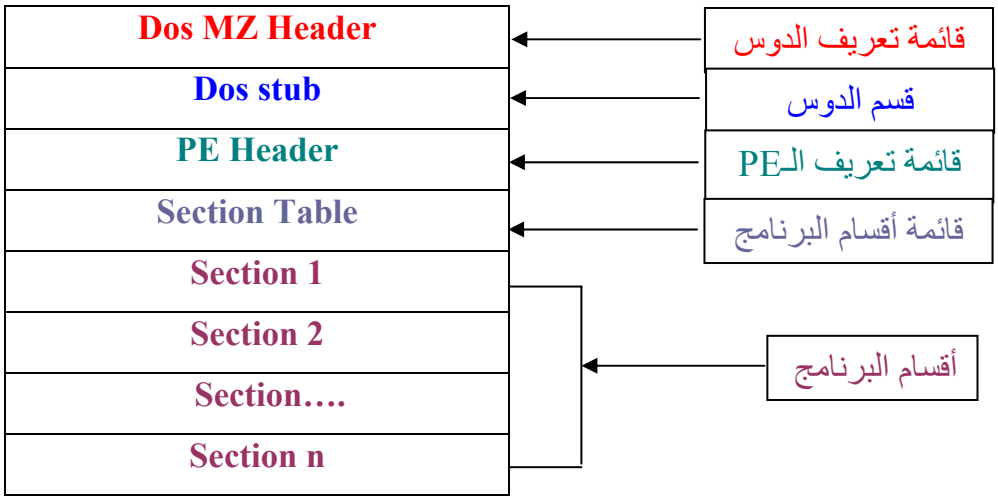
شرح الـ PE

الـPE هي إختصار (Portable Executable) وهي الملفات التي تعمل في بيئة WIN32 مثل (DLL) و (EXE) و (OCX) ولكن الملفات الأكثر إستخداماً التي سوف نتعرض لها هي (EXE) و (DLL) ومعرفة الـPE تُمكنا من إضافة كود لملف ما أو إضافة وظيفة ما. كما تمكنا من فك ضغط أو تشفير الملفات التنفيذية بطريقة يدوية لأننا في أيامنا هذه لا يوجد برنامج إلا ومشفر أو مضغوط ليتم حمايته من قبل قراصنة البرامج وفكرة التشفير أو الضغط تعتمد هي تدمير الدوال المستوردة ألا وهي دوال الـAPI وكذلك نصوص البرنامج حتي لا يتم إيجادها بينما تعمل برامج الضغط أو التشفير علي فك تشفير البرنامج في الذاكرة لكي يتم القفز إلي نقطة البداية الصحيحة (**O**riginal **E**ntry **P**oint) ولو إستطعنا عمل Dump أي إستخلاص الملف من التنفيذي من التشفير أو الضغط عندئذ يمكن إصلاح دوال الـAPI أو إصلاح أقسام البرنامج إذا كانت تحتاج لتصليح والأن كيف يتم عمل كل ذلك بدون معرفة المعلومات اللازمة عن صيغة الـPE.

الشكل التالي يوضح البنية الأساسية للملف في صيغة الـPE كما يلي :-



علي الأقل لابد من وجود قسمين في ملفات الـPE ولكن في كثير من البرامج قد يصل عدد الأقسام إلي ٨ أقسام.

The DOS Header

كل ملفات الـPE تبدأ بالـDOS Header الذي يشغل مساحة 64 بايت من الملف وفي هذه الحالة يتم تشغيل البرنامج من الدوس فإذا كان ملائم سوف يتم نداء المقاطعات الخاصة بالدوس وهي 21h أما إذا كان البرنامج يعمل تحت بيئة WIN32 عندئذ سوف يتم نداء المقاطعة 21h ثم الخدمة رقم 9 وهي الخاصة بطبع النصوص مثل "This Program must be run under Microsoft windows" وغالبية البرامج حالياً تعمل علي بيئة WIN32 والـDOS Header له بنية خاصة به وهذه البنية مُعرّفة في ملف "Windows.inc" بالنسبة لمستخدمي لغة الأسمبلي أو في ملف "winnt.h" بالنسبة لمستخدمي لغة الـC وهذه البنية تتكون من ١٩ دالة كما في الشكل التالي :

```
IMAGE_DOS_HEADER STRUCT
  e_magic        WORD      ?
  e_cblp         WORD      ?
  e_cp           WORD      ?
  e_crlc         WORD      ?
  e_cparhdr      WORD      ?
  e_minalloc     WORD      ?
  e_maxalloc     WORD      ?
  e_ss           WORD      ?
  e_sp           WORD      ?
  e_csum         WORD      ?
  e_ip           WORD      ?
  e_cs           WORD      ?
  e_lfarlc       WORD      ?
  e_ovno         WORD      ?
  e_res          WORD   4 dup(?)
  e_oemid        WORD      ?
  e_oeminfo      WORD      ?
  e_res2         WORD  10 dup(?)
  e_lfanew       DWORD     ?
IMAGE_DOS_HEADER ENDS
```

في الشكل السابق نجد الدالة e_magic معرفة في ملف الـPE وهي تمثل أو بايتين وسوف تجد برنامج تحت إسم "Example.exe" مرفق مع هذا الشرح وهذا البرنامج

سوف نُجري علية جميع إختباراتنا ولو قمت بفتح هذا البرنامج ببرنامج Hex Workshop سوف تري الشكل التالي :

```
00000000 4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00 MZP.............
00000010 B8 00 00 00 00 00 00 00 40 00 1A 00 00 00 00 00 ........@.......
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 ................
00000040 BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90 ........!..L.!..
```

كما تري في الشكل السابق فإن أول بايتين يمثلون الحروف MZ وهذه الحروف هي إختصار لإسم Mark Zbikowsky وهو واحد من أفضل المصممين لـ MS-DOS والشكل التالي سوف يوضح قسم الـThe DOS Header مع قسم الـDOS Stub كما يلي :

```
00000000 4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00 MZP.
00000010 B8 00 00 00 00 00 00 00 40 00 1A 00 00 00 00 00
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
00000040 BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90
00000050 54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73 This program mus
00000060 74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57 t be run under W
00000070 69 6E 33 32 0D 0A 24 37 00 00 00 00 00 00 00 00 in32..$7........
00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000100 50 45 00 00 4C 01 08 00 19 5E 42 2A 00 00 00 00 PE..L....^B*....
00000110 00 00 00 00 E0 00 8E 81 0B 01 02 19 00 80 05 00 ................
```

The DOS Header

هذه البايتات هي الخاصة بالدالة e_ifanew وإذا عكسنا هذه البايتات سوف تصبح 00000100 وهي الخاصة بهذا العنوان

DOS Stub

كما تري في الشكل السابق فإن الدالة e_ifanew قيمتها 00 00 01 00 ولكن نحن عرفنا أن الـMemory تعمل علي عكس هذه القيم فالعكس يتم عن طريق وضع البايت الأيمن في الأول ثم يلي البايت الثاني بعد العكس سوف تصبح القيمة 00000100 وهذه القيمة الخاصة بالـOffset الموضح في الشكل السابق وهو الـOffset الخاص ببداية الـPE Header وبهذه الدالة يتم نهاية قسم الـ DOS Header ثم ندخل في الـDOS Stub (قسم الدوس) ولو رأيت في الشكل السابق سوف تري نص يخبرك بأن هذا البرنامج يجب أن يعمل تحت بيئة WIN32 وبذلك يتم تخطي هذا القسم للدخول في قسم الـPE Header.

The PE Header

الـPE Header يمثل الجزء الرئيسي من بينة الملف ويمثل أيضاً 248 بايت وهي موضحة كما في الشكل التالي :

```
000000F0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
00000100 | 50 45 00 00 4C 01 08 00 19 5E 42 2A 00 00 00 00 | PE..L....^B*....
00000110 | 00 00 00 00 E0 00 8E 81 0B 01 02 19 00 80 05 00 | ................
00000120 | 00 D2 00 00 00 00 00 00 84 8F 05 00 00 10 00 00 | ................
00000130 | 00 90 05 00 00 00 40 00 00 10 00 00 00 02 00 00 | ......@.........
00000140 | 04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 | ................
00000150 | 00 BA 06 00 00 04 00 00 00 00 00 00 02 00 00 00 | ................
00000160 | 00 00 10 00 00 40 00 00 00 00 10 00 00 10 00 00 | .....@..........
00000170 | 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 | ................
00000180 | 00 C0 05 00 BC 21 00 00 00 80 06 00 00 3A 00 00 | .....!.......:..
00000190 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
000001A0 | 00 10 06 00 08 61 00 00 00 00 00 00 00 00 00 00 | .....a..........
000001B0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
000001C0 | 00 00 06 00 18 00 00 00 00 00 00 00 00 00 00 00 | ................
000001D0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
000001E0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
000001F0 | 00 00 00 00 00 00 00 00 43 4F 44 45 00 00 00 00 | ........CODE....
```

وهذا الجزء له بنية خاصة به وهي IMAGE_NT_HEADERS وتحتوي علي 3 عناصر وهذه البنية مُعرّفه كما يلي :

```
IMAGE_NT_HEADERS STRUCT
  Signature          DWORD                  ?
  FileHeader         IMAGE_FILE_HEADER      <>
  OptionalHeader     IMAGE_OPTIONAL_HEADER32 <>
IMAGE_NT_HEADERS ENDS
```

Signature : يمثل الـ4 بايت الأولي وهو يشير إلي القيم 50h , 45h , 00h , 00h وهذه القيم الخاصة ببداية الـPE Header.

FileHeader : يمثل الـ20 بايت بعد الـSignature ويشمل معلومات حول خصائص الملف مثل عدد الأقسام، تاريخ إنشاء الملف، نوع الجهاز وهكذا.والعنصر FileHeader له بنية خاصة به وهي مُعرّفة كما يلي :

```
IMAGE_FILE_HEADER STRUCT
  Machine                 WORD    ?
  NumberOfSections        WORD    ?
  TimeDateStamp           DWORD   ?
  PointerToSymbolTable    DWORD   ?
  NumberOfSymbols         DWORD   ?
  SizeOfOptionalHeader    WORD    ?
  Characteristics         WORD    ?
IMAGE_FILE_HEADER ENDS
```

Machine : هذه الدالة تدل علي نوعية الـCPU المستخدمة ونحن نستعمل Intel 386 وهذاهو أغلب الإستخدام لنا وقيمتها هي (14Ch) وإليك الشكل التالي :

```
000000C0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
000000D0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
000000E0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
000000F0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
00000100 | 50 45 00 00 4C 01 08 00 19 5E 42 2A 00 00 00 00 | PE..L....^B*....
00000110 | 00 00 00 00 E0 00 8E 81 0B 01 02 19 00 80 05 00 | ................
00000120 | 00 D2 00 00 00 00 00 00 84 8F 05 00 00 10 00 00 | ................
00000130 | 00 90 05 00 00 00 40 00 00 10 00 00 00 02 00 00 | ......@.........
00000140 | 04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 | ................
```

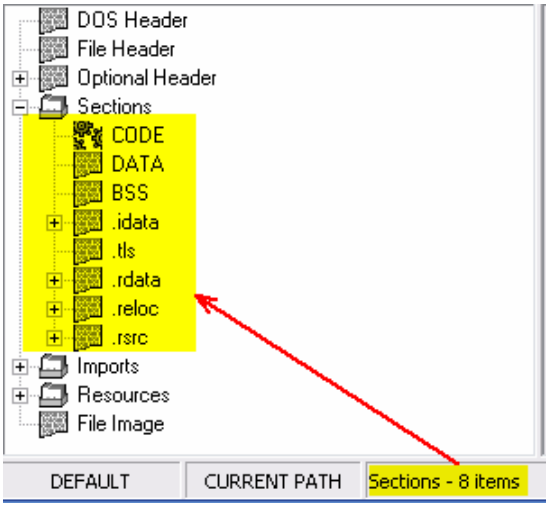
كما تري فإن القيمة 14C تشير إلي Intel 80386 Processor ، أما لو كانت القيمة 14D كانت عندئذ سوف تشير إلي Intel 80486 Processor.

NumberOfSections : وهذه الدالة خاصة بعدد أقسام البرنامج إذا أردنا إضافة أو حذف قسم من البرنامج فسوف نعدل هذه القيمة علي حسب ما نريد والشكل التالي يوضح عدد أقسام البرنامج كما يلي :

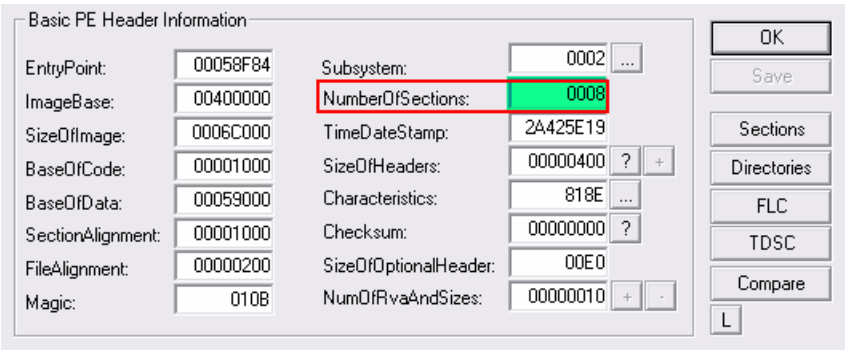
```
00000000 4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00 MZP.............
00000010 B8 00 00 00 00 00 00 00 40 00 1A 00 00 00 00 00 ........@.......
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 ................
00000040 BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90 ........!..L.!..
00000050 54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73 This program mus
00000060 74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57 t be run under W
00000070 69 6E 33 32 0D 0A 24 37 00 00 00 00 00 00 00 00 in32..$7........
00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (
000000B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (
000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000000F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000100 50 45 00 00 4C 01 08 00 19 5E 42 2A 00 00 00 00 PE..L....^B*....
00000110 00 00 00 00 E0 00 8E 81 0B 01 02 19 00 80 05 00 ................
```

عند عكسها سوف تصبح 08 00
والمقصود بـ8 عدد الأقسام

والرقم السابق يدل علي أن عدد الأقسام 8 ويمكن أن نتأكد من هذه القيمة عن طريق برنامج PEBrowsePro كما في الشكل التالي :



وأيضاً عن طريق برنامج LordPE كما في الشكل التالي :



TimeDateStamp : هذه الدالة تدل علي وقت وتاريخ إنشاء الملف.

PointerToSymbolTabel و NumberOfSymbols : هـــذين العنـــصرين يستخدموا لـDebug information وليس لهم أهمية بالنسبة لنا.

SizeOfOptionalHeader : حجم الـOptionalHeader كما في الشكل التالي :

```
--------|-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --|................
000000E0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|................
000000F0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|................
00000100|50 45 00 00 4C 01 08 00 19 5E 42 2A 00 00 00 00|PE..L....^B*....
00000110|00 00 00 00 E0 00 8E 81 0B 01 02 19 00 80 05 00|................
00000120|00 D2 00 00 00 00 00 00 84 8F 05 00 00 10 00 00|................
00000130|00 90 05 00 00 00 40 00 00 10 00 00 00 02 00 00|......@.........
00000140|04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00|................
```

كما تري فلو عسنا هذه القيمة لتأخذ وضعها الطبيعي سوف تصبح 00E0 وهي تساوي في الـDecimal 224 وهو فعلاً حجم الـOptionalHeader.

Characteristics : هذه الدالة تشير إلي خصائص الملف من حيث كون الملف EXE أو DLL وإليك الشكل التالي للتوضيح :

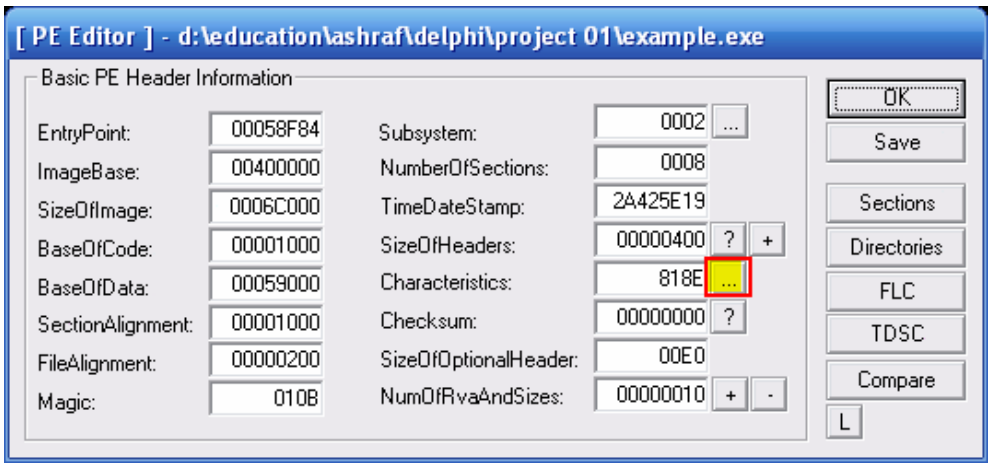
```
000000D0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|................
000000E0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|................
000000F0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|................
00000100|50 45 00 00 4C 01 08 00 19 5E 42 2A 00 00 00 00|PE..L....^B*....
00000110|00 00 00 00 E0 00 8E 81 0B 01 02 19 00 80 05 00|................
00000120|00 D2 00 00 00 00 00 00 84 8F 05 00 00 10 00 00|................
00000130|00 90 05 00 00 00 40 00 00 10 00 00 00 02 00 00|......@.........
00000140|04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00|................
00000150|00 C0 06 00 00 04 00 00 00 00 00 00 02 00 00 00|................
```

كما تري في الشكل السابق فلو عكسنا القيمة 8E81 فسوف تصبح 818E وهذه هي الخاصية الخاصة بهذا الملف.

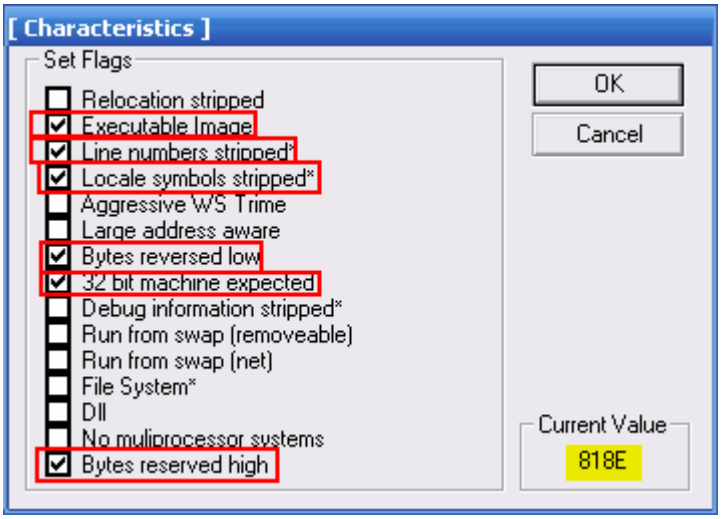
وهذا الجدول يوضح خصائص الملف :

الخاصية	الوصف
0001	Relocation info stripped from file
0002	File is executable
0004	Line numbers stripped from file
0008	Local symbols striooed from file
0010	Lets OS aggressively trim working set
0020	App can handle >2Gb addresses
0080	Low bytes of machine word are reversed
0100	requires 32-bit WORD machine
0200	Debugging info stripped from file into .DBG file
0400	If image is on removable media, copy and run from swap file
0800	If image is on a network, copy and run from swap file
1000	System file
2000	File is DLL
4000	File should only be run on a single-processor machine
8000	High bytes of machine word are reversed

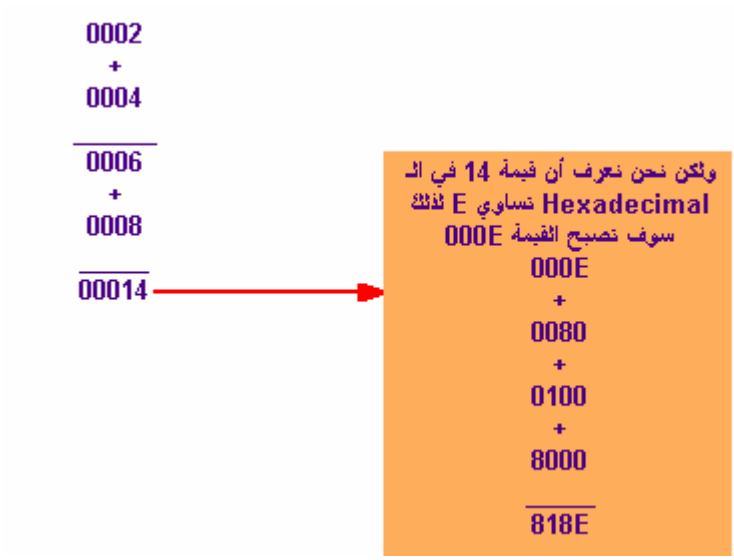
ولو قمت بتشغيل برنامج LordPE ثم ضغطت علي زر PE Header ثم إخترت البرنامج المثال سوف تري هذا الشكل :



إضغط علي الزر المشار إليه في الشكل السابق سوف تري هذا الشكل :



كما تري فإن القيمة 818E هي نتاج الخصائص المشار إليها في الشكل السابق فلو رأينا القيم المذكورة في الجدول السابق فسوف نجد أن 0002 + 0004 + 0008 + 0080 + 00100 + 8000 = 818E وسوف أوضح كيفية جمع هذه القيم في الشكل التالي :



OptionalHeader : تمثـل الــ224 بايـت بعـد الــFileHeader ويـشمل علــي معلومات خاصة بملف الـPE مثل حجم الكود وحجم البيانـات ونقطـة الـدخول الأصـلية للبرنـامج وهذا العنصر له بنية خاصة به مُعرّفة كما في الشكل التالي :

```
IMAGE_OPTIONAL_HEADER32 STRUCT
  Magic                        WORD      ?
  MajorLinkerVersion           BYTE      ?
  MinorLinkerVersion           BYTE      ?
  SizeOfCode                   DWORD     ?
  SizeOfInitializedData        DWORD     ?
  SizeOfUninitializedData      DWORD     ?
  AddressOfEntryPoint          DWORD     ?
  BaseOfCode                   DWORD     ?
  BaseOfData                   DWORD     ?
  ImageBase                    DWORD     ?
  SectionAlignment             DWORD     ?
  FileAlignment                DWORD     ?
  MajorOperatingSystemVersion  WORD      ?
  MinorOperatingSystemVersion  WORD      ?
  MajorImageVersion            WORD      ?
  MinorImageVersion            WORD      ?
  MajorSubsystemVersion        WORD      ?
  MinorSubsystemVersion        WORD      ?
  Win32VersionValue            DWORD     ?
  SizeOfImage                  DWORD     ?
  SizeOfHeaders                DWORD     ?
  CheckSum                     DWORD     ?
  Subsystem                    WORD      ?
  DllCharacteristics           WORD      ?
  SizeOfStackReserve           DWORD     ?
  SizeOfStackCommit            DWORD     ?
  SizeOfHeapReserve            DWORD     ?
  SizeOfHeapCommit             DWORD     ?
  LoaderFlags                  DWORD     ?
  NumberOfRvaAndSizes          DWORD     ?
  DataDirectory                IMAGE_DATA_DIRECTORY
IMAGE_NUMBEROF_DIRECTORY_ENTRIES dup(<>)
IMAGE_OPTIONAL_HEADER32 ENDS
```

سوف أذكر أهم الدوال بالنسبة لنا :-

AddressOfEntryPoint : هذه الدالة تحدد نقطة إنطلاقة البرنامج وهو أول أمر سوف يتم تنفيذه وهذه النقطة هي عبارة عن عنوان في الـMemory ويتم الإشارة لها بـRelative Virtual Address (RVA) ويبدأ البرنامج التشغيل بداية من هذا العنوان.

ImageBase : تشير هذه الدالة إلي عنوان بداية البرنامج في الـMemory وهذا العنوان هو 400000h وهذه القيمة ثابته في ٩٩% من البرامج.

SectionAlignment : هذه الدالة تشير إلي تنسيق محتويات أقسام البرنامج في الـMemory وفي الأغلب تكون قيمة هذه الدالة (1000h) وعلي سبيل المثال إذا كانت قيمة هذه الدالة 4096 (1000h) عندئذ كل قسم يجب أن يبدأ في مضاعفات العدد 4096 فإذا كان القسم الأول يبدأ عند 401000h عندئذ القسم الثاني سوف يكون 402000h.

FileAlignment : هذه الدالة تشير إلي تنسيق محتويات البرنامج علي القرص الصلب وفي الأغلب تكون قيمة هذه الدالة (200h) وعلي سبيل المثال إذا كانت قيمة هذه الدالة 512 (200h) عندئذ كل قسم يجب أن يبدأ في مضاعفات العدد 512 فإذا كان الـOffset للقسم الأول يساوي 200h عندئذ القسم الثاني يجب أن يحدد عند الـOffset 400h.

SizeOfImage : هو حجم البرنامج ككل في الـMemory.

SizeOfHeaders : حجم كل الـHeaders بالإضافة إلي قائمة أقسام البرنامج وهذه القيمة مساوية لحجم الملف ناقص حجم أقسام البرنامج.

DataDirectory : عبارة عن مصفوفة تتكون من 16 عنصر ولها بنية خاصة بها وهي IMAGE_DATA_DIRECTORY وهي تمثل 128 بايت من

الـOptionalHeader والـDataDirectory لها بنية خاصة بها وهي مُعرَّفة كما في الشكل التالي :

```
IMAGE_DATA_DIRECTORY STRUCT
  VirtualAddress    DWORD     ?
  isize             DWORD     ?
IMAGE_DATA_DIRECTORY ENDS
```

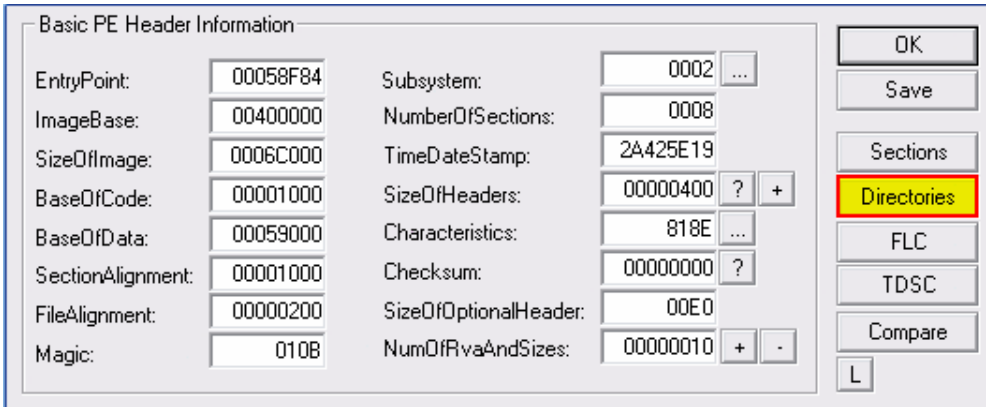
VirtualAddress : الـRVA يُنسب إليه فهو يساوي RVA + ImageBase.

isize : يشمل حجم بيانات كل عنصر من العناصر الـ16 ولكن بالبايتات.

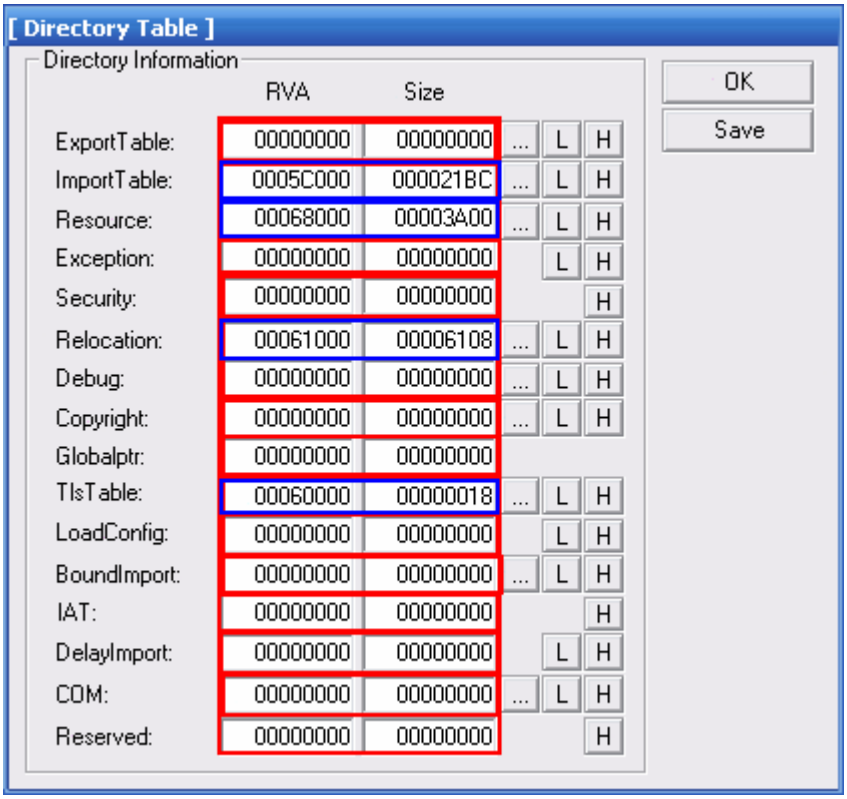
كما ذكرنا من قبل فإن هذه المصفوفة تحتوي علي 16 عنصر كما في الشكل التالي :

```
IMAGE_DIRECTORY_ENTRY_EXPORT              equ   0
IMAGE_DIRECTORY_ENTRY_IMPORT              equ   1
IMAGE_DIRECTORY_ENTRY_RESOURCE            equ   2
IMAGE_DIRECTORY_ENTRY_EXCEPTION           equ   3
IMAGE_DIRECTORY_ENTRY_SECURITY            equ   4
IMAGE_DIRECTORY_ENTRY_BASERELOC           equ   5
IMAGE_DIRECTORY_ENTRY_DEBUG               equ   6
IMAGE_DIRECTORY_ENTRY_COPYRIGHT           equ   7
IMAGE_DIRECTORY_ENTRY_GLOBALPTR           equ   8
IMAGE_DIRECTORY_ENTRY_TLS                 equ   9
IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG         equ   10
IMAGE_DIRECTORY_ENTRY_BOUND_IMPORT        equ   11
IMAGE_DIRECTORY_ENTRY_IAT                 equ   12
IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT        equ   13
IMAGE_DIRECTORY_ENTRY_COM_DESCRIPTOR      equ   14
IMAGE_NUMBEROF_DIRECTORY_ENTRIES          equ   16
```

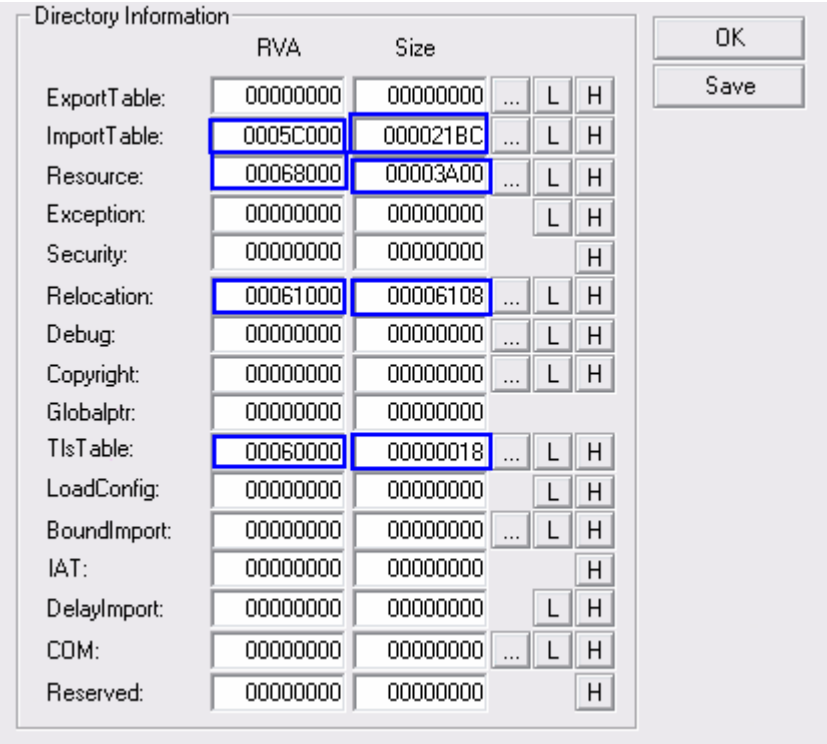
الشكل السابق يوضح الـ16 عنصر عندئذ قم بتشغيل برنامج LordPE ثم إضغط علي زر PE Header ثم إختر المثال "Example.exe" لتري الشكل التالي :



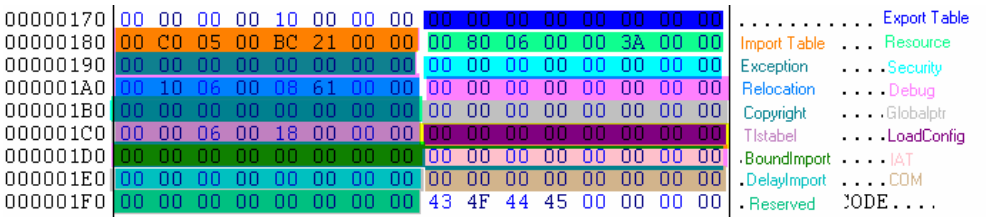
إضغط علي الزر المشار إليه في الصورة السابقة لتري هذا الشكل :



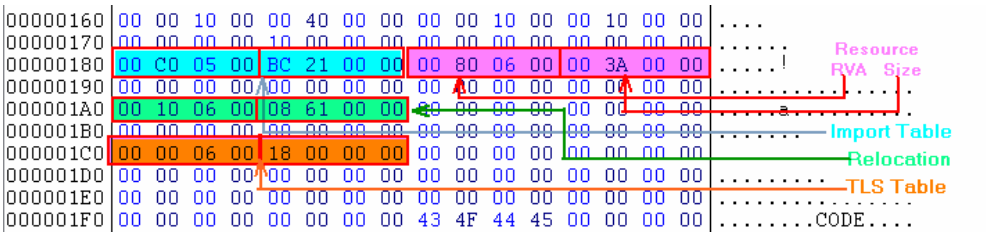
كما تري في الشكل السابق الـ16 عنصر ولكن توجد أربع عناصر منهم لهم قيم والأخرون ليست لهم قيم كما في الشكل التالي :



ويتم توضيحهم عن طريق برنامج Hexworkshop كما في الشكل التالي :



وإذا لا حظت سوف تجد الـRVA والـSize الخاص بكل عنصر ولكن توجد 4 عناصر لهم قيم وهم موضحين كما في الشكل التالي :



ملحوظة : الـRVA هو عبارة عن VirtualAddress - ImageBase