程式設計師的自我修養

Ch5. Windows PE/COFF

Samuel Chen

January 16, 2013

Outline

COFF

Common Object File Format

- ELF
- PE(Protable Executable)
  - x86: Support NT, 95, XP, Vista, CE
  - 64bit: PE32+
  - custom sections
- Default object file format of Visual C++: COFF

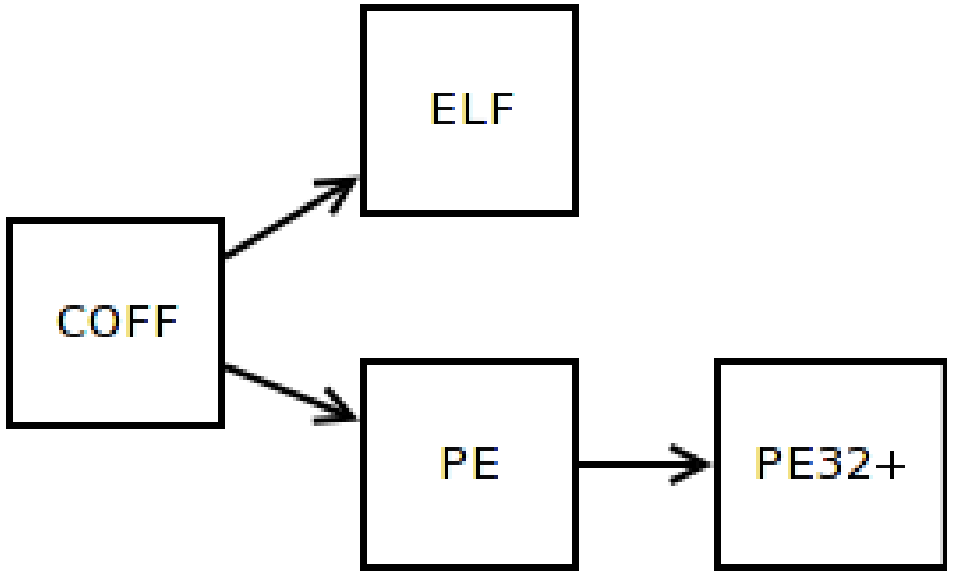


Figure : COFF, ELF, PE and PE32+

Custom Section

```
1 #pragma data_seg("FOO")
2 int global = 1;
3 #pragma data_seg(".data")
```

#pragma

- line1 #pragma之後的全域變數放進"FOO" section
- line3 #再切換回來放進去預設的".data" section

Visual C++

- Compiler: *cl*
- Linker: *link*
- COFF Binary Dump tool: *dumpbin*
- Visual C++ Express 2010(Free) *download link*

Complie with VC++

```
1 cl /c /Za SimpleSection.c
2
3 dumpbin /ALL SimpleSection.obj > SimpleSection.txt
4 dumpbin /SUMMARY SimpleSection.obj
```

cl

- /c: compile only
- /Za: 停用擴充功能

dumpbin

- /ALL: 輸出obj 所有相關資訊
- /SUMMARY: 輸出obj 基本資訊

COFF Object File Format

- Image File Header
- Image Section Header
- Sections
  - *.text*
  - *.data*
  - *.debug$S*
- Symbol Table
- "VC\PlatformSDK \include **WinNT.h**"

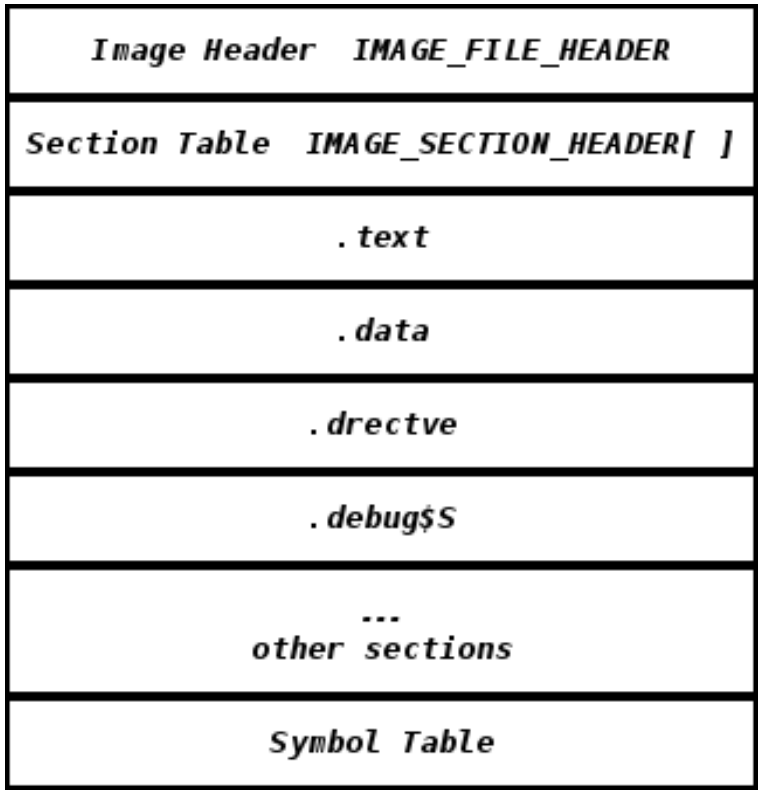
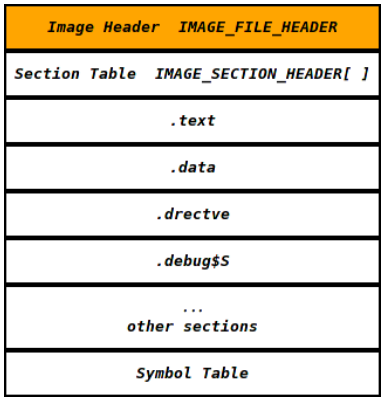


Figure : COFF File Format

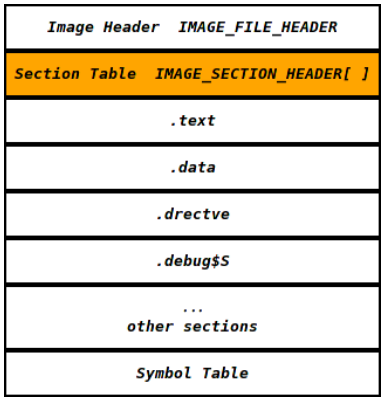
Image File Header




```
Microsoft (R) COFF/PE Dumper Version 10.00.30319.01
Copyright (C) Microsoft Corporation.  All rights reserved.


Dump of file SimpleSection.obj

File Type: COFF OBJECT

FILE HEADER VALUES
             14C machine (x86)
               5 number of sections
        50F452BE time date stamp Tue Jan 12 19:47:26 2013
             1E8 file pointer to symbol table
              14 number of symbols
               0 size of optional header
               0 characteristics

```

Figure : image file header information

Image Section Header




```
 18 SECTION HEADER #1
 19 .drectve name
 20 +-- 21 lines: 0 physical address--
 41 SECTION HEADER #2
 42 .debug$S name
 43 +-- 23 lines: 0 physical address--
 66 SECTION HEADER #3
 67    .data name
 68 +-- 16 lines: 0 physical address--
 84 SECTION HEADER #4
 85    .text name
 86 +-- 30 lines: 0 physical address--
116 SECTION HEADER #5
117     .bss name
118 +--- 12 lines: 0 physical address-
```

Figure : image section header information

Section Attributes

- name
- virtual size/addr
- size of raw data
- characteristics

.drectve section

連結指示資訊

```
18 SECTION HEADER #1
19 .drectve name
20        0 physical address
21        0 virtual address
22       18 size of raw data
23       DC file pointer to raw data (000000DC to 000000F3)
24        0 file pointer to relocation table
25        0 file pointer to line numbers
26        0 number of relocations
27        0 number of line numbers
28   100A00 flags
29          Info
30          Remove
31          1 byte align
32
33 RAW DATA #1
34   00000000: 20 20 20 2F 44 45 46 41 55 4C 54 4C 49 42 3A 22     /DEFAULTLIB:"
35   00000010: 4C 49 42 43 4D 54 22 20                            LIBCMT"
36
37    Linker Directives
38    -----------------
39    /DEFAULTLIB:"LIBCMT"
40
```

- name: Directive 的縮寫
- Characteristics: 0x100A00
- 0x100A00 = 0x100000 + 0x800 + 0x200
  p.140 表5-2
- LIBCMT: Library C Multithreaded, 表示
  靜態連結的多緒程C函式庫

.debug section

除錯資訊

```
SECTION HEADER #2
.debug$S name
       0 physical address
       0 virtual address
      68 size of raw data
      F4 file pointer to raw data (000000F4 to 0000015B)
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
42100040 flags
         Initialized Data
         Discardable
         1 byte align
         Read Only

RAW DATA #2
  00000000: 04 00 00 00 F1 00 00 00 59 00 00 00 1B 00 01 11  ....n...Y.......
  00000010: 00 00 00 00 59 3A 5C 53 69 6D 70 6C 65 53 65 63  ....Y:\SimpleSec
  00000020: 74 69 6F 6E 2E 6F 62 6A 00 3A 00 3C 11 00 22 00  tion.obj.:.<..".
  00000030: 00 07 00 10 00 00 00 6F 76 01 00 10 00 00 00 6F  .......ov......o
  00000040: 76 01 00 4D 69 63 72 6F 73 6F 66 74 20 28 52 29  v..Microsoft (R)
  00000050: 20 4F 70 74 69 6D 69 7A 69 6E 67 20 43 6F 6D 70   Optimizing Comp
  00000060: 69 6C 65 72 00 00 00 00                          iler....

```

- name:
  - .debug$S(symbol contained)
  - .debug$P(precompiled header files contained)
  - .debug$T(type contained)

Symble Table

```
131 COFF SYMBOL TABLE
132 000 00AA766F ABS    notype       Static       | @comp.id
133 001 00000001 ABS    notype       Static       | @feat.00
134 002 00000000 SECT1  notype       Static       | .drectve
135     Section length   18, #relocs    0, #linenums    0, chec
136 004 00000000 SECT2  notype       Static       | .debug$S
137     Section length   68, #relocs    0, #linenums    0, chec
138 006 00000004 UNDEF  notype       External     | _global_uni
139 007 00000000 SECT3  notype       Static       | .data
140     Section length    C, #relocs    0, #linenums    0, chec
141 009 00000000 SECT3  notype       External     | _global_ini
142 00A 00000004 SECT3  notype       Static       | $SG893
143 00B 00000008 SECT3  notype       Static       | ?static_var
    ain'::`2'::static_var)
144 00C 00000000 SECT4  notype       Static       | .text
145     Section length   4E, #relocs    5, #linenums    0, chec
146 00E 00000000 SECT4  notype ()    External     | _func1
147 00F 00000000 UNDEF  notype ()    External     | _printf
148 010 00000020 SECT4  notype ()    External     | _main
149 011 00000000 SECT5  notype       Static       | .bss
150     Section length    4, #relocs    0, #linenums    0, chec
151 013 00000000 SECT5  notype       Static       | ?static_var
    main'::`2'::static_var2)
152
153 String Table Size = 0x5D bytes
```

Figure : COFF symbol table

- column1: index
- column2: size
- column3: position
  - section number: SECT1, SECT2 ...
  - global: ABS
- column4: type(notype/notype ()), 可供強弱符號運用
- column5: scope
  - External: 全域
  - Static: 區域
- column6: symbol name
- String Table Size

- COFF extension
- 2 major differeces
  1. Started by **DOS MZ File Header and Stub** compitible with MZ(old DOS .exe file format)
  2. **IMAGE_NT_HEADERS**
- IMAGE_DOS_HEADER:
  - "e_cs, e_ip", point to "DOS Stub"
  - "This program cannot be run in DOS"
  - **"e_lfanew"**
    - MZ: 0
    - PE: offset to **IMAGE_NT_HEADERS**

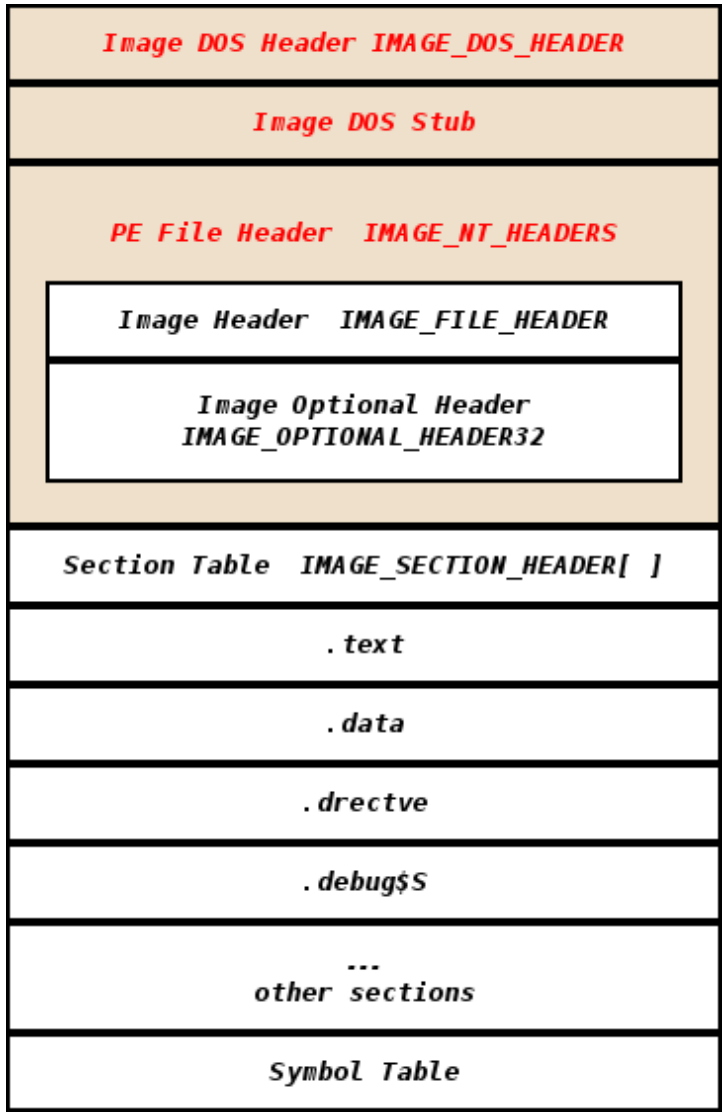


Figure : PE File Format

PE File Header

IMAGE_NT_HEADERS

- PE 真正的HEADER, Optional for COFF, but required for PE-executable file, include DLL files
- Signiture: PE\0\0
- PE data directory: defined in IMAGE_OPTIONAL_HEADER
  - 匯入表
  - 重定表
  - 資源表
  - 異常表
  - 重定表
  - 除錯資訊表
  - 緒程私有儲存表(TLS)... 等的位置與長度
- WinNT.h

Thank you!