

# الفصل الرابع

## معاينة الذاكرة

تضمن الفصل أداتين لمعاينة الذاكرة هما خدمات أمر التصحيح debug ، وأمر استعراض معلومات الذاكرة mem ، وتناول برنامج التصحيح debug لفحص وتغيير الذاكرة وانشاء برامج صغيرة وتحميل أجزاء من القرص إلي الذاكرة وحفظ الذاكرة إلي القرص وتنفيذ مهمات أخرى مختلفة ، كما تناول أمر استعراض الذاكرة mem لعرض بيانات الذاكرة التقليدية الموسعة والممتدة في الحاسب الشخصي والكميات المتوفرة منها بخياراته المختلفة .



## معاينة الذاكرة

يحتوي نظام تشغيل القرص علي برنامج التصحيح (اكتشاف وتصحيح الأخطاء) Debug ، وأمر استعراض الذاكرة Mem ، وهما من المنافع التي يتيح استخدامها عددا من العمليات التي تساهم إلي حد كبير في فهم وفحص ذاكرة الحاسب الشخصي .

### برنامج DEBUG

يعد هذا البرنامج واحدا من البرامج القيمة الذي يملك أدوات اعداد البرامج بلغة الماكينة ، والسماح بالتجول خلال ذاكرة الحاسب ومعالجتها بعرض محتوياتها وتغيير ماتحتويه ، وكتابة برامج صغيرة فيها ، وتخزين أجزاء من القرص اليها ثم تغيير هذه الأجزاء واعادة حفظها مرة أخرى ، ومن بين تطبيقات المنافع الأخرى التي تتيح مثل هذا الأداة برامج منافع نورتون وأدوات الحاسب الشخصي .

يتضمن برنامج التصحيح مجموعة من الأوامر ، وكل هذه الأوامر يتم اجهاضها باستخدام مفتاحي ctrl + c ، اذ عند الضغط علي هذين المفتاحين معا يتوقف التنفيذ ، كما أن المفتاحين ctrl + s يتسبب عنهما عند الضغط عليهما معا أن يتوقف جريان عرض الشاشة مؤقتا حتي يتم الضغط علي أى مفتاح آخر .

يمكن بداية تشغيل برنامج debug بواحدة من طريقتين :

١- الطريقة الاولي عن طريق كتابة أمر تنفيذ البرنامج بكتابة اسم البرنامج شاملا المسار من مشيرة نظام تشغيل القرص ثم الضغط علي مفتاح الادخال :

```
c:>c:\dos\debug
```

فتظهر مشيرة البرنامج وهي عبارة عن الشرطة الطويلة (الواصلة hyphen) التي تبين أن البرنامج جاهز للعمل .

٢- الطريقة الثانية لتنفيذ البرنامج تتم عن طريق كتابة كل الاوامر المطلوب تنفيذها

من البرنامج مرة واحدة علي الصورة :

c:>c:\dos\debug [filename (arglist)]

عندئذ يقوم البرنامج بالعمل وتحميل الملف المطلوب filename في الذاكرة بالمعاملات arglist التي تم وضعها في أمر تشغيل البرنامج .

أوامر برنامج debug بعد تشغيله تكتب علي صورة حرف واحد ، وهو الحرف الأول من الكلمة الدالة علي الأمر ، يلي هذا الحرف واحد أو أكثر من المعاملات ، وإذا حدث خطأ في كتابة الأمر فسوف تظهر رسالة تبين وجود خطأ علي الصورة: Error  
وقد يتبعها دليل يبين موطن الخطأ علي صورة رسالة خطأ .

المعاملات التي تلي الأوامر يمكن كتابتها في جميع الأوامر (ماعدا أمر الخروج Q الذي ليست له معاملات) ، ويمكن أن توضع فيها فاصلة أو مسافة خالية .

في شرح اسلوب استخدام الأوامر سوف يتم كتابة الأمر في صيغته العامة والتعبير عن المعاملات باسمها مثل filename كمعامل يكتب ليقوم المستخدم بكتابة اسم الملف بدلا منه .

عند كتابة المعامل بين القوسين علي الصورة [filename] فإن هذا المعامل يعد اختياريا بمعنى أنه يمكن كتابته و عدم كتابته اعتمادا علي رغبة المستخدم أو المطلوب تنفيذه .

المعاملات التي تستخدم مع الأوامر تكون واحدة من المعاملات التالية :

- معامل المشغل drive : عند وجود معامل المشغل فإنه يكتب بدلا منه رقم يحدد مشغل الأقراص الذي يتم العمل عليه ، وهذه الأرقام ترمز إلي مشغل الأقراص الذي يتم قراءة ملف منه أو كتابة ملف علي القرص الموجود به .

الأرقام هي 0 للتعبير عن مشغل الأقراص الأول A وهكذا بالترتيب مثل :

0 ..... A

1 ..... B

2 ..... C

3 ..... D

- معامل البايت byte : عند وجود معامل البايت في الأمر يكتب بدلا منه رقمان مكتوبان بنظام الستة عشر لكي يتم وضع هذه القيمة في عنوان من عناوين الذاكرة أو في أحد المسجلات ، أو أن يتم البحث عن و قراءة هذا البايت من مكان معين في الذاكرة أو أحد المسجلات .

- معامل السجل record : عند وجود هذا المعامل في صيغة أحد الأوامر فإن المستخدم يقوم بكتابة رقم أو ثلاثة أرقام بنظام الستة عشر للإشارة الي رقم السجل المنطقي علي القرص وعدد قطاعات القرص التي sectors يجب أن تكتب (في حالة الكتابة علي القرص) أو تقرأ (في حالة القراءة من علي القرص) .

- معامل القيمة value : ويكتب بدلا منه رقم بنظام الستة عشر لايزيد طوله عن أربعة أرقام digits ، وهذه القيمة تحدد عدد المرات التي يجب أن يقوم الأمر بتنفيذ وظيفته مكررة بعدد القيمة المكتوبة .

- معامل العنوان address : ويتكون هذا المعامل من قسمين يفصل بينهما النقطتين الرأسيتين colon .

\* القسم الاول يكتب فيه واحد من قيمتين :

رمز يتكون من حرفين يرمز الي المسجل .

أو رقم طوله أربعة أرقام يحدد عنوان المقطع segment .

\* القسم الثاني من معامل العنوان يكون علي صورة رقم طوله أربعة أرقام ويكتب فيه قيمة الازاحة offset .

وكل القيم في القسمين تكون مكتوبة بنظام الستة عشر مثل :

cs: 0100

04ba: 0100

ويلاحظ في هذين المثالين أن علامة النقطتين الرأسيتين تفصل بين اسم المقطع (القسم الأول) وبين الاذاحة ، وفي المثال الأول كتب المقطع علي صورة حرفية للتعبير عن اسم المسجل ، بينما اشتمل المثال الثاني علي رقم المقطع .

- معامل المدى range : المدى يبين نطاقا معيناً من الذاكرة سوف يتم التعامل معه ، وعلي ذلك فإن ما يتم كتابته بدلا من هذا المعامل سوف يكون عنوانين أحدهما يكون عنوان بداية المدى والثاني يكون عنوان نهاية المدى المطلوب العمل عليه .

قد يكتب عنوان واحد يحدد بداية المدى ، وبدلا من كتابة عنوان ثان يبين نهاية المدى المطلوب التعامل معه يكتب رقم يوضح طول المدى الذي يتم التعامل معه وفي هذه الحالة يجب أن يسبق الرقم الذي يحدد طول المدى حرف يبين أن الرقم يعبر عن طول ، ولا يعبر عن عنوان النهاية ، ويستخدم الحرف L لهذا الغرض .

مثال لكتابة المدى :

cs: 100 110

cs: 100 L10

CS: 100

في المثال الأول كتب العنوان الأول ليحدد بداية المدى والعنوان الثاني ليحدد نهاية المدى ، وفي المثال الثاني كتب العنوان الأول ليحدد بداية المدى وكتب طول المدى الذي سيتم التعامل معه لتحديد المدى .

في المثال الثالث لم يكتب سوي العنوان الأول فقط بينما لم يكتب أي من العنوان الثاني أو الطول ، وفي هذه الحالة الأخيرة يكون المدى ٨٠ باعتبارها هي الحالة المسجلة الافتراضية في البرنامج عند عدم تحديد الطول .

- معامل القائمة list : عندما تكتب صورة أمر من الأوامر والتي جواره معامل القائمة فإن معنى هذا أن المستخدم يجب عليه في هذه الحالة أن يقوم بكتابة سلسلة

من القيم التي تمثل كل منها بايت ، وكل قيمة من هذه القيم تكون علي صورة أرقام بنظام الستة عشر مثل :

fcs:100 41 52 45 54 42

والقائمة هنا في هذا المثال هي عبارة عن مجموعة الأرقام التي تبدأ برقم ٤١ ، وهي عبارة عن قائمة من الأرقام تمثل سلسلة من القيم التي تمثل كل منها بايتا واحدة .

- معامل السلسلة الحرفية string : السلسلة الحرفية هي مجموعة من الحروف المكتوبة ، وعندما يراد البحث عن مجموعة معينة من الحروف بين نص معين فان هذه الحروف يتم كتابتها كما هي دون تغيير وبأي عدد من الحروف .

في حالة كتابة أمر من اوامر debug التي تتعمل مع سلاسل الحروف مثل أمر البحث فان الأمر يكتب في صورته العامة علي الشكل :

S range strig

بما معناه أنه يراد البحث في المدى الذي تكتب قيمته عن سلسلة الحروف التي تكتب بدلا من معامل السلسلة الحرفية وتكتب بين علامتي تخصيص فردية أو مزدوجة مثل :

"This is an example for string"

"This is also an example for string"

ويمكن استخدام رموز الاسكي بدلا من الحروف .

**أوامر برنامج DEBUG**

الآتي بعد عرض لأوامر برنامج debug مرتبة أبجديا تبعا للغة الانجليزية :

**\* أمر التجميع**

يقوم هذا الأمر بتجميع منبهات الذاكرة للمعالج الدقيق مباشرة الي ذاكرة الحاسب

ويستخدم في عملية كتابة البرامج البسيطة بلغة الآلة ويكون علي الصيغة :

A [address]

والعنوان address هو الموقع الذي يطبع عنده منبه الذاكرة ، ولن يمكن التعرض للغة التجميع في هذا الموجز البسيط عن أوامر برنامج debug .

### \* أمر المقارنة Compare

يقارن جزءا من الذاكرة محددا بواسطة مدي range مع جزء آخر بنفس الحجم يبدأ عند عنوان معين address ويكتب علي الصورة .

C range address

وإذا كانت المساحتان المطلوب مقارنتهما متطابقتين فلن يظهر عرض أى بيان علي الشاشة ويعود البرنامج الي مشيرته منتظرا اصدار أمر آخر مما يفهم منه أنه لا يوجد اختلاف بين المساحتين .

إذا كانت هناك فروق في المقارنة فانها سوف تظهر علي الشاشة علي صورة رقم مكتوب يبين العنوان يليه رقم يمثل البايت المختلف .

مثال

C100, IFF, 300

C100 L100 300

وكل أمر من هذه الأوامر يقارن كتلة من الذاكرة مع كتلة أخرى لتظهر علي الشاشة الفروق بين محتويات الكتلتين .

### \* أمر عرض محتويات الذاكرة DUMP

يقوم أمر العرض (افراغ محتويات الذاكرة) بعرض محتويات كتلة معينة من الذاكرة علي الشاشة ، ويكتب علي صيغة :

D [range]

إذا تم تحديد مدي معين من العناوين مع أمر العرض D فإن محتويات هذا المدي المعين يتم عرضها علي الشاشة ، لكن إذا لم يتم تحديد مدي معين فإن عدد ١٢٨ بايت الأولي من العنوان الأول DS: 100 هي التي سوف يتم عرضها في البداية ، فإذا كتب الأمر مرة أخرى فإن عدد ١٢٨ بايت تالية سوف يتم عرضها علي الشاشة ، وهكذا يستمر الأمر مع تكرار اصدار هذا الأمر .

العرض يتم علي صورة قسمين ، وفي القسم الأول يظهر رقم يمثل البايث بنظام الستة عشر وفي القسم الثاني تظهر رموز أسكي حيث يظهر محتويات كل بايت علي صورة رمز أسكي ، وكل سطر يعرض ١٦ بايت .

مثال :

Dcs: 100 110

Dcs: 100 120

Dcs: 100 115

### \* أمر الادخال ENTER

الغرض من هذا الأمر هو وضع قيم معينة في عنوان معين من الذاكرة ، عن طريق تحديد هذه القيم وتحديد العنوان من الذاكرة الذي توضع فيه هذه القيم ، وصيغة الأمر هي :

E address (list)

عند استخدام أمر الادخال ENTER فإن القائمة الاختيارية [list] يمكن أن تكتب أو لا تكتب ، وفي حالة كتابتها فانها تحتوي علي قيم لوضعها في ذاكرة الحاسب عند العنوان المحدد address .

إذا كتب العنوان بدون تحديد قائمة المحتويات التي سوف توضع فيه فان البرنامج يقوم بعرض العنوان ومحتوياته الحالية ، ثم يتولي اعادة عرض العنوان فقط في السطر التالي وينتظر ريثما يتم اجراء واحدة من العمليات التالية :

١ - استبدال قيمة حالية لبايت بقيمة أخرى يتم كتابتها بواسطة المستخدم ، فإذا كانت القيمة التي قام المستخدم بكتابتها مكتوبة بالنظام السادس عشر ولكنها غير صحيحة القيمة (أو كانت مكتوبة بأكثر من عددين (2 digits) فإن الرقم الغير صحيح أو الذي تزيد مكوناته عن أكثر من رقمين لن يظهر له أي تأثير .

٢ . ضغط عصا المسافة للوصول إلي البايث التالي ، ولتغيير القيمة ببساطة يتم كتابة القيمة الجديدة .

٣ . إذا تطلب الأمر تغيير بايت سابق فان كتابة الواصلة hyphen تعيد البايث السابق ليكون في المتناول تغييره .

الضغط علي مفتاح الادخال عند أي موقع يعني الخروج من أمر الادخال .

مثال بفرض كتابة الأمر علي الشكل التالي :

ecs:100

وبفرض أن البرنامج عرض الصورة التالية :

04BA:0100 EB.-

لتغيير هذه القيمة الي 41 مثلا يتم كتابة هذا الرقم عند العلامة المضيئة مباشرة ليصبح الشكل علي الصورة :

048A: 0100 EB. 41-

للمرور علي البايث التالي يتم الضغط علي عصا المسافة ، والرجوع الي أي بايت سابق يتم كتابة الوصلة .

\* أمر الملء fill

الغرض منه هو ملء عناوين معينة في مدي محدد بالقيم التي تنوع في القائمة وصيغته .

f range list

وإذا كان المدي يحتوي علي عدد من البايت أكثر من عدد القيم المدرجة في القائمة list فإن محتويات القائمة سوف تتكرر حتي يمتلئ عدد البايت في المدي المحدد .  
إذا كانت القائمة تحتوي علي قيم أكثر من عدد البايت في المدي المحدد .

إذا كانت القائمة تحتوي علي قيم أكثر من عدد البايت في المدي فإن القيم التي تزيد عن عدد البايت سوف همل .

إذا كانت الذاكرة في المدي غير متاحة أو سيئة أو غير موجودة فإن رسالة خطأ سوف تحدث في كل المواقع التالية .

مثال : لنفرض أنه قد تم كتابة الأمر عي الصورة التالية :

f04BA: 100 1 100 42 45 52 45 41

فان استجابة البرنامج لتنفيذ هذا الامر سوف تكون علي شكل وضع القيم الخمسة المذكورة في القائمة (42 45 52 54 41) في مواقع الذاكرة بدءاً من الموقع 04BA: 100 وبطول ١٠٠ بايت اي أن هذه القيم سوف تتكرر عشرين مرة .

### \* أمر الذهاب Go

يقوم هذا الأمر بتنفيذ برنامج موجود في الذاكرة وضيغته

g [=address (address)]

إذا كتب الأمر علي صورج حرف g فقط فان البرنامج الحالي في الذاكرة سينفذ ،  
وإذا ماوضع العنوان فان تنفيذ البرنامج سوف يبدأ من عند العنوان المحدد ، وعلامة  
التساوي مطلوبة مثال .

g = c800:5

### \* أمر ستة عشر hex

يقول بتنفيذ عمليات حسابية علي قيمتين محددتين وضيغته

h value value

عند كتابة هذا الأمر فإن البرنامج يقوم بجمع القيمتين أولاً ثم يطرح القيمة الثانية من القيمة الأولى ويعرض ناتج العمليتين في سطر واحد علي شكل رقمين مكتوبين بنظام الستة عشر الأول منهما هو ناتج عملية الجمع والثاني هو ناتج عملية طرح القيمتين المكتوبتين في الأمر مثل :

h19F 10A

الذي تكون نتيجته هي :

02A9 0095

### \* أمر الادخال input

يقول بادخال وعرض بايت واحد من المنفذ المحدد في القيمة وصيغته

i value

مثال :

i 2FB

### \* أمر التحميل load

يستخدم لتحميل ملف إلي الذاكرة وصيغته

l [address (drive:record record)]

والملف يجب أن يكون له اسم يكتب اما عند بداية البرنامج أو باستخدام أمر التسمية من خلال برنامج debug .

### \* أمر التحريك move

يقوم بتحريك كتلة من الذاكرة محددة بمدي معين إلي موقع يبدأ في عنوان معين في الذاكرة وصيغته .

m range address

مثال بكتابة الأمر التالي :

mcs: 100 100 cs:500

سوف يستجيب البرنامج للأمر بتحريك العنوان cs: 100 الي العنوان cs: 510 ثم وهكذا حتي يتم نقل المساحة كلها بطول المدي المحدد .

**\* أمر التسمية name**

أمر التسمية يقوم بوظيفتين ، فهو أولا يحدد اسم الملف الذي سيتم استخدام أمر التحميل لتحميله L أو أمر الكتابة W لكتابه علي القرص ، وثانيا يقوم أمر التسمية بتغيير اسم الملف أو تسميته اذا لم يكن له اسم وصيغته .

n [filename (filename ...)]

مثال

n file 1.exe

L

مثال آخر

n file 2.dat file3.dat

**\* أمر الاخراج output**

يرسل بايت محدد إلي منفذ خارجي وصيغته

O value byte

مثال

اذا افترضنا الرغبة في ارسال البايث الذي قيمته 4F إلي المنفذ الخارجي ذي الرقم 2F8 فإن الأمر يكتب علي الصورة :

O 2F8 4F

## \* أمر الاستمرار والتقدم **proceed**

ينفذ حلقة أو تعليمات حرفية متكررة أو مقاطعة برنامج أو برنامجا فرعيا أو يحافظ علي التتبع خلال تعليمة معينة وصيغته .

P [=address (number)]

وإذا لم يتحدد عدد مرات التنفيذ فإن القيمة المفترضة تكون الواحد مثال

P= 143F

## \* أمر الخروج من البرنامج **quit**

ينهي عمل برنامج debug ويكتب علي صورة

Q

بدون أي معامل فيعود البرنامج إلي مشيرة نظام تشغيل القرص .

## \* أمر المسجل **register**

يعرض محتويات واحد أو اكثر من مسجلات وحدة المعالج الدقيق وصيغته .

R [registername]

وأسماء المسجلات هي :

AX BP SS

BX SI CS

CX DI IP

DX DS PC

SP ES F

ويمكن تغيير محتويات أي مسجل .

## \* أمر البحث search

ويبحث في مدي معين عن قائمة من البايت وصيغته .

S range list

وأمر البحث قد يحتوي في قائمته علي بايت أو أكثر بين كل بايت والآخر مسافة خالية مثال :

Scs: 100 110 41

سوف يظهر البرنامج شيئاً مشابهاً للآتي :

04BA:0104

04BA:010D

## \* أمر التتبع trace

ينفذ ايعازاً ويعرض محتويات المسجلات والرايات وصيغته

T [=address] [value]

إذا تضمن أمر التتبع عنواناً address فإن التتبع يحدث عند هذا العنوان المحدد والذي تسبقه علامة التساوي .

الخيار value يسبب قيام برنامج التصحيح بتنفيذ وتتبع عدد من الخطوات المحددة بالقيمة value .

مثل لعرض محتويات المسجلات والرايات :

- t

مثال لتنفيذ ١٦ تعليمة بدءاً من العنوان 011a :

t=011a 10

## \* أمر فك التجميع unassemble

يقوم بعرض الجمل الأصلية والتي تتقابل مع مجموعة من البايتات المعينة وصيغته :

u [rang]

يظهر ناتج الأمر علي شكل قائمة ملف جري تجميعه ، وفي كل مرة يكتب الأمر بدون معاملات فإن عشرين بايتا فقط بنظام ترقيم الستة عشر سوف تظهر في العنوان الأول ثم تليها عشرين بايتا آخرين .

إذا تضمن الأمر كتابة قيمة المدي range فكل بايت في هذا المدي سوف يظهر .

مثال :

OOU4BA: 100 L10

### \* أمر الكتابة write

يكتب علي القرص الملف الذي يجري التعامل معه وصيغته

W [address (drive:record record)]

إذا كتب الأمر ومعه معامل العنوان فقط فان الملف سوف يكتب بدءا من هذا العنوان ، ويجب تسمية الملف باستخدام أمر التسمية قبل استخدام أمر الكتابة .

إذا استخدمت المعاملات مع أمر الكتابة فيجب ملاحظة أن مشغل الاقراص الأول A سوف يرمز له بالرمز O وهكذا الحال بالنسبة لباقي المشغلات .

مثال :

WCS:100 1 37 2 B

### \* أوامر التعامل مع الذاكرة الموسعة

يحتوي برنامج التصحيح debug أربعة أوامر تبدأ بحرف X للتعامل مع الذاكرة الموسعة .

### \* أمر تقسيم الذاكرة الموسعة allocate expanded memory

يقوم بتقسيم عدد من صفحات محددة للذاكرة الموسعة وصيغته :

XA [count]

حيث count تحدد عدد صفحات ذات ستة عشر كيلو بايت للذاكرة الموسعة ، وإذا كان العدد المحدد من الصفحات متاحا فان البرنامج يعرض رسالة برقم ستة عشر تشير إلي العدد الذي تم انشاؤه ، وإلا فان البرنامج سوف يعرض رسالة خطأ .  
مثال لتخصيص ٨ صفحات في الذاكرة الموسعة .

Xa8

ويعرض البرنامج رسالة مشابهة للرسالة التالية

handle created -0003

**\* أمر الغاء تقسيم الذاكرة الموسعة Deallocate Expand Memory**

هو أمر يلغي تخصيص ماهو في المتناول من صفحات الذاكرة الموسعة وصيغته .

XD [handle]

حيث handle هو عدد يحدد ما يراد الاؤه ويكتب العدد بنظام الستة عشر مثال :

XD0003

واذا نفذ الأمر صحيحا فان البرنامج يعرض الرسالة :

Handle 0003 deallocated

**\* أمر تسطير صفحات الذاكرة الموسعة Map Expanded Memory Pages**

يقوم بعمل خريطة لصفحة منطقية في الذاكرة الموسعة لجعلها في المتناول كصفحة حقيقية وصيغته .

XM [lpage] [ppage] [handle]

حيث lpage تحدد رقم الصفحة المنطقية للذاكرة الموسعة التي يراد تسطيرها الي صفحة فيزيائية ppage ، وحيث ppage تحدد عدد الصفحات الفيزيائية التي يراد تسطير الصفحة المنطقية لها مثال :

وإذا كان الأمر صحيحا تظهر رسالة مثل :

logical 05 mapped to physical page 02

## \* أمر عرض حالة الذاكرة الموسعة display Expanded Memory Status

يعرض حالة الذاكرة الموسعة كما سبق الإشارة اليه وصيغته :

-X S

## استعمال برنامج Debug لمعاينة الذاكرة

عند ظهور مشيرة نظام تشغيل القرص DOS التي تكون في الغالب علي شكل حرف واحد يدل علي مشغل الاقراص الذي بدأ منه النظام العمل وأمامها علامة أكبر من يقوم المستخدم بكتابة أمر تشغيل البرنامج بطبع الأمر Debug ثم الضغط علي مفتاح الادخال Enter .

يجب أن يكون البرنامج موجودا في القرص الموضوع في مشغل الاقراص الذي يتم العمل عليه ، وبعد الضغط علي مفتاح الادخال يقوم الحاسب باستدعاء البرنامج لتشغيله ، وبعد برهة وجيزة تظهر مشيرة برنامج Debug وهي عبارة عن واصلة (شرطة سفلية طويلة) hyphen .

C > DEBUG

يحتوي البرنامج علي مجموعة من الأوامر الذي يختص كل واحد فيها بعمل معين ، وللإطلاع علي هذه الاوامر واستعراضها يتم كتابة علامة الاستفهام ثم الضغط علي مفتاح الادخال .

استعمال الاوامر في البرنامج يتم كتابة الحرف الأول من الأمر يليه عدد من المعاملات التي تعتمد علي الأمر .

أمر Dump يعرض ماتحتويه صفوف الذاكرة مكتوبة علي صورة البت تلو الاخر ، ولاستعمال الأمر Dump ، يتم كتابة الحرف الأول منه d ثم الضغط علي

مفتاح الادخال .

نتيجة تنفيذ هذا الامر هو ظهور محتويات مساحة ١٢٨ بايت من ذاكرة الحاسب الشخصي الذي يتم العمل عليه مرتبة .

كل سطر من خرج أمر Dump يحتوي علي ١٦ بايت من الذاكرة ، ويحتوي السطر علي ثلاثة أعمدة ، وكل عمود من هذه الأعمدة يحتوي علي بيانات مختلفة :  
العمود الأول في السطر يحمل عنوان البايث الأول .

العمود الثاني في السطر يحتوي علي ١٦ بايت من الذاكرة ابتداء من العنوان المعروض في العمود الأول .

العمود الثالث في السطر يشتمل علي رموز آسكي ASSII التي تطابق كل منها واحدة من الستة عشر بايت الموجودة في السطر في العمود الثاني .

يلاحظ أن عناوين الذاكرة وقيمة كل بايت مكتوبة في نظام ترقيم الستة عشر (hexadecimal) .

بكتابة الامر Dump ثم الضغط علي مفتاح الادخال يعرض الأمر 128 بايتا من مساحة الذاكرة السفلي المعروفة بمساحة بيانات أساسيات نظام الادخال والايخراج BIOS.

بين هذه البايثات توجد بيانات مكتوبة بنظام ترقيم الستة عشر عن كمية الذاكرة الموجودة في الحاسب الذي يتم العمل عليه ، وبيانات عدد مشغلات الاقراص في النظام ومساحة تخزين ضربات مفاتيح لوحة المفاتيح ، ونظام العرض المرئي ، والوقت الحالي .

والآن اطبع الامر التالي واضغط علي مفتاح الادخال .

-d fe00.0

يعرض أمر debug مساحة ١٢٨ بايت من الذاكرة العليا في ذاكرة القراءة فقط ROM . وهذا الجزء هو مكان أساسيات الادخال والايخراج في الحاسب وفيه يمكن قراءة حقوق طبع البرنامج .

## تغيير الذاكرة

يسمح أمر برنامج التصحيح debug بمعاينة أقسام الذاكرة التي يستطيع نظام تشغيل القرص الوصول إليها ، كما يمكن تغيير محتويات الذاكرة .

لتغيير الذاكرة فإن مكان ذاكرة العرض المرئي تعد أفضل مكان لبداية التجارب ، فإن أى تغيير يحدث فيها يظهر على شاشة وحدة العرض المرئي مباشرة بعد الضغط على مفتاح الادخال .

مثال لنكتب الامر التالي ونضغط علي مفتاح الادخال :

```
-f b800:0 FA0 21 ce
```

يطلب هذا الأمر من برنامج التصحيح debug وضع بايت له الرمز 21 ce في ذاكرة الحاسب وتكراره في مساح قدرها fa0 بايت ( ٤٠٠٠ بايت في النظام العشري) بدءا من الموقع b800:000 ، وهو موقع ذاكرة العرض المرئي الملون .

يبدل هذا الأمر اللون الخلفي إلي الاحمر ، ويسبب امتلاء الشاشة بعلامات تعجب صفراء اللون .

ملحوظة (يستخدم الموقع b000:0000 علي الحاسب ذي الشاشة أحادية اللون) .

السبب في تحديد قدر المساحة ليكون ٤٠٠٠ بايت هو أن :

العنوان b800:0000 ( أو b000:0000 للحاسب ذي الشاشة أحادية اللون) هو بداية ذاكرة العرض المرئي ، ولما كانت شاشة العرض المرئي ذات النصوص تملك ثمانين عمودا مرتبة في خمسة وعشرين صفا بما يشكل ألفى حرف على الشاشة .

ولما كان كل حرف يأخذ اثنين من البايت لتخزينه في الذاكرة ، فالبايت الأول هو قيمة الحرف في شفرة الترميز ascii وهو الحرف الذي سيتم عرضه ، والبايت الثاني هو سمة الحرف على الشاشة (لضبط اللون وكثافته ووميضه و...) .

لذلك فان شاشة النص ذات الثمانين عمودا تساوى (٤٠٠٠ بايت) ، وهي التي لها القيمة fa0 بايت في نظام ترقيم الستة عشر .

المقابل للحرف الذى له رقم ٢١ (فى نظام ترقيم الستة عشر) فى نظام الترميز ASCII هو علامة تعجب ، وسمة شاشة العرض المرئى التى تقابل الرمز ce هو ظهور حرف يومض باللون الأصفر على خلفية حمراء ، وهكذا تمتلئ مساحة الشاشة (٢٠٠٠ موضع) بعلامات تعجب صفراء وامضة على خلفية حمراء .

يمكن التأكد مما يحتويه المساحة b800 من الذاكرة بطبع أمر dump التالى :

-d b800:0

إذا كانت الشاشة أحادية اللون يوضع الرقم b000 بدلا من الرقم b800 .

تظهر على الشاشة نتيجة تنفيذ هذا الأمر على صورة ١٢٨ بايت محتوية على الرمز ce 21 متكررا الواحد بعد الآخر .

## فحص الذاكرة الموسعة

إذا كان الحاسب يحتوى على بطاقة ذاكرة موسعة مركبة فيه فإن برنامج debug من الاصدارين الخامس والسادس يحتوى على أوامر تتيح العمل مع هذا النوع من الذاكرة كما لو كنت تعمل مع الذاكرة التقليدية .

يسمح الأمر XS بعرض حالة سواقة الذاكرة الموسعة فبتنفيذ برنامج debug وطبع

الأمر التالى ثم الضغط على مفتاح الادخال :

-XS

عند وجود ذاكرة موسعة سوف يظهر على الشاشة مايشبه هذا البيان

Handle xxxx has xxxx page allocated

Physical page xx = Frame segment xxxx

Physical page xx = Framw segment xxxx

-----

xx of a total xxx EMS pages have been allocated

xx of a total xxx EMS handles have been allocated

الرموز xx هي قيم حقيقية ، وفي حالة عدم امتلاك ذاكرة موسعة سيعرض برنامج debug الرسالة التالية :

EMS not installed

## استعمال أمر استعراض الذاكرة MEM

بينما يعرض أمر التصحيح debug محتويات الذاكرة فإن أمر استعراض بيانات ومعلومات الذاكرة MEM يبين البرامج وسواقات الأجهزة الموجودة في الذاكرة .

كان أمر استعراض الذاكرة MEM معروفا منذ الاصدار الرابع من نظام تشغيل القرص DOS 4 ، وهو أول اصدار من اصدارات نظام تشغيل القرص ساند فعليا الذاكرة الموسعة والذاكرة الممتدة .

يعطى الأمر MEM بدون خيارات تقريرا عن الذاكرة المتوفرة في موجز سريع يبين كيفية استعمال الذاكرة داخل الحاسب .

```
c:\>mem
```

```
655360 bytes total coventional memory
```

```
655360 bytes available to MS-DOS
```

```
637600 largest executable program size
```

اذا وجدت ذاكرة موسعة أو ممتدة يعطى الأمر mem تقريرا عن كمية الذاكرة الكاملة في الحاسب ، كما يعطى أيضا بيانات بالكمية المتوفرة التي لم تستعمل من الذاكرة بعد .

بالطبع ، كلما كانت الذاكرة المتوفرة أكبر كلما أمكن القيام بأعمال أكثر ولكن المسألة ليست ببساطة مجرد امتلاك ٦٤٠ كيلو بايت من الذاكرة التقليدية و ٢ مليون بايت من الذاكرة الممتدة فقط ، ولكن الموضوع الأهم هو ماهى الكمية التي يمكن استخدامها من ذاكرة الحاسب ؟

في المثال السابق يتضح أن حجم أكبر برنامج يمكن تنفيذه على الحاسب الشخصي الذي تظهر بياناته (largest executable program size) هو 637 كيلو بايت ويعود فضل هذا إلي نظام تشغيل القرص في اصداراته الجديدة ، ويمكن لهذه الكمية أن تزيد إلي أكثر من هذا الحد وهو الأمر الذي يتناوله الشرح فيما بعد في فصول الكتاب .

## معاملات أمر استعراض الذاكرة

يملك الأمر MEM في الاصدار الخامس من نظام تشغيل القرص ثلاث صيغ اختيارية لكتابة الأمر بالإضافة إلي كتابته منفردا ، بينما يملك نفس الأمر مع نظام تشغيل القرص في الاصدار السادس معاملات أكثر ، وسوف نبدأ بالمعاملات في الاصدار الخامس ثم نثنى بمعاملات الاصدار السادس .

### خيارات الاصدار الخامس program, debug, classify

يعطى الخيار program / قائمة بكل البرامج الموجود في الذاكرة ، ومساحات بيانات النظام وسواقات الأجهزة المركبة فيها مع أماكنها وأحجامها ، وهذه البيانات تكون مكتوبة على صورتين (بالنظام العشري ونظام الستة عشر) ، كما يعطى الأمر على هذه الصورة الكمية الكاملة للذاكرة التقليدية الموسعة والممتدة في الحاسب مع بيان الكمية المتوفرة لكل منها .

يعطى الخيار debug / نفس المعلومات المتاحة في الاختيار السابق بالإضافة أيضا إلي عرض سواقات الأجهزة والنظام (كالطباعة والساعة) .

يعطى الخيار classify / قائمة بأسماء وأحجام كل البرامج الموجودة في الذاكرة .

أن خرج الأمر Mem عند استعماله مع أحد الخيارات الاختيارية قد يكون طويلا جدا، ولذلك يفضل استخدامه من خلال رمز التمرير (|) ومع أمر المرشح More عند اصدار الأمر Mem مع أى من الخيارات السابقة (في الاصدار الخامس من نظام تشغيل القرص) .

كمثال لذلك يكتب الأمر على الصورة التالية :

c:\> mem / program (!) more

بعد الضغط على مفتاح الإدخال تظهر الشاشة الأولى من المعلومات التي تعرض البرامج وسواقات الأجهزة الموجودة في الذاكرة .

الأعمدة الأربعة الموجودة في العرض تبين مكان (عنوان) كل برنامج أو سواقة جهاز في الذاكرة ، اسم البرنامج أو سواقة الجهاز ، والحجم بالبايتات (في نظام الستة عشر) ، ونوع الذاكرة .

بالضغط على قضيب المسافة يتم عرض شاشة أخرى من المعلومات ، وتعرض الشاشة الأخيرة كمية الذاكرة التقليدية والموسعة والممتدة في الحاسب والكمية المتوفرة من كل منها .

خرج الأمر Mem/classify يكون أيضا طويلا ويفضل كتابته أيضا على الصورة:

c:\>mem / classify (!) more

في العرض تتضح البرامج الموجودة في الذاكرة التقليدية والمساحة التي يحتلها كل برنامج .

كمثال لذلك قد نرى أربعة برامج وأربع مساحات فارغة (FREE) والبرامج الأربعة هي :

برنامج MSDOS الذي يحتل مساحة 4.8 كيلو بايت .

وبرنامج (COMMAND.COM) الذي يحتل مساحة 4.8 كيلوبايت .

وبرنامج سواقة (الماوس) مستخدمة مساحة قدرها 14.6 كيلو بايت .

والبرنامج المقيم في الذاكرة DOSKEY مستخدما مساحة قدرها كيلو بايت .

مجموعة المساحات الفارغة (FREE) هو في هذا المثال مساحة قدرها 556.3 كيلو

بايت وكلها متوفرة لتطبيقات نظام تشغيل القرص DOS .

يمكن اختصار الخيارات classify, / debug, / program وكتابتها على الصورة المختصرة /c, /d, /p بالترتيب بدلا من كتابة الخيار بالتفصيل .

أمر استعراض معلومات الذاكرة mem يبين كمية الذاكرة التقليدية والموسعة والممتدة الموجودة فى الحاسب ، ومقدار الكمية المتوفرة من كل منها ، وعند استخدام واحد من الخيارات مع هذا الأمر فإن الأمر mem يعطى بيانا عن البرامج الموجودة فى الذاكرة أيضا ومكان وجودها وكمية الذاكرة التى يحتلها كل برنامج .

يقوم هذا الأمر أيضا باعطاء بيان هام عن توافر بندين آخرين فى الحاسب (اذا كانا موجودين) ، وهما مساحة الذاكرة العالية (HMA) ، ومجموعات الذاكرة العليا (UMB) .

مساحة الذاكرة العالية HIMEM هى المساحة الأولى المؤلفة من ٦٤ كيلو بايت من الذاكرة الممتدة فى حاسب ذى معالج من المعالجات القوية 80286,80386 أو أعلى .

عند تركيب برنامج لادارة الذاكرة الممتدة : مثل برنامج سواقة الجهاز HIMEM الذى يأتى مع نظام تشغيل القرص فى اصداراته الجديدة يصبح بإمكان نظام تشغيل القرص الوصول الى مساحة الذاكرة العالية hma ، ويمكن بالتالى نقل جزء من نظام تشغيل القرص (البرنامج MSDOS) الى هذه المساحة hma محررا بذلك مساحة جيدة من الذاكرة التقليدية .

إذا كان الحاسب يحتوى على المعالج 80286,80386 أو أعلى مع ذاكرة ممتدة وتم تركيب مدير الذاكرة الممتدة فإن السطر الأخير من الأمر mem سيكون 64kb high memory area available .

من الطبيعى أن خرج أمر استعراض بيانات الذاكرة سوف يختلف من حاسب الى آخر ، كما سوف يختلف اذا كان نظام تشغيل القرص قد جرى نقله الى المساحة الذاكرة العالية hma ، واذا كان هناك شئ آخر يستعمل المساحة hma ستظهر رسالة تبين أن مساحة الذاكرة العالية قيد الاستخدام .

عند انشاء مجموعات الذاكرة العليا umb لتخزين برامج سواقات الأجهزة والبرامج المقيمة فى الذاكرة سوف يعطى خرج الأمر mem/ classify قائمة معلومات عن مجموعات الذاكرة العليا umb وكمية الذاكرة الفارغة المتوفرة ومجموعة الذاكرة المتوفرة للبرامج .

لكن هذا الحديث سابق لأوانه فى هذا المكان اذ أن الفصول التالية سوف تتناول هذا الأمر بكثير من التفاصيل .

## معاملات أمر استعراض الذاكرة فى الاصدار السادس من نظام تشغيل القرص .

اختلفت المعاملات الموجودة مع أمر استعراض معلومات الذاكرة فى الاصدار السادس DOS 6 من نظام تشغيل القرص عنها فى المعاملات التى كانت موجودة مع نفس الأمر فى الاصدار الخامس قليلا .

لوحظ أن المستخدم يضطر الى استخدام أمر الترشيح more دائما عند تنفيذ الأمر وذلك بسبب الكم الكبير من المعلومات التى ترد الى الشاشة بسرعة لايسطيع ملاحظتها. لذلك أضيف خيار الصفحة الجديد /PAGE بحيث يتوقف العرض على الشاشة بعد امتلاء الصفحة ، واستتبع ذلك استبعاد خيار البرنامج /PROGRAM من الخيارات المستخدمة مع أمر استعراض الذاكرة .

تسهيلا لمعرفة كمية الذاكرة الخالية مباشرة فى كل من الذاكرة التقليدية والذاكرة العليا، فقد أضيف خيار جديد ليعطى بيانا سريعا وموجزا عن المساحات الفارغة فى كل من الذاكرتين ، وهو الخيار (حر) free .

بقى خيار debug كما هو وان كانت امكانيات قد زادت فى قيامه بعرض مقاطع الذاكرة وبيانات المشغلات الداخلية ومعلومات أخرى عن توزيع البرامج على المقاطع المختلفة من الذاكرة .

كمابقى خيار التقسيم classify كما هو أيضا متاحا كخيار من خيارات البرنامج ليبين

تقسيمات استخدام البرامج للذاكرة مع تقديم ملخص عن استخدامات الاجزاء المختلفة للذاكرة ، مع بيان كتلة الذاكرة المتاحة للاستخدام .

أضيف الى الخيارات خيار آخر جديد يتمكن من عرض قائمة تفصيلية لجزء من الذاكرة هو خيار MODULE وهو الذى يكتب اختصارا M/ ويتبعه كتابة نقطتين رأسيين (: colon) بعد الخيار يليها رقم يحدد القطاع المراد استعراض تفاصيله .

Modules using memory below 1 MB:

Name	Total	= Conventional	+ Upper Memory
system	11885 (12k)	11821 (12k)	64 (0k)
himem	1200 (1k)	1200 (1k)	0 (0k)
emm386	3248 (3k)	3284 (3k)	0 (0k)
clmmand	2896 (3k)	2896 (3k)	0 (0k)
ansi	4208 (4k)	0 (0k)	4208 (4k)
gmouse	14928 (15k)	0 (0k)	14928 (15k)
ramdrive	1200 (1k)	0 (0k)	1200 (1k)
doskey	4144 (4k)	0 (0k)	4144 (4k)
smartdrv	25632 (25k)	0 (0k)	25632 (25k)
free	680736 (665k)	636208 (62k)	44528 (43k)

Memory Summary:

Type of Memory	Size	= Used	+ Free
Conventional	655365 (640K)	19152 (19K)	636208 (621K)
Upper	94704 (292K)	50176 (49K)	44528 (43K)
Adapter RAM/ROM	298512 (292K)	298512 (292K)	0 (OK)
Extended (XMS)	3145728 (3072K)	794624 (776K)	2351104 (2296K)
Extended (EMS)	0 (ok)	0 (ok)	0 (ok)
Total memory	4194304 (4096k)	1162464 (1134k)	3031840 (2961k)

Largest executable program size 636192 (621k)

Largest free upper memory block 44448 (43)

MS-DOS is resident in the high memory area

وملف تجهيز النظام المستخدم (Config. Sys) هو التالي:

```
device = c:\himem, sys
dos - high, umb
device = c:\dos\emm386, noms
files = 20
buffers = 20
shell = c:\dos\command, com c:\dos\p
stacks = 0,5
devicehigh = c:\dos\mouse, sys
devicehigh = c:\dos\gmouse, sys
```

- \* هناك أدوات تسمحان بمعاينة الذاكرة هما خدمات أمر التصحيح debug ، وأمر استعراض معلومات الذاكرة mem .
- \* يمكن اعتبار الأمر mem من الأوامر البسيطة التي يستعملها المستخدم العادي، أما برنامج debug فهو واحدة من الأدوات التي يستخدمها المستخدم الأكثر خبرة .
- \* يسمح برنامج التصحيح debug بفحص وتغيير الذاكرة وانشاء برامج صغيرة وتحميل أجزاء من القرص إلي الذاكرة وحفظ الذاكرة إلي القرص وتنفيذ مهمات قوية أخرى مختلفة .
- \* يحتوى برنامج التصحيح على عدد من الأوامر من بينها أمر الاستعراض dump الذى يسمح برؤية البايتات المخزنة فى عناوين الذاكرة معروضة على صورة أرقام فى نظام الستة عشر .
- \* أمر استعراض بيانات الذاكرة الموسعة XS يسمح بعرض معلومات عن الذاكرة الموسعة وهو من أوامر برنامج التصحيح debug ويعمل بعد تشغيل برنامج التصحيح .
- \* يبين أمر استعراض الذاكرة mem مجموع الذاكرة التقليدية والموسعة والممتدة فى الحاسب الشخصى والكميات المتوفرة منها .
- \* أمر استعراض الذاكرة mem له ثلاثة خيارات فى الاصدار الخامس من نظام تشغيل القرص .
- يعرض المعامل الاختيارى /program مع الأمر mem خلاصة عن البرامج وسواقات الأجهزة الموجودة فى الذاكرة كما يعرض أماكنها وأحجامها وأنواعها .
- ينتج المعامل الاختيارى /debug نفس بيانات المعامل السابق مع عرض سواقات أجهزة النظام .

يبين المعامل الاختياري /debug مع الأمر mem البرامج المحملة فى الذاكرة وكمية الذاكرة التى تحتلها وتفيذ هذه المعلومات فى نقل سواقات الأجهزة والبرامج المقيمة فى الذاكرة الى مجموعات الذاكرة العليا .

\* فى اطار تعديل نظام تشغيل القرص فى اصداره السادس فقد أضيفت الى أمر استعراض الذاكرة MEM خيارات جديدة توسع من امكانياته مع زيادة امكانيات اختيارات الموجودة فيه مسبقا .

\* أضيف خيار الصفحة /PAGE ليتوقف العرض على الشاشة بعد امتلاء الصفحة .  
\* استبعد خيار البرنامج /PROGRAM من الخيارات المستخدمة مع أمر استعراض الذاكرة .

أضيف خيار (حر) free ليعطى بيانا سريعا وموجزا عن المساحات الفارغة فى الذاكرة .

بقى خيار debug كما هو وان كانت امكانياته قد زادت ، كما بقى خيار التقسيم classify كما هو أيضا .

أضيف خيار MODULE ليتمكن من عرض قائمة تفصيلية لجزء من الذاكرة .

