الشكل التالي يوضح الأقسام التي رأيناها من قبل 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 ....
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 ....
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....
00000200 CC 7F 05 00 00 10 00 00 00 80 05 00 00 04 00 00 ................
00000210 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60 ............ ..
00000220 44 41 54 41 00 00 00 00 1C 11 00 00 00 90 05 00 DATA............
```

Dos Header
64 Bytes

Dos Stub
192 Bytes

PE Header
Signature (4 Byte)
File Header (20 Byte)
Optional Header (224

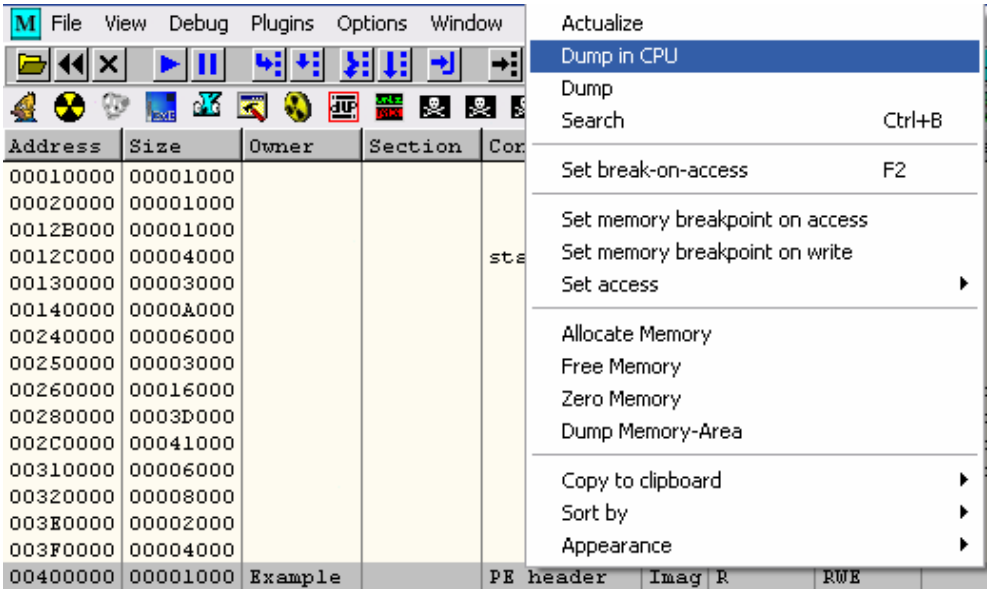
Data
Directory
128 Byte

Section
Tabels

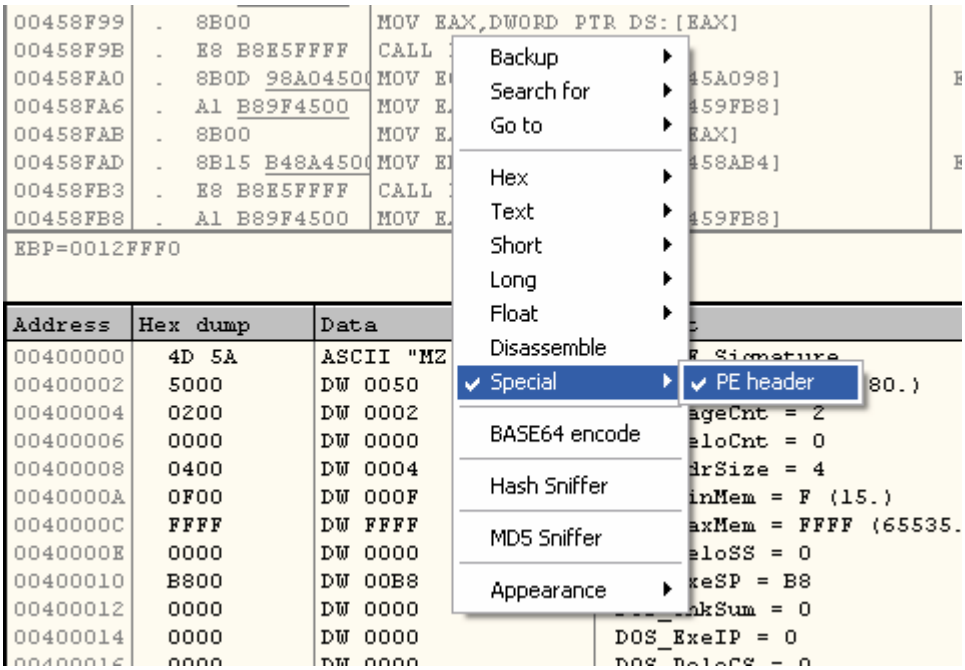
فإذا قمنا بتشغيل برنامج ollydbg ثم قمنا بتحميل الملف "Example.exe" عندئذ إضغط ALT+M للذهاب إلي صفحة الـMemory لتري الشكل التالي :

00400000	00001000	Example		PE header	Imag 01001002	R	RWE
00401000	00058000	Example	CODE	code	Imag 01001002	R	RWE
00459000	00002000	Example	DATA	data	Imag 01001002	R	RWE
0045B000	00001000	Example	BSS		Imag 01001002	R	RWE
0045C000	00003000	Example	.idata	imports	Imag 01001002	R	RWE
0045F000	00001000	Example	.tls		Imag 01001002	R	RWE
00460000	00001000	Example	.rdata		Imag 01001002	R	RWE
00461000	00007000	Example	.reloc	relocations	Imag 01001002	R	RWE
00468000	00004000	Example	.rsrc	resources	Imag 01001002	R	RWE

إضغط علي السطر المشار إليه في الشكل السابق Right Click ثم إختر هذا الإختيار :



ثم قف علي النافذة الخاصة بنافذة البيانات ثم إضغط عليها Right Click ثم إختر هذا الإختيار:



بعد إختيار الـheader PE سوف تري المعلومات الخاصة بالـPE كما في الشكل التالي :

Address	Hex dump	Data	Comment
00400100	50 45 00 0	ASCII "PE"	PE signature (PE)
00400104	4C01	DW 014C	Machine = IMAGE_FILE_MACHINE_I386
00400106	0800	DW 0008	NumberOfSections = 8
00400108	195E422A	DD 2A425E19	TimeDateStamp = 2A425E19
0040010C	00000000	DD 00000000	PointerToSymbolTable = 0
00400110	00000000	DD 00000000	NumberOfSymbols = 0
00400114	E000	DW 00E0	SizeOfOptionalHeader = E0 (224.)
00400116	8E81	DW 818E	Characteristics = EXECUTABLE_IMAGE|32BIT_MACHINE|LI
00400118	0B01	DW 010B	MagicNumber = PE32
0040011A	02	DB 02	MajorLinkerVersion = 2
0040011B	19	DB 19	MinorLinkerVersion = 19 (25.)
0040011C	00800500	DD 00058000	SizeOfCode = 58000 (360448.)
00400120	00D20000	DD 0000D200	SizeOfInitializedData = D200 (53760.)
00400124	00000000	DD 00000000	SizeOfUninitializedData = 0
00400128	848F0500	DD 00058F84	AddressOfEntryPoint = 58F84
0040012C	00100000	DD 00001000	BaseOfCode = 1000
00400130	00900500	DD 00059000	BaseOfData = 59000
00400134	00004000	DD 00400000	ImageBase = 400000
00400138	00100000	DD 00001000	SectionAlignment = 1000
0040013C	00020000	DD 00000200	FileAlignment = 200
00400140	0400	DW 0004	MajorOSVersion = 4
00400142	0000	DW 0000	MinorOSVersion = 0
00400144	0000	DW 0000	MajorImageVersion = 0
00400146	0000	DW 0000	MinorImageVersion = 0
00400148	0400	DW 0004	MajorSubsystemVersion = 4
0040014A	0000	DW 0000	MinorSubsystemVersion = 0
0040014C	00000000	DD 00000000	Reserved
00400150	00C00600	DD 0006C000	SizeOfImage = 6C000 (442368.)
00400154	00040000	DD 00000400	SizeOfHeaders = 400 (1024.)
00400158	00000000	DD 00000000	CheckSum = 0
0040015C	0200	DW 0002	Subsystem = IMAGE_SUBSYSTEM_WINDOWS_GUI
0040015E	0000	DW 0000	DLLCharacteristics = 0
00400160	00001000	DD 00100000	SizeOfStackReserve = 100000 (1048576.)
00400164	00400000	DD 00004000	SizeOfStackCommit = 4000 (16384.)
00400168	00001000	DD 00100000	SizeOfHeapReserve = 100000 (1048576.)
0040016C	00100000	DD 00001000	SizeOfHeapCommit = 1000 (4096.)
00400170	00000000	DD 00000000	LoaderFlags = 0
00400174	10000000	DD 00000010	NumberOfRvaAndSizes = 10 (16.)
00400178	00000000	DD 00000000	Export Table address = 0
0040017C	00000000	DD 00000000	Export Table size = 0
00400180	00C00500	DD 0005C000	Import Table address = 5C000
00400184	BC210000	DD 000021BC	Import Table size = 21BC (8636.)
00400188	00800600	DD 00068000	Resource Table address = 68000

وبذلك نكون قد إنتهينا من شرح الـHeader PE وسوف ندخل في الجزء التالي الخاص بـThe Section Table.

The Section Table

نري هذا الجزء مباشرة بعد الـPE Header وهو لـه بنية خاصة وهذا الجزء يمثل قائمة اقسام البرنامج ويحتوي هذا الجزء علي معلومات عن كل قسم في ملف الـPE وكما ذكرنا سابقاً أن العنصر الثاني من الـFileHeader يحتوي علي دالة تفيد عدد أقسام البرنامج وإذا كان ملف الـPE يحتوي علي 8 أقسام مثلاً فسوف تجد 8 نسخ من هذه البنية IMAGE_SECTION_HEADER وهي مُعرّفة كما يلي :

```
IMAGE_SECTION_HEADER STRUCT
    Name1 db IMAGE_SIZEOF_SHORT_NAME dup(?)
    union Misc
        PhysicalAddress dd  ?
        VirtualSize dd      ?
    ends
    VirtualAddress dd       ?
    SizeOfRawData dd        ?
    PointerToRawData dd     ?
    PointerToRelocations dd ?
    PointerToLinenumbers dd ?
    NumberOfRelocations dw  ?
    NumberOfLinenumbers dw  ?
    Characteristics dd      ?
IMAGE_SECTION_HEADER ENDS
```

Name1 **:** هذه الدالة تدل علي إسم القسم والطول الأقصي لهذا الإسم هو8 بايتات والإسم فقط هو علامة لهذا القسم لا أكثر، فمثلاً إذا كان هذا هو قسم الكود فسوف تري الكلمة "CODE" وهكذا.

VirtualSize **:** هي دالة تشبر إلي الحجم الفعلي لبيانات القسم بالبايت، ربما يكون حجم بيانات القسم أقل من حجم القسم علي القرص الصلب (SizeOfRawData)

VirtualAddress : هـو الـــRVA الخـــاص بالقـسم وعنــد تحميــل الـــPE فـــي الـMemory يتم إستخدام قيمة هذه الدالة فإذا كانت قيمة هذه الدالة 1000h والـPE تم تحميلة عند العنوان 400000h فسوف يصبح القسم عند العنوان 401000h.

SizeOfRawData (RawSize) : هي دالة تشير إلي حجم القسم ككل علي القرص الصلب.

PointorToRawData (RawOffset) : هــي دالــة تــشير إلــي بدايــة البيانــات الموجودة في القسم.

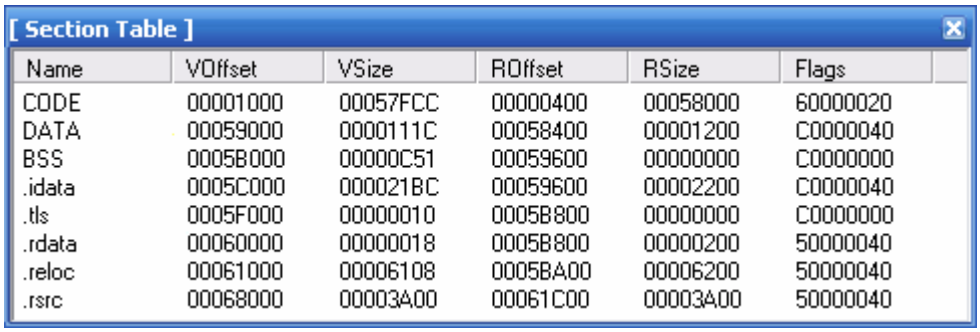
Characteristics : تشير إلي خصائص القسم من حيث أن هذا القسم قابل للقراءة فقط أو يمكن التعديل فيه أو طريقة التعامل مع هذا القسم.

والجول التالي يوضح خصائص القسم :-

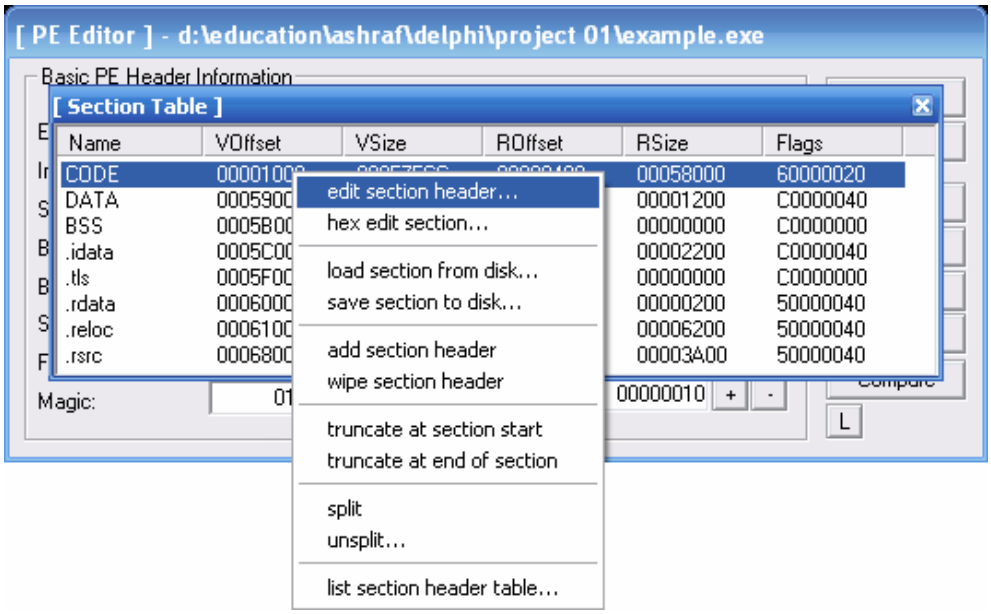
الخاصية	الوصف
00000008	Section should not be padded to next boundary
00000020	Section contains code
00000040	Section contains initialised data (which will become initialized with real values before the file is launched
00000080	Section contains uninitialised data (which will be initialized as 00 byte values before launch)
00000200	Section contains comments for the linker
00000800	Section contents will not become part of image
00001000	Section contents comdat (Common Block Data)
00008000	Section contents cannot be accessed relative to GP
00100000 to 00800000	Boundary alignment settings
01000000	Section contains extended relocations
02000000	Sections can be discarded

04000000	Section is not cacheable
08000000	Section is pageable
10000000	Section is shareable
20000000	Section is executable
40000000	Section is readable
80000000	Section is writable

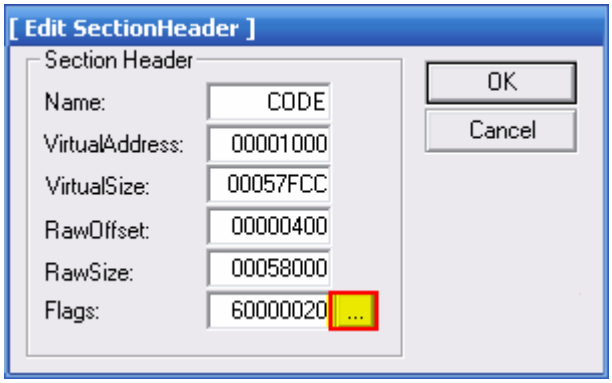
قم بتشغيل برنامج LordPE ثم إضغط علي زر Sections لتري هذا الشكل :



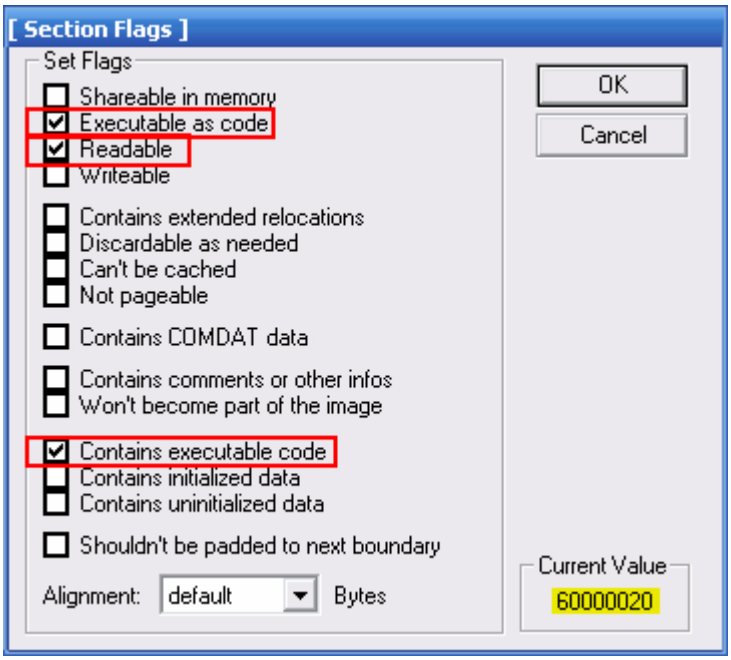
إضغط علي أول قسم ثم إضغط Right Click ثم إختر هذا الإختيار :



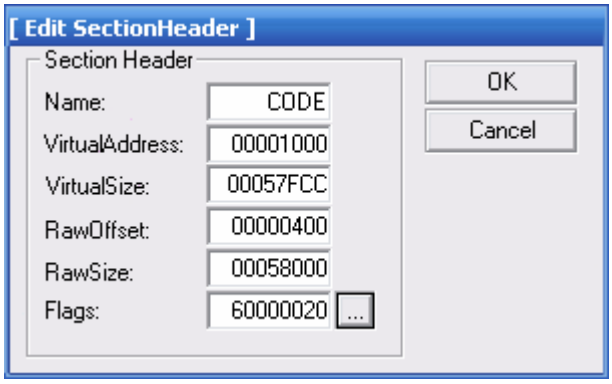
لتري هذا الشكل :



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



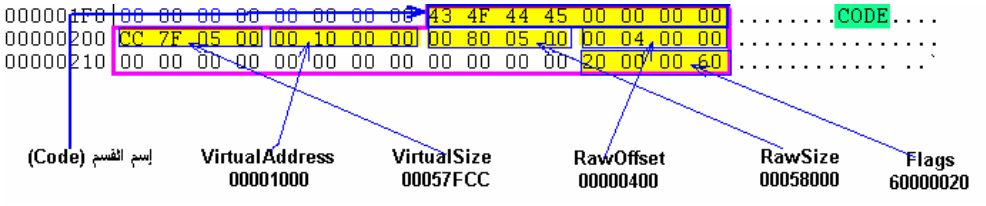
كما تري فإن القيمة 60000020 هي نتاج الخصائص المشار إليها في الشكل السابق فلو رأيت قيم هذه الخصائص في الجدول الذي يبين كل خاصية وقيمتها بالنسبة لخصائص الأقسام سوف تري هذه القيم 00000020 + 20000000 + 40000000 = 60000020 وهي القيمة المشار إليها في الشكل السابق وهكذا بالنسبة لباقي الأقسام ولو تفحصنا الشكل التالي :



سوف تجد أننا قد شرحنا كل جزئية وهذه هي المعلومات الخاصة بالقسم "CODE"

وإذا قمنا بتحميل المثال "Example.exe" داخل برنامج Hex Workshop سوف

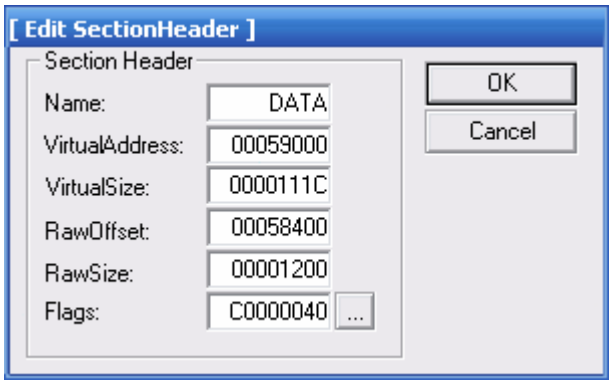
تري الشكل التالي :



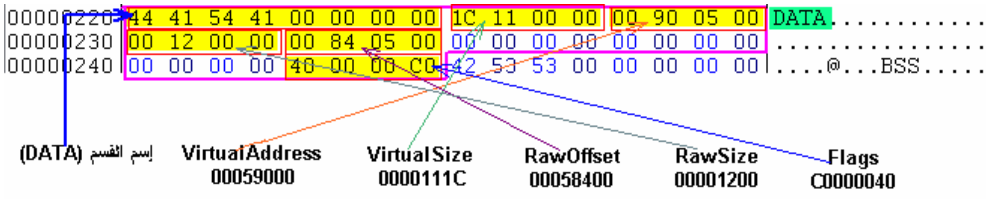
كما تري في الشكل السابق فإن المعلومات التي نراها مطابقة للمعلومات التي رأيناها

في برنامج LordPE.

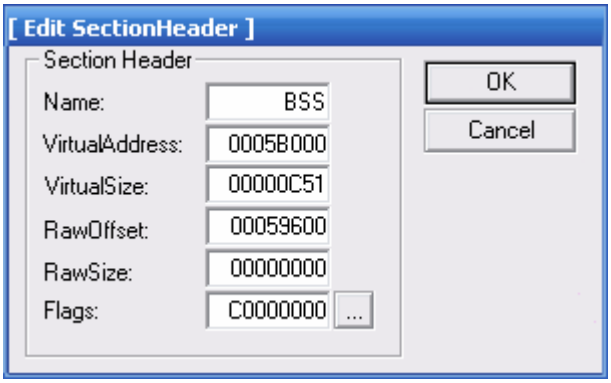
والشكل التالي يشمل المعلومات الخاصة بالقسم "DATA" كما يلي :



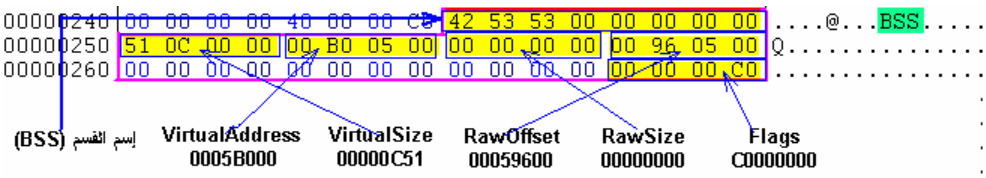
وبرنامج Hex Workshop يؤكد هذه المعلومات كما يلي :



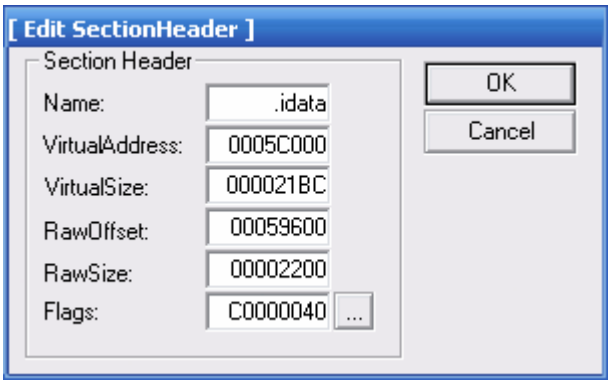
والشكل التالي يشمل المعلومات الخاصة بالقسم "BSS" كما يلي :



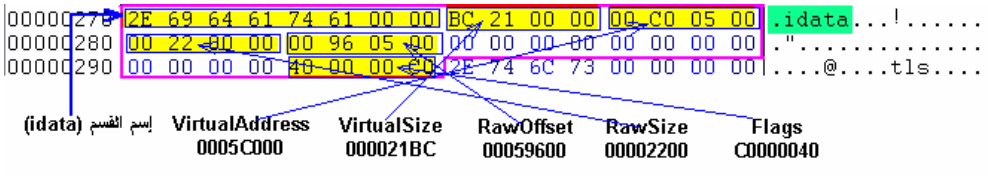
وبرنامج Hex Workshop يؤكد هذه المعلومات كما يلي :



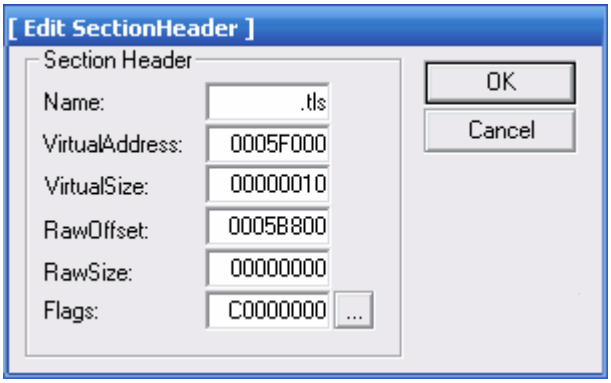
والشكل التالي يشمل المعلومات الخاصة بالقسم ".idata" كما يلي :



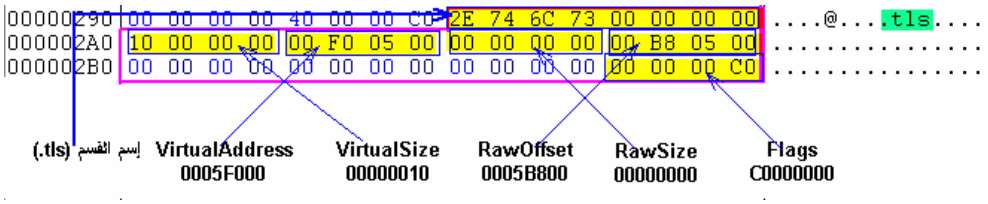
وبرنامج Hex Workshop يؤكد هذه المعلومات كما يلي :



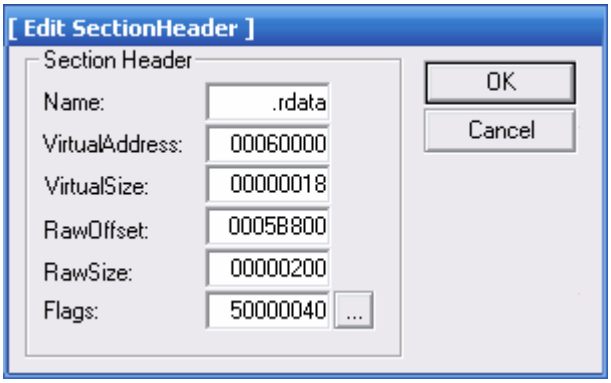
والشكل التالي يشمل المعلومات الخاصة بالقسم ".tls" كما يلي :



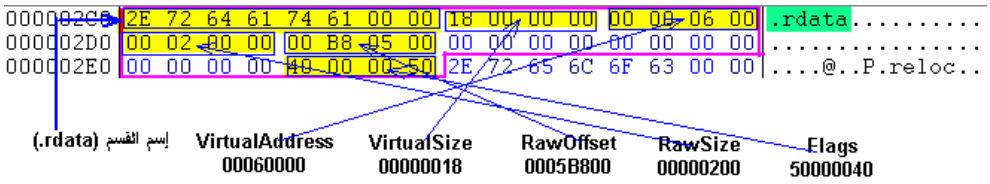
وبرنامج Hex Workshop يؤكد هذه المعلومات كما يلي :



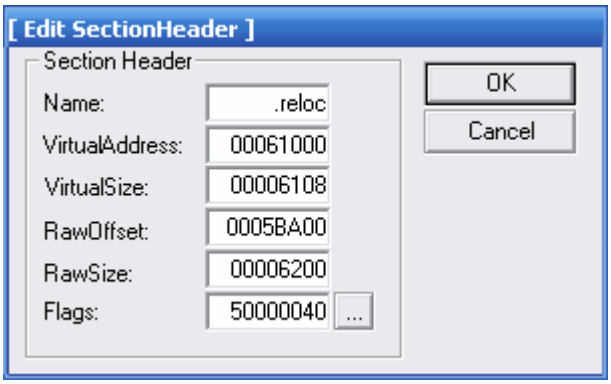
والشكل التالي يشمل المعلومات الخاصة بالقسم ".rdata" كما يلي :



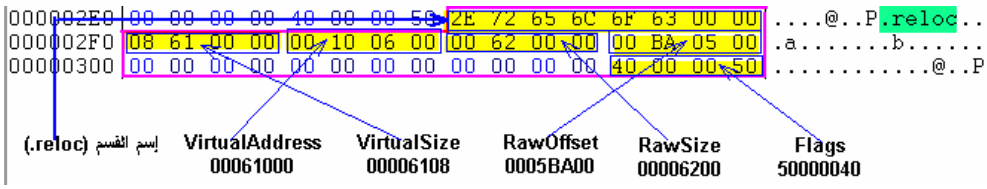
وبرنامج Hex Workshop يؤكد هذه المعلومات كما يلي :



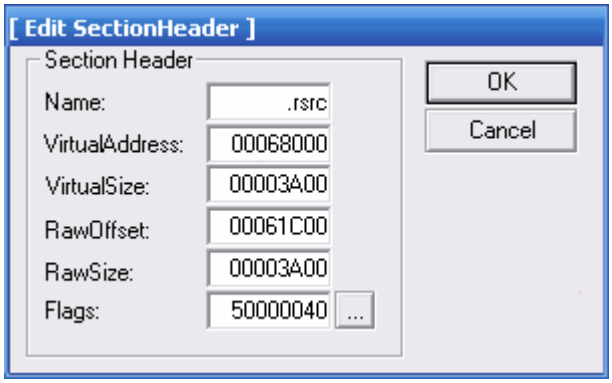
والشكل التالي يشمل المعلومات الخاصة بالقسم ".reloc" كما يلي :



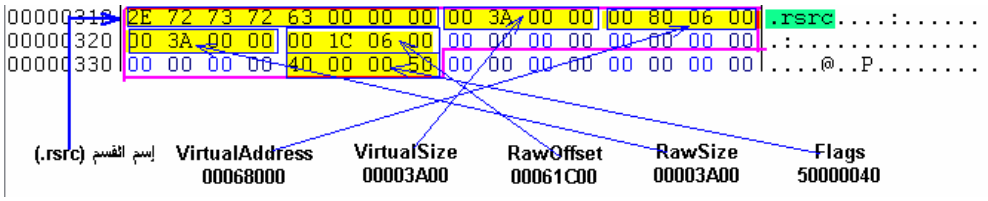
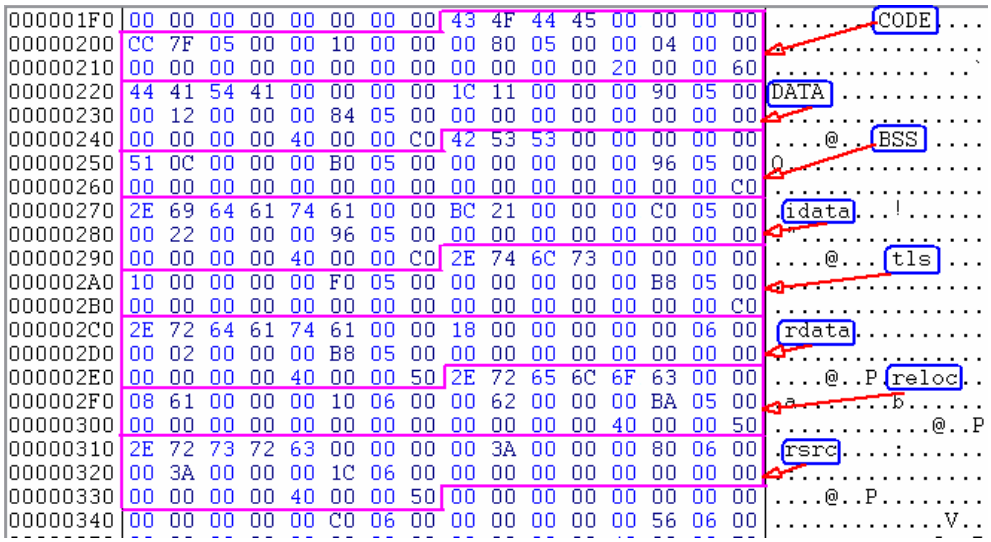
وبرنامج Hex Workshop يؤكد هذه المعلومات كما يلي :



والشكل التالي يشمل المعلومات الخاصة بالقسم ".rsrc" كما يلي :



وبرنامج Hex Workshop يؤكد هذه المعلومات كما يلي :



وكل قسم من أقسام البرنامج يمثل 40 بايت والشكل التالي يوضح أقسام البرنامج مجتمعة كلها كما يلي :