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 أخري مختلفة ، كما تناول أمر استعراض الذاكرة mem لعرض بيانات الذاكرة التقليدية والموسعة 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 أخري ، ومن بين تطبيقات المنافع الأخري التي تتيح مثل هذا الأداء 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 (الواصلةhyphen) 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 ، ويلي هذا الحرف واحد أو أكثر من المعاملات ، واذا 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 بمعني 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 الاقراص الذي يتم العمل عليه ، وهذه الارقام ترمز إلي مشغل الأقراص الذي يتم قراءة ملف منه أو كتابة ملف علي القرص الموجود به .

الارقام هي o للتعبير عن مشغل الاقراص الأول A وهكذا بالترتيب مثل :

o ..... 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 الازاحة ، وفي المثال الأول كتب المقطع علي صورة حرفية للتعبير عن اسم المسجل ، بينما اشتمل المثال الثاني علي رقم المقطع .

- معامل المدي range : المدي يبين نطاقا معينا من الذاكرة سوف يتم التعامل معه ، وعلي ذلك فان مايتم كتابته بدلا من هذا المعامل سوف يكون عنوانين أحدهما يكون عنوان بداية المدي والثاني يكون عنوان نهاية المدي المطلوب العمل عليه .

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

مثال لكتابة المدي :

```
cs: 100 110

cs: 100 L10

CS: 100
```

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

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

- معامل القائمة list : عندما تكتب صورة أمر من الأوامر والي جواره معامل القائمة فان معني 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 المدي 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

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

اذا 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 أخري 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]

اذا تم تحديد مدي معين من العناوين مع أمر العرض D فان محتويات هذا المدي المعين يتم عرضها علي الشاشة ، لكن اذا لم يتم تحديد مدي معين فان عدد ١٢٨ بايت الأولي من العنوان الأول DS: 100 هي التي سوف يتم عرضها في البداية ، فاذا كتب الأمر مرة أخري فان عدد ١٢٨ بايت تاليـة سوف يتم عرضها علي الشاشة ، وهكذا يسـتمر الأمر مع تكرار اصدار هذا الأمر .

العـرض يتم علي صورة قـسـمين ، وفي القسم الأول يظهـر رقم يمثل البـايت بنظام الستـة عشر وفي القسم الثـاني تظهر رموز أسكي حـيث يظهر محـتويات كل بايت علي صورة رمز آسكي ، وكل سطر يعرض ١٦ بايت .

مثال :

Dcs: 100 110

Dcs: 100 120

Dcs: 100 115

*** أمر الادخال ENTER**

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

E address (list)

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

اذا كتب العنوان بدون تحـديد قائمـة المحتويات التي سـوف توضع فيه فـان البرنامج يقوم بعرض العنوان ومـحتوياته الحالية ، ثم يتولي اعـادة عرض العنوان فقط في السطر التالي وينتظر ريثما يتم اجراء واحدة من العمليات التالية :

١- استبدال قيمة حالية لبايت بقيمة أخري 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 البايت التالي ، ولتغيير القيمة ببساطة يتم كتابة القيمة الجديدة .

٣. إذا تطلب الأمر تغيير بايت سابق فان كتابة الواصلة hyphen تعيد البايت 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 البايت 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 توع في القائمة وصيغته.

f range list

وإذا كـان المدي يحـتوي علي عـدد من البـايت أكـثر مـن عدد القـيم المدرجـة 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 اكـثر من عدد البايت في المدي فان القيم التي تزيد عن عدد البايت سوف همل .

اذا 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 استجـابة البرنامج لتنفيذ هذا الامر سوف تكون علي شكل وضع القـيم الخمسة المذكورة في القائمة (41 54 52 45 42) في مواقع الذاكرة بدءا 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)]

اذا كتب الأمر عـلي صورج حرف g فقط فان البرنامج الحـالي في الذاكرة سينفذ ، واذا ماوضع العنوان فان تنفـيذ البرنامج سوف يبدأ من عند العنوان المحـدد ، وعلامة التساوي مطلوبة مثال .

g = c800:5

*** أمر ستة عشر hex**

يقول بتنفيذ عمليات حسابية علي قيمتين محددتين وصيغته

h value value

عند كتابة هذا الأمر فإن البرنامج يقوم بجمع القيمتين أولا ثم يطرح القيمة الثانية من القيمة الأولي 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 البـرنامج للأمر بتحريك العنوان 100 :cs الي العنوان 510 :cs ثم وهكذا حتي يتم نقل المساحة كلها بطول المدي المحدد .

*** أمر التسمية name**

أمر التسـمية يقوم بوظيفتين ، فـهو أولا يحدد اسم الملف الذي سيتم اسـتخدام أمر التحميل لتحميله L أو أمر الكـتابة W لكتابتـه علي القرص ، وثانيا يقـوم أمر التسمـية 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 افتـرضنا الرغـبة في ارسـال البايت الذي قـيمـته 4F إلي المنفـذ الخـارجي 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)]

واذا 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 أى 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 شيئا مشابها للآتي :

04BA:0104

04BA:010D

* **أمر التتبع trace**

ينفذ ايعازا ويعرض محتويات المسجلات والرايات وصيغته

T [=address] [value]

اذا تضمن أمر التتبع عنوانا address فان 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 بدءا 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 الأمـر علي شكل قائمـة ملف جري تجمـيعه ، وفي كل مـرة يكتب الأمر بدون معـاملات فإن عشرين بـايتا فقط بنظام ترقيم السـتة عشر سـوف تظهر في العنوان الأول ثم تليها عشرين بايتا آخرين .

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

مثال :

OUO4BA: 100 L10

*** أمر الكتابة write**

يكتب علي القرص الملف الذي يجري التعامل معه وصيغته

W [address (drive:record record)]

إذا كـتب الأمر ومـعـه معـامل العنوان فـقط فـان الملف سـوف يكتب بدءا من هذا العنوان، ويجب تسمية الملف باستخدام أمر التسمية قبل استخدام أمر الكتابة .

اذا 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 التـصحيح debug أربعة أوامر تـبدأبحرف X للتعـامل 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 Expande 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 ooo3 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 :

-XM 5 2 0003

واذا 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 الاشارة 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 ، وللاطلاع 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 البايت 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 والاخراج BIOS.

بين هذه البايتات 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 المرئى ، 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 فقطROM. وهذا الجزء هو مكان أساسيات الادخال والاخراج في الحاسب وفيه يمكن قراءة حقوق طبع البرنامج .

تغيير الذاكرة

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

لتغيير الذاكرة فإن مكان ذاكرة العرض المرئى تعد أفضل مكان لبداية التجارب ،فإن أى تغيير يحدث فيها يظهر على شاشة وحدة العرض المرئى 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 فى تحديد قدر المساحة ليكون ٤٠٠٠ بايت هو أن :

العنوان b800:0000 (أو b000:0000 للحاسب ذى الشاشة أحادية اللون) هو بداية ذاكرة العرض المرئى ، ولما كانت شاشة العرض المرئى 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 فى الذاكرة ، فالبايت الأول هو قيمة الحرف فى شفرة الترميز ascii وهو الحرف الذى سيتم عرضه ، والبايت الثانى هو سمة الحرف على الشاشة (لضبط اللون وكثافته وومیضه و . . .) .

لذلك فان شاشة النص ذات الثمانين عمودا تساوى (٤٠٠٠ بايت) ، وهى التى لها القيمة fa0 بايت فى 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 التي 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 علي الرمز ce 21 متكررا الواحد بعد الآخر .

فحص الذاكرة الموسعة

اذا 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 هى قيم حقيقية ، وفى 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 فى 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 فى 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 فى الحاسب ، كما يعطى أيضا بيانات بالكمية المتوفرة التى 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 التى يمكن استخدامها من ذاكرة الحاسب ؟

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

معاملات أمر استعراض الذاكرة

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

خيارات الاصدار الخامس program, debug, classify

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

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

يعطى الخيار classify / قائمة بأسماء وأحجام كل البرامج الموجودة فى 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 (فى الاصدار الخامس من نظام تشغيل القرص) .

كمثال لذلك يكتب الأمر على الصورة التالية :

c:\> mem / program (¦) more

بعد الضغط على مفتاح الادخال تظهر الشاشة الأولى من المعلومات التى تعرض البرامج وسواقات الأجهزة الموجودة فى الذاكرة .

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

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

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

c:\>mem / classify (¦) more

فى العرض تتضح البرامج الموجودة فى الذاكرة التقليدية والمساحة التى يحتلها كل برنامج .

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

برنامج MSDOS الذى يحتل مساحة 4.8 كيلو بايت .

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

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

والبرنامج المقيم فى الذاكرة DOSKEY مستخدما مساحة قدرها كيلو بايت .

مجموعة المساحات الفارغة (FREE) هو فى هذا المثال مساحة قدرها 556.3 كيلو بايت وكلها متوفرة لتطبيقات نظام تشغيل القرص DOS .

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

أمر استعراض معلومات الذاكرة mem يبين كمية الذاكرة التقـليدية والموسعة والممتدة الموجودة فى الحاسب ، ومقدار الكمية المتـوفرة 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 الأمـر أيضا باعطاء بيان هام عـن توافر بندين آخرين فى الحـاسب (اذا كانا موجـودين) ، وهما مسـاحة الذاكرة العـالية (HMA) ، ومجـموعـات الذاكرة العلـيا (UMB).

مساحة الذاكرة العالية HIMEM هى المساحـة الأولى المؤلفة من ٦٤ كـيلو 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 تشغـيل القرص فى اصداراته الجـديدة يصبح بامكان نظام تشغـيل القرص الوصول الى مساحـة الذاكرة العالية hma ، ويمكن بالتـالى نقل جـزء من نظام تشغـيل القرص (البرنامج MSDOS) الى هذه المسـاحة hma محـررا بذلك مساحة جـيدة من الذاكرة التقليدية .

إذا كان الحاسب يحـتوى على المعالج 80286,80386 أو أعلى مع ذاكـرة ممتدة وتم تركيب مـدير الذاكرة الممتـدة فإن السطر الأخيـر من الأمر mem سيكون 64kb high memory area available .

من الطبيـعى أن خرج أمر اسـتعراض بيـانات الذاكرة سوف يخـتلف من حاسب الى آخر، كما سوف يختلف اذا كان نظام تشغـيل القرص قد جرى نقله الى المساحة الذاكرة العالية hma ، واذا كان هناك شئ آخـر يستعمل المسـاحة hma ستظهـر رسالة تبين أن مساحة الذاكرة العالية قيد الاستخدام .

عند انشاء مجمـوعات الذاكرة العليا umb لتخزين برامج سواقـات الأجهزة والبرامج المقـيمـة فى الذاكرة سـوف يعطى خرج الأمـر mem/ classify قائمـة معلومـات عن مجموعات الذاكرة العليا umb وكمية الذاكرة الفارغة المتوفـرة ومجموعة الذاكرة المتوفرة للبرامج .

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

معاملات أمر استعـراض الذاكرة فى الاصدار السادس من نظام تشغيل القرص .

اخـتلفت المعامـلات الموجودة مع أمـر اسـتعـراض معلومـات الذاكرة فى الاصـدار السادس DOS 6 من نظام تشغيل القرص عنها فى المعاملات التى كانت موجودة مع نفس الأمر فى الاصدار الخامس قليلا .

لوحظ أن المستخـدم يضطر الى استخدام أمـر الترشيح more دائما عند تنفـيذ الأمر وذلك بسبب الكم الكبير من المعلومات التى ترد الى الشاشة بسرعة لايستطيع ملاحقتها.

لذلك أضيف خيار الصفحة الجديد PAGE/ بحيث يتوقف العـرض على الشاشة بعد امتلاء الصـفحة ، واستتـبع ذلك استبعاد خـيار البرنامج PROGRAM/ من الخـيارات المستخدمة مع أمر استعراض الذاكرة .

تسهـيلا لمعرفة كـمية الذاكرة الخـالية مباشـرة فى كل من الذاكرة التقليـدية والذاكرة العليا، فقد أضيف خيار جديد ليعطى بيـانا سريعا وموجزا عن المساحات الفارغة فى كل من الذاكرتين ، وهو الخيار (حر) free .

بقى خيار debug كمـا هو وان كانت امكانيـات قد زادت فى قيـامه بعرض مـقاطع الذاكرة وبيـانات المشغلات الداخلية ومـعلومات أخرى عن توزيع البـرامج على المقاطع المختلفة من الذاكرة .

كما بقى 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 خيارMODULE وهو الذى يكتب اختصارا M/ ويتبعه كتابة نقطتين رأسيتين (:) colon بعد الخيار يليها رقم يحدد القطاع المراد استعراض تفاصيله .

مثال لأمر MEM/C في الاصدار DOS 6

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)
smartorv	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 blaock 44448 (43)
MS -DOS is reident in the high memory area

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

```
device = c:\ himem, sys
dos - high, umb
device = c: \ dos \ emm 386, noms
files = 20
bufeers = 20
shell = c: \ dos \ command, com c: \ dos \ /p
stacks = 0,5
devicehigh = c: \ dos \ mouse, sys
devicehigh = c: \ dos \ gmouse, sys
```

موجز

* هناك أداتا تسمحان بمعاينة الذاكرة هما خدمات أمر التصحيح debug ، وأمر استعراض معلومات الذاكرة mem .

* يمكن اعتبار الأمر mem من الأوامر البسيطة التى يستعملها المستخدم العادى، أما برنامج debug فهو واحدة من الأدوات التى 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 أخري 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 الاختيارى debug/ مع الأمر mem البرامج المحملة فى الذاكـرة وكمية الذاكرة التى تحتلها وتفـيذ هذه المعلومات فى نقل سواقات الأجهزة والبـرامج المقيمة فى الذاكرة الى مجموعات الذاكرة العليا .

* فى اطار تعديل نظـام تشغيل القـرص فى اصداره السادس فـقد أضيـفت الى أمر استعراض الذاكرة MEM خيـارات جديدة توسع من امكانياتـه مع زيادة امكانيات اخيارات الموجودة فيه مسبقا .

* أضيف خيار الصفحة PAGE/ ليتوقف العرض على الشاشة بعد امتلاء الصفحة .

* استبعد خيار البرنامج PROGRAM/ من الخيارات المستخـدمة مع أمر استعراض الذاكرة .

أضيف خـيار (حر) free ليعـطى بيانا سـريعا ومـوجزا عن المسـاحات الفارغـة فى الذاكرة.

بقى خيار debug كمـا هو وان كانت امكانياته قـد زادت ، كما بقى خـيار التقـسيم classify كما هو أيضا .

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