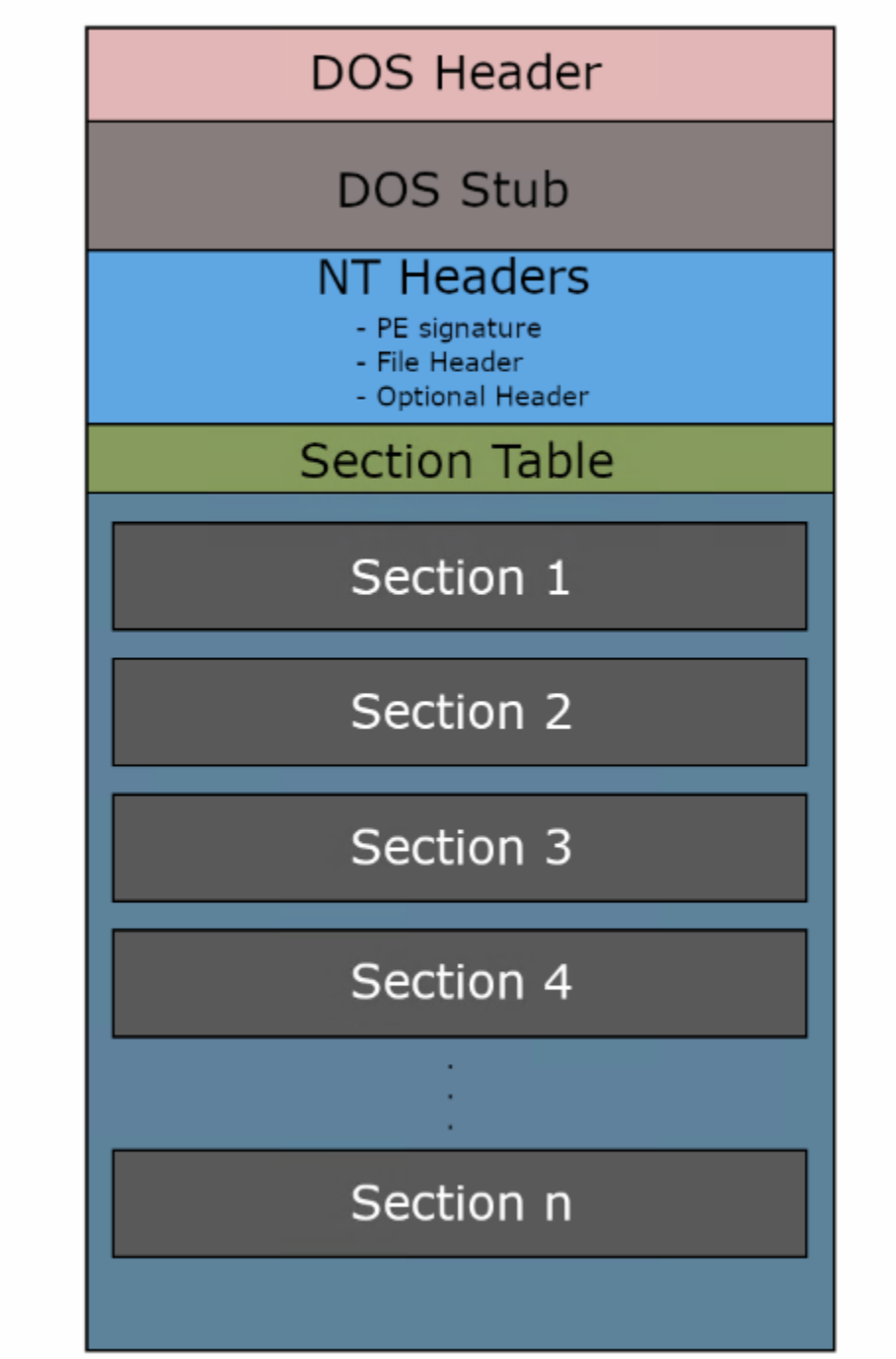
PE header

1. PE header là gì?

PE là viết tắt của Portable Executable, đây là một định dạng tệp cho các tệp thực thi được sử dụng trong hệ điều hành Windows. Tệp PE là một cấu trúc dữ liệu chứa thông tin cần thiết cho bộ tải hệ điều hành để có thể tải tệp thực thi đó vào bộ nhớ và thực thi nó.



Đây là cấu trúc của 1 tệp PE

-DOS header là phần mở đầu của tệp với độ dài 64 byte bao gồm các thông tin cơ bản của file
-DOS Stub : vị trí của nó nằm ngay sau DOS header với chức năng in thông báo “This program cannot be run in DOS mode”
-NT header: gồm 3 phần chính

- PE signature
- File header
- Optional Header

-Section Table (Section Headers) : một mảng Image Section Headers chứa thông tin về Section mà nó đề cập đến
-Section : nơi lưu trữ nội dung của file bao gồm dữ liệu, tài nguyên mà chương trình sử dụng và cả mã thực thi. Mỗi section sẽ có một nội dung và mục đích khác nhau

2. DOS header

Đây là phần mở đầu tệp dài 64 byte với cấu trúc như sau

Member	Offset	Size	Value
e_magic	00000000	Word	5A4D
e_cblp	00000002	Word	0090
e_cp	00000004	Word	0003
e_crlc	00000006	Word	0000
e_cparhdr	00000008	Word	0004
e_minalloc	0000000A	Word	0000
e_maxalloc	0000000C	Word	FFFF
e_ss	0000000E	Word	0000
e_sp	00000010	Word	00B8
e_csum	00000012	Word	0000
e_ip	00000014	Word	0000
e_cs	00000016	Word	0000
e_lfarlc	00000018	Word	0040
e_ovno	0000001A	Word	0000
e_res	0000001C	Word	0000
	0000001E	Word	0000
	00000020	Word	0000
	00000022	Word	0000
e_oemid	00000024	Word	0000
e_oeminfo	00000026	Word	0000
e_res2	00000028	Word	0000
	0000002A	Word	0000
	0000002C	Word	0000
	0000002E	Word	0000
	00000030	Word	0000
	00000032	Word	0000
	00000034	Word	0000
	00000036	Word	0000
	00000038	Word	0000
	0000003A	Word	0000
e_lfanew	0000003C	Dword	000000C0

Trong DOS header cần quan tâm tới 2 trường quan trọng là

- e_magic: đây là phần tử đầu tiên của DOS Header, chiếm 2 byte, nó đóng vai trò như một chữ ký đánh dấu tệp là tệp thực thi MS-DOS nếu giá trị là 5A4D
- e_lfanew: đây là phần tử cuối cùng của DOS heade nằm ở vị trí 0x3C chiếm 4 byte nó cho ta biết vị trí của nt header

3. NT header

Signature là phần tử đầu tiên của NT header, nó có giá trị cố định là 0x50450000

File header chứa thông tin cơ bản của file PE
Đây là cấu trúc của file header

```
typedef struct _IMAGE_FILE_HEADER {
    WORD    Machine;
    WORD    NumberOfSections;
    DWORD   TimeDateStamp;
    DWORD   PointerToSymbolTable;
    DWORD   NumberOfSymbols;
    WORD    SizeOfOptionalHeader;
    WORD    Characteristics;
} IMAGE_FILE_HEADER, *PIMAGE_FILE_HEADER;
```

Đây là ví dụ về giá trị và địa chỉ của file header

Member	Offset	Size	Value	Meaning
Machine	000000C4	Word	014C	Intel 386
NumberOfSections	000000C6	Word	0003	
TimeDateStamp	000000C8	Dword	633A977B	
PointerToSymbolTa	000000CC	Dword	00000000	
NumberOfSymbols	000000D0	Dword	00000000	
SizeOfOptionalHea	000000D4	Word	00E0	
Characteristics	000000D6	Word	010F	Click here

- Machine: đây là thông số cho biết loại máy (kiến trúc CPU)
- NumberOfSections: chứa số lượng Section của file
- TimeDateStamp: Cho biết thời gian tệp được tạo
- SizeOfOptionalHeader: Kích thước của Optional Header
- Characteristics: Các cờ chỉ ra các thuộc tính của tệp.

Optional Header (IMAGE_OPTIONAL_HEADER)
Đây là phần quan trọng nhất của NT header
Nó có 2 cấu trúc riêng biệt cho 32 bit và 64 bit

32 bit

```
typedef struct _IMAGE_OPTIONAL_HEADER {
    //
    // Standard fields.
    //

    WORD    Magic;
    BYTE    MajorLinkerVersion;
    BYTE    MinorLinkerVersion;
    DWORD   SizeOfCode;
    DWORD   SizeOfInitializedData;
    DWORD   SizeOfUninitializedData;
    DWORD   AddressOfEntryPoint;
    DWORD   BaseOfCode;
    DWORD   BaseOfData;

    //
    // NT additional fields.
    //

    DWORD   ImageBase;
    DWORD   SectionAlignment;
    DWORD   FileAlignment;
    WORD    MajorOperatingSystemVersion;
    WORD    MinorOperatingSystemVersion;
    WORD    MajorImageVersion;
    WORD    MinorImageVersion;
    WORD    MajorSubsystemVersion;
    WORD    MinorSubsystemVersion;
    DWORD   Win32VersionValue;
    DWORD   SizeOfImage;
    DWORD   SizeOfHeaders;
    DWORD   CheckSum;
    WORD    Subsystem;
    WORD    DllCharacteristics;
    DWORD   SizeOfStackReserve;
    DWORD   SizeOfStackCommit;
    DWORD   SizeOfHeapReserve;
    DWORD   SizeOfHeapCommit;
    DWORD   LoaderFlags;
    DWORD   NumberOfRvaAndSizes;
    IMAGE_DATA_DIRECTORY DataDirectory[IMAGE_NUMBEROF_DIRECTORY_ENTRIES];
} IMAGE_OPTIONAL_HEADER32, *PIMAGE_OPTIONAL_HEADER32;
```

64 bit

```
typedef struct _IMAGE_OPTIONAL_HEADER64 {
    WORD        Magic;
    BYTE        MajorLinkerVersion;
    BYTE        MinorLinkerVersion;
    DWORD       SizeOfCode;
    DWORD       SizeOfInitializedData;
    DWORD       SizeOfUninitializedData;
    DWORD       AddressOfEntryPoint;
    DWORD       BaseOfCode;
    ULONGLONG   ImageBase;
    DWORD       SectionAlignment;
    DWORD       FileAlignment;
    WORD        MajorOperatingSystemVersion;
    WORD        MinorOperatingSystemVersion;
    WORD        MajorImageVersion;
    WORD        MinorImageVersion;
    WORD        MajorSubsystemVersion;
    WORD        MinorSubsystemVersion;
    DWORD       Win32VersionValue;
    DWORD       SizeOfImage;
    DWORD       SizeOfHeaders;
    DWORD       CheckSum;
    WORD        Subsystem;
    WORD        DllCharacteristics;
    ULONGLONG   SizeOfStackReserve;
    ULONGLONG   SizeOfStackCommit;
    ULONGLONG   SizeOfHeapReserve;
    ULONGLONG   SizeOfHeapCommit;
    DWORD       LoaderFlags;
    DWORD       NumberOfRvaAndSizes;
    IMAGE_DATA_DIRECTORY DataDirectory[IMAGE_NUMBEROF_DIRECTORY_ENTRIES];
} IMAGE_OPTIONAL_HEADER64, *PIMAGE_OPTIONAL_HEADER64;
```

- magic: xác định xem tệp thực thi này 32bit hay 64 bit

0x10B : 32 bit
0x20B: 64 bit

- SizeOfCode: Trường này chứa kích thước của (.text) Section hoặc tổng của tất cả các phần nếu có nhiều phần
- SizeOfInitializedData: Trường này chứa kích thước của (.data) Section hoặc tổng của tất cả các phần nếu có nhiều phần

- SizeOfUninitializedData: Trường này chứa kích thước của (.bss) Section hoặc tổng của tất cả các phần nếu có nhiều phần
- AddressOfEntryPoint (RVA): địa chỉ ảo tương đối của câu lệnh đầu tiên sẽ được thực thi. Nếu muốn chương trình bắt đầu từ một địa chỉ khác (để thực thi câu lệnh với mục đích khác) thì cần thay đổi địa chỉ này về địa chỉ tương đối của câu lệnh muốn thực thi
- BaseOfCode: RVA của phần bắt đầu đoạn mã khi tệp được tải vào bộ nhớ.
- BaseOfData (PE32): RVA của phần bắt đầu phần dữ liệu khi tệp được tải vào bộ nhớ.
- ImageBase: Địa chỉ được ưu tiên nạp cho PE file.
- SizeOfImage: Kích thước của image (tính bằng byte), bao gồm tất cả các tiêu đề
- SizeOfHeaders: Kích thước của tất cả headers và section table, bằng kích thước file trừ đi tổng kích thước của các section trong file.
- DataDirectory: Một mảng cấu trúc IMAGE_DATA_DIRECTORY.

4. Section Table (Section Headers)

Section Table là thành phần ngày sau PE Header, bao gồm một mảng những cấu trúc IMAGE_SECTION_HEADER, mỗi phần tử chứa thông tin về một section trong PE file
Section header là 1 cấu trúc như sau

```
typedef struct _IMAGE_SECTION_HEADER {
    BYTE    Name[IMAGE_SIZEOF_SHORT_NAME];
    union {
            DWORD   PhysicalAddress;
            DWORD   VirtualSize;
    } Misc;
    DWORD   VirtualAddress;
    DWORD   SizeOfRawData;
    DWORD   PointerToRawData;
    DWORD   PointerToRelocations;
    DWORD   PointerToLinenumbers;
    WORD    NumberOfRelocations;
    WORD    NumberOfLinenumbers;
    DWORD   Characteristics;
} IMAGE_SECTION_HEADER, *PIMAGE_SECTION_HEADER;
```

ví dụ

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	000002F8	00001000	00000400	00000400	00000000	00000000	0000	0000	60000020
.rdata	0000012C	00002000	00000200	00000800	00000000	00000000	0000	0000	40000040
.data	000000B4	00003000	00000200	00000A00	00000000	00000000	0000	0000	C0000040

Name: tên của section
PhysicalAddress hoặc VirtualSize: Kích thước thật sự của dữ liệu trên section tính theo byte
VirtualAddress: RVA của section, là giá trị để ánh xạ khi section được load lên bộ nhớ

SizeOfRawData: Kích thước section data trên ổ đĩa
PointerToRawData: là offset từ vị trí đầu file tới section data.
Characteristics: đặc tính của section: thực thi, dữ liệu khởi tạo

Về RVA
RVA viết tắt của Relative Virtual Offset địa chỉ ảo tương đối. Được sử dụng để mô tả memory offset nếu ko biết địa chỉ base address
File offset = RVA - Virtual Offset + Raw Offset