

تحميل النواة من القرص المرن

هذا الدرس أردته استمرارا للدرس الذي وضعه الأخ IDOIT و الذي يوضح كيفية كتابة قطاع الانطلاق. و الهدف منه هو المرور إلى المرحلة الطبيعية الموالية و هي تحميل النواة من القرص المرن.

المطلوب قراءة الدرس الأول المشار إليه أعلاه حتى تتمكن من فهم ما سوف يأتي.

إذا تتبعنا الدرس السابق فقد وصلنا إلى مرحلة نسخ جملة على الشاشة النصية بعد تحميل قطاع الانطلاق، وهذا هو البرنامج

[BITS 16]

EntryPoint:

mov ax,07c0h

mov ds,ax

mov es,ax

mov ax,09000h

mov ss,ax

mov sp,0ffffh

mov si,Salam

call ShowMsg

k_loop:

jmp k_loop

ShowMsg:

push ax

push bx

.loop_start:

lodsb

cmp al,0

je .loop_end

mov ah,0Eh

mov bx,07h

int 10h

jmp .loop_start

.loop_end:

pop bx

pop ax

ret

Salam db "Salam Alikom !",13,10,0

times 510-(\$-\$\$) db 0

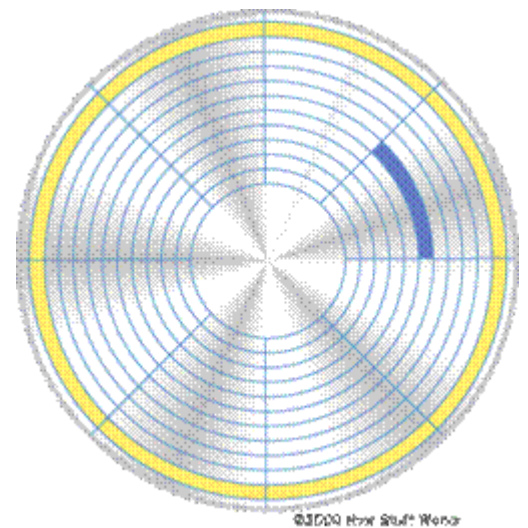
DW 0AA55h

الآن سوف نقوم بما يلي

- 1- استعمال الانقطاع 0x13 من أجل تحميل النواة .
- 2- كتابة برنامج النواة : في هذه المرحلة سوف نكتفي بعرض جملة أخرى على الشاشة، وفي مراحل قادمة إن شاء الله سنقوم بأشياء أكثر جدية

أولا - شرح موجز لبنية وطريقة عمل الأقراص :

داخل الغلاف البلاستيكي يوجد قرص مغلف بشريط مغناطيسي (مثل أشرطة الفيديو)، عندما يراد تخزين المعلومات داخل القرص يقوم قارئ الأقراص بتشفير المعلومات على شكل رموز مغناطيسية. بالطريقة نفسها يقوم القارئ عند القراءة بفك هذه الرموز و ترجمتها إلى رموز ثنائية (Binary).



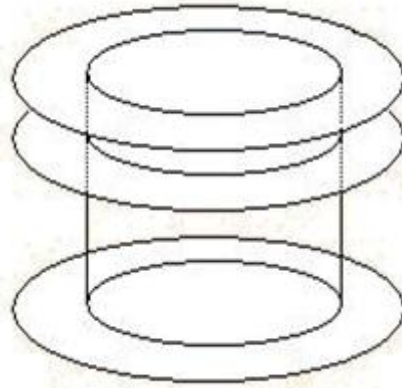
الشكل 1-1: الأقراص في القرص الصلب متراسة على شكل حزمة (stack)

يحتوي القارئ على: - محرك يقوم بجعل القرص يدور بسرعة معينة. - رأسين من أجل القراءة و الكتابة، كل رأس في جهة (هذا لأن القرص المرن يحتوي على جهتين). - كل من الرأسين متصل بذراع آلي يقوم بتحريك الرأس من وسط القرص إلى جانبه أو العكس عبر شعاع الحلقات. تتقسم مساحة القرص إلى حلقات دائرية بدءا بحواف القرص و انتهاء بوسطه (أنظر الشكل 1-1)، كل حلقة مقسمة إلى عدة قطاعات. و كل قطاع يحتوي على نفس الكمية من المعلومات، في أغلب الأحيان يكون طول القطاع 512 بايت. الأقراص الصلبة لها نفس التصميم، الفرق الوحيد هو أنه بدل قرص بلاستيكي يوجد قرص من مادة صلبة كالزجاج مما يتيح لها الدوران بسرعة أكبر بكثير من الأقراص المرنة. ذلك ما يجعل القراءة و الكتابة في الأقراص الصلبة أسرع بكثير. أيضا بدل الوجه المزدوج للقرص المرن توجد به عدة أقراص مرتبة فوق بعضها البعض.



الشكل 2-1: الأقراص في القرص الصلب متراسة على شكل كومة (stack)

بما أن للقرص المرن وجهان (و للقرص الصلب عدة أوجه)، تشكل الحلقات التي توجد في نفس الترتيب على مختلف الأوجه ما يشبه الموشور (cylinder).



الشكل 3-1: الحلقات الموجودة على نفس الترتيب تشكل موشورا

من وجهة نظر المبرمج، من أجل قراءة أو كتابة قطاع معين ينبغي تحديد : 1- رقم الموشور، 2- رقم الجهة (أو الرأس) داخل الموشور، 3- رقم القطاع داخل الحلقة.
يتم تخزين البيانات وفقا للترتيب التالي، أولا القطاع الأول على الجهة الأولى في الموشور الموجود على أقصى الحافة، ثم القطاع الثاني إلى أن تستنفذ كل قطاعات الجهة، ثم يتم المرور إلى الجهة الثانية في نفس الموشور وهكذا دواليك، بعد ذلك يتم المرور إلى الموشور الثاني فالثالث إلى غاية آخر موشور في الوسط.

يجب التنبيه هنا على أن أرقام القطاعات تبدأ من 1 بينما أرقام الموشورات والجهات تبدأ من الصفر.

من أجل قراءة القطاعات بالبيوس نستعمل الوظيفة رقم 2 في الإنقطاع 0x13. فيما يلي شرح لكيفية استعمال هذه الوظيفة لقراءة القرص المرن

استعمال الوظيفة 2: نقوم بقراءة القطاعات من القرص المرن أو الصلب	
المدخل (input)	
es:bx - الموقع الذي سيحمل إليه القطاع (أو القطاعات) في الذاكرة	
es : عنوان القسم (segment adress)	
bx : العنوان داخل القسم (Offset)	
ah - رقم الوظيفة (2)	
dh - رقم الوجه (القرص المرن له وجهان الوجه الأول يبدأ ب 0)	
dl - رقم القارئ (0 : القرص المرن A)	
ch - رقم الموشور	

cl - رقم القطاع (القطاعات تبدأ من 1)
al - عدد القطاعات المطلوب قراءتها

المخرج (output)

CF (carry flag) = 0 : نجحت العملية، و تم تحميل البيانات إلى العنوان المطلوب
 CF (carry flag) = 1 : هناك خطأ، السجل ah يحتوي على رمز الخطأ

في هذه المرحلة سيكون لدينا برنامجان، الأول هو قطاع الانطلاق و طوله 512 بايت يتم تخزينه على القطاع الأول، يأتي بعده برنامج النواة على القطاع الثاني.

بعد هذه المقدمة يمكننا المرور إلى كتابة البرنامج. فيما يلي الإضافات التي سندخلها على قطاع الانطلاق

dl يحتوي على رقم القرص الذي تم الانطلاق منه، نقوم بتخزينه في المتغير bootdrv	mov [bootdrv], dl
استعمال الوظيفة 0 : الهدف هو تهيئة القرص من أجل القراءة الإنقطاع 0x13 هو المدخل إلى الوظائف المتعلقة بالأقراص	xor ax, ax int 0x13
استعمال الوظيفة 2: تقوم بقراءة القطاعات من القرص المرن أو الصلب es:bx - الموقع الذي سيحمل إليه القطاع (أو القطاعات) في الذاكرة في هذه الحالة : 0x100:0 (= 0x1000) ah - رقم الوظيفة (2) dh - رقم الوجه (القرص المرن له وجهان الوجه الأول يبدأ ب 0) dl - رقم القارئ (0: القرص المرن A) ch - رقم الموشور cl - رقم القطاع al - عدد القطاعات المطلوب قراءتها في حالة وقوع خطأ (CF=1-> jc = true) نقفز إلى readFail حيث نقوم بكتابة الجملة "Error, unable to read from floppy"	mov ax,0x100 mov es,ax mov bx,0 mov ah,2 mov al,1 mov ch,0 mov cl,2 mov dh,0 mov dl,[bootdrv] int 0x13 jc readFail ret readFail: mov si, readErrorMsg call ShowMsg ret

بعد هذه التعليمات نكون قد حملنا النواة إلى العنوان 0x100:0 (أي العنوان [0x100*4+0]=0x1000 بالصيغة الخطية). للبدء بتنفيذ النواة نقوم بالقفز إلى العنوان المذكور.

فيما يلي النص الكامل، التغييرات تم إدخالها بخطوط بارزة

[BITS 16]

EntryPoint:

mov [bootdrv], dl

mov ax,07c0h

mov ds,ax

mov es,ax

mov ax,09000h

mov ss,ax

mov sp,0ffffh

mov si,Salam

call ShowMsg

xor ax, ax

int 0x13

mov ax, 0x100

mov es,ax

mov bx,0

mov ah,2

mov al,1

mov ch,0

mov cl,2

mov dh,0

mov dl, [bootdrv]

int 0x13

jc readFail

jmp dword 0x100:0

readFail:

mov si, readError

call ShowMsg

k_loop:

jmp k_loop

ShowMsg:

push ax

push bx

.loop_start:

lodsb

cmp al,0

je .loop_end

mov ah,0Eh

mov bx,07h

int 10h

jmp .loop_start

.loop_end:

```

pop    bx
pop    ax
ret

```

bootdrv db 0

Salam db "Salam Alikom !",13,10,0

readError db "Error, unable to read from floppy",13,10,0

times 510-(\$-\$\$) db 0

DW 0AA55h

2- كتابة برنامج النواة

هذا البرنامج بسيط و يقوم فقط بكتابة جملة is loadedkernel على الشاشة.

أنقل هذه التعليمات في ملف kernel.s ثم احفظ الملف في نفس المجلد الذي يحتضن ملف قطاع الانطلاق.

<p>التعليمات مشفرة على 16 بايت يتم إخبار Nasm بأن السجل cs يحتوي على العنوان 0</p>	<p>[BITS 16] ORG 0</p>
<p>1- نقوم بملئ المسجلات & esds بالعنوان الذي يتم التحميل إليه 2- لكتابة الجملة على الشاشة نستعين بنفس الكود الذي استعمل في كتابة قطاع الانطلاق. يمكن نسخ الكود و إلصاقه تحت call ShowMsg</p>	<pre> mov ax,0x100 mov ds,ax mov es,ax mov ax,0x9000 mov ss,ax mov sp, 0xffff mov si,knlMsg call ShowMsg knlMasg db "Kernel loaded",13,10,0 </pre>

ملحوظة: بما أننا نستعمل ShowMsg أكثر من مرة يمكننا وضعها في ملف مستقل و من ثم إدخالها في الملفين السابقين عبر الأمر %include

للترجمة نستعمل نفس التعليمية

Nasm boot.bin o boot.s bin -f

Nasm -f bin kernel.s kernel.bin

لنقل الملفات إلى القرص المرن

```
cat boot.bin kernel.bin | dd of=/dev/fd0 bs=512
```

بالنسبة لمستعملي ويندوز

```
boot.imgcopy /b boot.bin+kernel.bin
```

لنقل الملف boot إلى القرص المرن يمكنك الاستعانة بالبرنامج ntrawrite المرفق مع الدرس

```
-f boot.imgntrawrite
```

عند السؤال type the diskette driveplease عليك الإجابة ب A

إذا أردت كتابة الملف على صورة للقرص من أجل استعمالها مع برنامج لمحاكاة الحاسب مثل Boschs

```
cat boot.bin kernel.bin | dd of=image_file bs=512 conv=notrunc
```

مع استبدال mage_file باسم الملف الفعلي.

3- النتيجة:

إذا تتبعنا التعليمات أعلاه، لديك الآن برنامج لقطاع الانطلاق يستطيع تحميل النواة من القرص المرن. ولديك أيضا نواة، صحيح لا تقوم بالشيء الكثير لكنها البداية فقط و يمكننا تطويره فيما بعد.

في الدرس القادم إن شاء الله سنقوم بدراسة أوضاع المعالج x86 وتوضيح كيفية المرور إلى الوضع المحمي الذي تعمل عليه كل أنظمة التشغيل الحديثة.