate address. The term for this *is byte addressable memory*. In real-address mode, the linear (or absolute) address is 20 bits, ranging from 0 to FFFFF hexadecimal. Programs cannot use linear addresses directly, so addresses are expressed using two 16-bit integers. A ***segment-offset address*** includes the following:

20 بت حساب العنوان الخطي

يشير العنوان إلى موقع واحد في الذاكرة ، وتسمح معالجات x86 أن يكون لكل موقع بايت عنوان منفصل. مصطلح هذا هو بايت الذاكرة عنونة. في وضع العنوان الحقيقي ، يكون العنوان الخطي (أو المطلق) هو 20 بت ، تتراوح من 0 إلى FFFFF الست عشري. لا يمكن للبرامج استخدام العناوين الخطية مباشرة ، لذلك يتم التعبير عن العناوين باستخدام عددين صحيحين 16 بت. يتضمن عنوان إزاحة المقطع ما يلي:

•A 16-bit segment value, placed in one of the segment registers (CS, DS, ES, SS)

•A 16-bit offset value

قيمة مقطع 16 بت ، موضوعة في أحد مسجلات المقاطع (CS، DS، ES، SS)

• قيمة إزاحة 16 بت

The CPU automatically converts a segment-offset address to a 20-bit linear address. Suppose a variable's hexadecimal segment-offset address is 08F1:0100. The CPU multiplies the segment value by 16 (10 hexadecimal) and adds the product to the variable's offset:

08F1h x 10h = 08F10h (adjusted segment value)
Adjusted Segment value: 0 8 F 1 0
Add the offset: 0 1 0 0
Linear address: 0 9 0 1 0

A typical program has three segments: code, data, and stack. Three segment registers, CS, DS, and SS, contain the segments' base locations:

•CS contains the 16-bit code segment address

•DS contains the 16-bit data segment address

•SS contains the 16-bit stack segment address

•ES, FS, and GS can point to alternate data segments, that is, segments that supplement the default data segment

Protected Mode وضع حماية

Protected mode is the more powerful "native" processor mode. When running in protected mode, a program's linear address space is 4 GBytes, using addresses 0 to FFFFFFFF hexadecimal.

الوضع المحمي هو وضع المعالج "الأصلي" الأكثر قوة. عند التشغيل في الوضع المحمي ، تبلغ مساحة العنوان الخطي للبرنامج 4 غيغابايت ، باستخدام العناوين 0 إلى FFFFFFFF الست عشري.

In the context of the Microsoft Assembler, the ***flat segmentation model*** is appropriate for protected mode programming. The flat model is easy to use

because it requires only a single 32-bit integer to hold the address of an instruction or variable. The CPU performs address calculation and translation in the background, all of which are transparent to application programmers. Segment registers (CS, DS, SS, ES, FS, GS) point to ***segment descriptor tables***, which the operating system uses to keep track of locations of individual program segments.

A typical protected-mode program has three segments: code, data, and stack, using the CS, DS, and SS segment registers:

في سياق مجمع Microsoft ، يكون نموذج التجزئة المسطح مناسبًا لبرمجة الوضع المحمي. النموذج المسطح سهل الاستخدام لأنه لا يتطلب سوى عدد صحيح واحد 32 بت للاحتفاظ بعنوان تعليمة أو متغير. تقوم وحدة المعالجة المركزية بحساب العنوان والترجمة في الخلفية ، وكلها شفافة لمبرمجي التطبيقات. تشير سجلات القطاعات (CS، DS، SS، ES، FS، GS) إلى جداول واصفات القطاعات ، والتي يستخدمها نظام التشغيل لتتبع مواقع مقاطع البرنامج الفردية.

يحتوي برنامج الوضع المحمي النموذجي على ثلاثة أقسام: التعليمات البرمجية والبيانات والمكدس ، باستخدام سجلات مقاطع CS و DS و SS:

- CS references the descriptor table for the code segment
- DS references the descriptor table for the data segment
- SS references the descriptor table for the stack segment

- تشير CS إلى جدول التوصيف لمقطع الكود
- يشير DS إلى جدول التوصيف لقطاع البيانات
- تشير SS إلى جدول التوصيف لمقطع المكدس

Flat Segmentation Model نموذج التقسيم المسطح

In the flat segmentation model, all segments are mapped to the entire 32-bit physical address space of the computer. At least two segments are required, one for code and one for data. Each segment is defined by a segment descriptor, a 64-bit integer stored in a table known as the ***global descriptor table (GDT)***.

Figure 3–3 shows a segment descriptor whose ***base address field*** points to the first available location in memory (00000000). In this figure, ***the segment limit*** is 0040. The ***access field*** contains bits that determine how the segment can be used. All modern operating systems based on x86 architecture use the flat segmentation model.

في نموذج التجزئة المسطح ، يتم تعيين جميع المقاطع إلى مساحة العنوان الفعلية الكاملة للكمبيوتر 32 بت. مطلوب جزأين على الأقل ، أحدهما للتعليمات البرمجية والآخر للبيانات. يتم تعريف كل مقطع بواسطة واصف مقطع ، وهو عدد صحيح 64 بت مخزن في جدول يُعرف باسم جدول الواصف العام (GDT).

يوضح الشكل 3-3 واصف مقطع يشير حقل العنوان الأساسي الخاص به إلى أول موقع متوفر في الذاكرة (00000000). في هذا الشكل ، حد المقطع هو 0040. يحتوي حقل الوصول على بتات تحدد كيفية استخدام المقطع. تستخدم جميع أنظمة التشغيل الحديثة القائمة على معمارية x86 نموذج التجزئة المسطح.

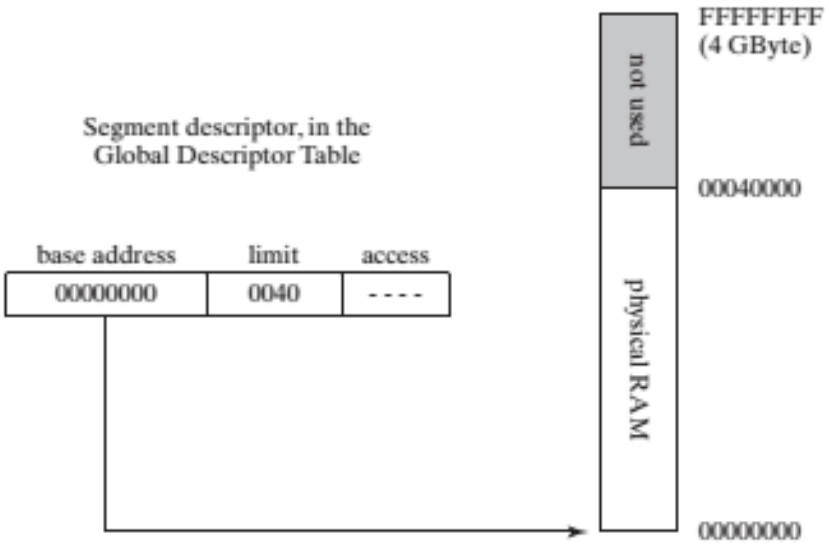


Figure 3–3: Flat Segmentation Model

Multi-Segment Model نموذج متعدد القطاعات

In the multi-segment model, each task or program is given its own table of segment descriptors, called a **local descriptor table (LDT)**. Each descriptor points to a segment, which can be distinct from all segments used by other processes. Each segment has its own address space. In Figure 3-4, each entry in the LDT

points to a different segment in memory. Each segment descriptor specifies the exact size of its segment. For example, the segment beginning at 3000 has size 2000 hexadecimal, which is computed as (0002 x 1000 hexadecimal). The segment beginning at 8000 has size A000 hexadecimal.

في النموذج متعدد الأجزاء ، يتم إعطاء كل مهمة أو برنامج جدول خاص به من واصفات المقطع ، يسمى جدول واصف محلي (LDT). يشير كل واصف إلى مقطع يمكن تمييزه عن جميع المقاطع التي تستخدمها العمليات الأخرى. كل جزء له مساحة العنوان الخاصة به. في الشكل 3-4 ، يشير كل إدخال في اختبار LDT إلى مقطع مختلف في الذاكرة. يحدد واصف كل مقطع الحجم الدقيق لمقطعه. على سبيل المثال ، المقطع الذي يبدأ عند 3000 له حجم 2000 سداسي عشري ، والذي يتم حسابه على أنه (0002 × 1000 سداسي عشري). المقطع الذي يبدأ من 8000 له حجم A000 سداسي عشري.

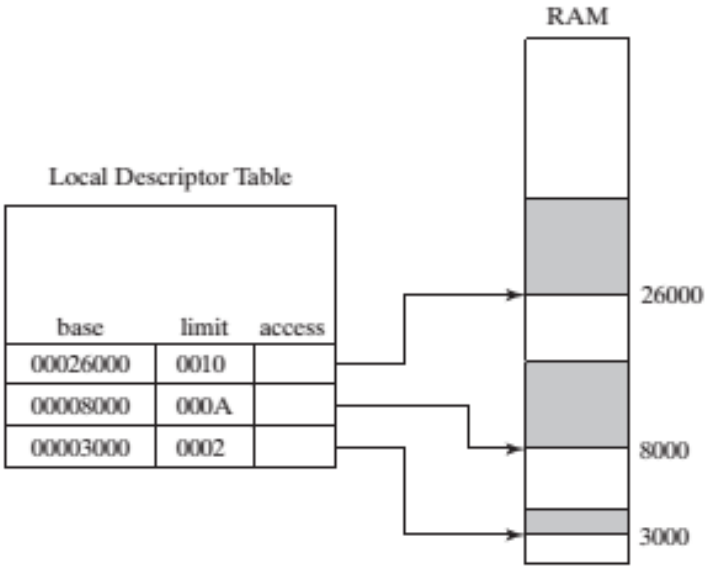


Figure 3-4: Multi-Segment Model.

Paging

x86 processors support paging, a feature that permits segments to be divided into 4,096-byte blocks of memory called pages. Paging permits the total memory used by all programs running at the same time to be much larger than the computer's physical memory. The complete collection of pages mapped by the operating system is called virtual memory. Operating systems have utility programs named virtual memory managers.

النداء

تدعم معالجات x86 الترحيل ، وهي ميزة تسمح بتقسيم المقاطع إلى 4096 بايت من كتل الذاكرة تسمى الصفحات. يسمح الترحيل بإجمالي الذاكرة المستخدمة بواسطة جميع البرامج التي تعمل في نفس الوقت لتكون أكبر بكثير من الذاكرة الفعلية للكمبيوتر. تسمى المجموعة الكاملة للصفحات التي تم تعيينها بواسطة نظام التشغيل الذاكرة الظاهرية. تحتوي أنظمة التشغيل على برامج مساعدة تسمى مديري الذاكرة الظاهرية.

Paging is an important solution to a vexing problem faced by software and hardware designers. A program must be loaded into main memory before it can run, but memory is expensive. Users want to be able to load numerous programs into memory and switch among them at will. Disk storage, on the other hand, is cheap and plentiful. Paging provides the illusion that memory is almost unlimited in size. Disk access is much slower than main memory access, so the more a program relies on paging, the slower it runs.

يعد الترحيل حلاً هامًا لمشكلة مزعجة يواجهها مصممو البرامج والأجهزة. يجب تحميل البرنامج في الذاكرة الرئيسية قبل تشغيله ، لكن الذاكرة باهظة الثمن. يريد المستخدمون أن يكونوا قادرين على تحميل العديد من البرامج في الذاكرة والتبديل بينها حسب الرغبة. التخزين على القرص ، من ناحية أخرى ، رخيص ووفير. يوفر الاستدعاء الوهم بأن الذاكرة تكاد تكون غير محدودة في الحجم. يعد الوصول إلى القرص أبطأ بكثير من الوصول إلى الذاكرة الرئيسية ، لذلك كلما زاد اعتماد البرنامج على الترحيل ، كان تشغيله أبطأ.

When a task is running, parts of it can be stored on disk if they are not currently in use. Parts of the task are paged (swapped) to disk. Other actively executing pages remain in memory. When the processor begins to execute code that has been paged out of memory it issues a page fault, causing the page or pages containing the required code or data to be loaded back into memory. To see how this works, find a computer with somewhat limited memory and run many large applications at the same time. You should notice a delay when switching from one program to another because the operating system must transfer paged portions of each program into memory from disk. A computer runs faster when more memory is

installed because large application files and programs can be kept entirely in memory, reducing the amount of paging.

عند تشغيل مهمة ما ، يمكن تخزين أجزاء منها على القرص إذا لم تكن قيد الاستخدام حاليًا. يتم ترحيل أجزاء من المهمة (تبديلها) إلى قرص. تظل الصفحات الأخرى التي يتم تنفيذها بنشاط في الذاكرة. عندما يبدأ المعالج في تنفيذ التعليمات البرمجية التي تم ترحيلها من الذاكرة ، فإنه يصدر خطأ في الصفحة ، مما يتسبب في إعادة تحميل الصفحة أو الصفحات التي تحتوي على الكود أو البيانات المطلوبة إلى الذاكرة. لمعرفة كيفية عمل ذلك ، ابحث عن جهاز كمبيوتر ذا ذاكرة محدودة نوعًا ما وقم بتشغيل العديد من التطبيقات الكبيرة في نفس الوقت. يجب أن تلاحظ تأخيرًا عند التبديل من برنامج إلى آخر لأن نظام التشغيل يجب أن يقوم بنقل الأجزاء المقسمة إلى صفحات من كل برنامج إلى الذاكرة من القرص. يعمل الكمبيوتر بشكل أسرع عند تثبيت المزيد من الذاكرة لأنه يمكن الاحتفاظ بملفات التطبيقات الكبيرة والبرامج بالكامل في الذاكرة ، مما يقلل من مقدار الترحيل.