- DOS MZ Header gồm 64 byte:

+ Trong đó, 2 byte đàu tiên: 4D 5A => MZ (signature)

+4 byte cuối cùng biểu thị địa chỉ vị trí offset của PE Header

+Tiếp theo vùng nhớ cho đến vị trí địa chỉ offset của PE Header là Dos stub => Đây xem như là chương trình con hiển thị thông báo lỗi

- PE Header gồm: signature (4 byte) + fileHeader(20 byte) + optionalHeader(224 byte)

- FileHeader:

+ Machine(2 byte): Xác định dòng máy PE file sẽ thực thi (14c-i386)

+NumberOfSection:(2 byte) Cho biết số Section cua File

+ Characteristics:(2 byte) cách offset 22byte: Là bít cờ, xác định định dạng của PE file

0x0102 là file \*.exe

0x2102 là file \*.DLL

- Optional Header:

+ Magic (2 byte) cách offset 24 byte: Xác định là file 32(0B01) hay 64(0B20)

+ AddressOfEntryPoint (4 byte) cách offset 40 byte: Chứa địa chỉ ảo tương đối (RVA) của câu lệnh đầu tiên sẽ được thực thi khi PE Loader sẵn sàng để chạy PE File(.text )

=> Nếu muốn chương trình bắt đầu từ một địa chỉ để thực thi một câu lệnh khác với mục đích khác thì chỉ cần thay đổi địa chỉ này về địa chỉ tương đối của câu lệnh muốn thực thi.

- ImageBase:(4 byte) cách offset 52 byte Địa chỉ nạp được ư tiên cho PE File

-SectionAlignment:(4 byte) cách offset 56 byte Là phần liên kết của các section trong bộ nhớ

-FileAlignment:(4 byte) cách offset 60 byte Là phần liên kết của các Section trong File

-SizeOf Image(4 byte):Toàn bộ kích thước của PE image trong bộ nhớ, là tổng của tât cả các headers và section được liên kết tới Section Alignment

- SizeOfHeader(4byte): Kích thước của tất cả các headers + section table, bằng kích thước của file trừ đi kích thước được tổng hợp của toàn bộ sections trong file. Nó sử dụng như một file offset của section đầu tiên trong PE file

- DataDirectory (128 byte) là một mảng gồm 16 phần tử IMAGE\_DATA\_DIRECTORY, trong đó mỗi phần liên quan đến một cấu trúc dữ liệu quan trọng trong PE file

+Cấu trúc của Data Directory có 2 thành phần gồm thông tin và kích thước của cấu trúc dữ liệu.

. VirtualAddress là một địa chỉ ảo tương đối(RVA) của cấu trúc dữ liệu (4 byte)

. Size bao gồm kích thước theo byte của cấu trúc dữ liệu(4byte)

\* Section Table là thành phần tiếp theo ngay sau PE Header, nó là mảng của những cấu trúc IMAGE\_SECTION\_HEADER

- Mỗi phần tử sẽ chứa thông tin về một section trong PE File ví dụ như thuộc tính của nó và offset ảo(virtual offset).

- Mỗi Struct Header là 40 byte vầ k có padding giữa chúng tức là sẽ không chèn thêm các byte có giá trị 00h vào.

+ Name (8 byte): Tên này chỉ là một nhãn và có thể để trống. Chú ý, đây không phải một chuỗi ASCII vì vậy nó k cần phải kết thúc bằng việc thêm 0

+ VirtualSize(4 byte): Kicht thước thật sự của Section'data theo byte. Nó có thể nhỏ hơn kích thước của section trên đĩa và sẽ là những gì mà trình loader định rõ vị trí trong bộ nhớ cho section này.

+ Virtual Address(4 byte) RVA của Section. PE Loader sẽ phân tích và sử dụng giá trị trong trường này khi nó ánh xạ section vào trong bộ nhớ. Vì vậy nếu giá trị trong trường này là 1000h và PE File được nạp tại địa chỉ 400000h, thì section sẽ được nạp tại địa chỉ 401000h

+ SizeOfRawData (4 byte) Kích thước của section's data trong file trên đĩa, được làm tròn lên bội số tiếp theo của sự liên kết file bởi trình biên dịch

+PointerToRawData (4 byte) thành phần này rất hữu dụng bởi vì nó là offset từ vị trí bắt đàu của file cho tới phần section'data. Nếu nó có giá trị là 0, thì section's data không được chứa trong file và sẽ không bị bó buộc vào thời gian nạp (load time) . Trình PE Loader sẽ sử dụng giá trị trong trường này để tìm kiếm phần data trong section là ở đâu trong file.

+ Characteristics(4 byte) Bao gồm các cờ ví dụ như section này có thể chứa executable code, initialized data, uninitialized data, có thể được ghi hoặc đọc.

=> Sau khi có được Section Header chúng ta sẽ tìm kiếm các sections. Trong file ở trên ổ đĩa, mỗi section bắt đầu tại một ofset mà là bội số lần của giá trị FileAlignment được tìm thấy trong Optionalheader. Giữa các section'data sẽ là các byte 00 được thêm vào.

- Khi được nạp lên RAM, các Section luôn được bắt đầu trên một page boundary, nên byte đầu tiên của section tượng ứng với một trang bộ nhớ (memory page).

+ Các trang trên X86 là 4KB aligned

+Các trang trên x64 là 8KB aligned

=> Giá trị liên kết này (alignment value) được lưu trong SectionAlignment, và cũng được lưu trong OptionalHeader