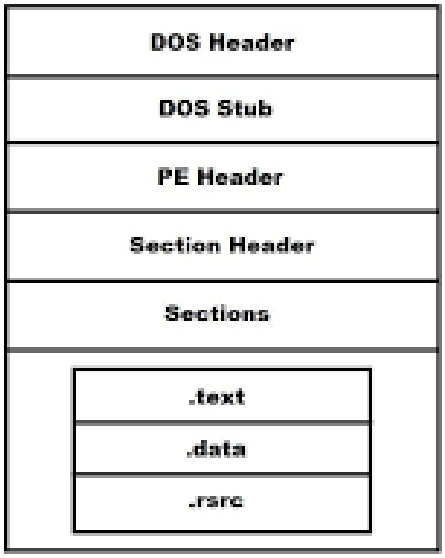
Cấu trúc file PE (Portable Executable) của Windows

PE (Portable Executable) là định dạng file tiêu chuẩn cho các file thực thi (EXE), file thư viện liên kết động (DLL), file object (.OBJ), và các loại file khác trên hệ điều hành Windows 32-bit và 64-bit. Tên gọi "Portable" (Linh động) có nghĩa là định dạng này được thiết kế để có thể chạy trên mọi phiên bản Windows.

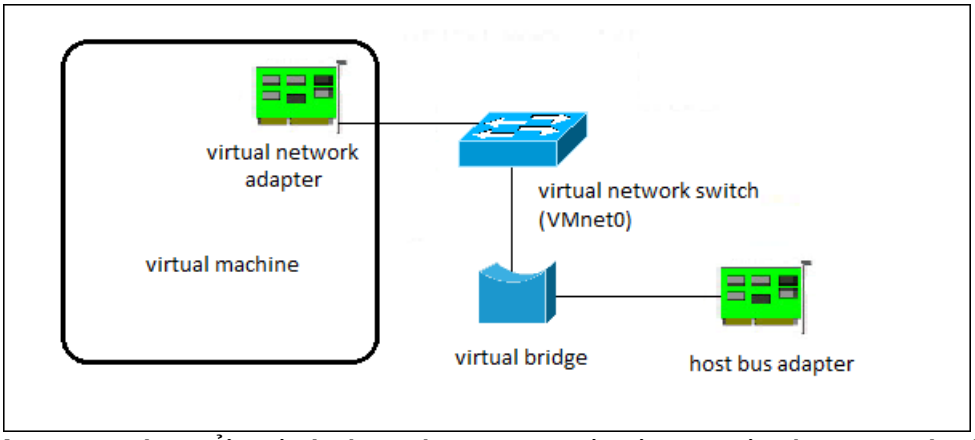


1. **DOS Header (IMAGE_DOS_HEADER):**
    - Đây là phần đầu tiên của file, tồn tại chủ yếu để tương thích ngược với hệ điều hành MS-DOS cũ.
    - Nó chứa một "magic number" là `MZ` (ký tự đầu tiên của file khi xem ở dạng hexa), là tên viết tắt của Mark Zbikowski, một trong những kiến trúc sư của MS-DOS.
2. **PE Header (IMAGE_NT_HEADERS):**
    - **File Header (IMAGE_FILE_HEADER):** Chứa các thông tin cơ bản về file, như: kiến trúc máy nó được biên dịch cho (x86, x64), số lượng các section, và thời gian file được tạo.
    - **Optional Header (IMAGE_OPTIONAL_HEADER):** Tên gọi "Optional" (Tùy chọn) hơi gây hiểu nhầm, vì phần này **cực kỳ quan trọng**. Nó chứa các thông tin thiết yếu mà bộ nạp hệ điều hành (OS Loader) cần để chạy file, ví dụ:
        - `AddressOfEntryPoint`: Địa chỉ nơi chương trình bắt đầu thực thi mã lệnh đầu tiên.
        - `ImageBase`: Địa chỉ ưu tiên trong bộ nhớ ảo mà file muốn được nạp vào.
        - `DataDirectories`: Một mảng các con trỏ chỉ đến các cấu trúc dữ liệu quan trọng khác, như bảng import, bảng export, thông tin tài nguyên, v.v.

3. **Section Table (Bảng Section):**
    - Đây là một "bảng chỉ dẫn" nằm ngay sau PE Header. Nó là một mảng các `IMAGE_SECTION_HEADER`.
    - Mỗi entry trong bảng này mô tả một **Section**: tên của nó (ví dụ: `.text`, `.data`), kích thước, vị trí trong file, và các thuộc tính (quyền đọc, ghi, thực thi).
4. **Sections (Các Section):**
    - Đây là nơi chứa dữ liệu và mã lệnh thực sự của chương trình, được chia thành các "phòng" (section) chức năng:
        - **.text**: Chứa mã lệnh thực thi (mã máy) của chương trình. Section này thường có quyền chỉ đọc và thực thi.
        - **.data**: Chứa các biến toàn cục và biến tĩnh đã được khởi tạo giá trị.
        - **.rdata**: Chứa dữ liệu chỉ đọc, ví dụ như các chuỗi hằng số.
        - **.bss**: Chứa các biến toàn cục và biến tĩnh chưa được khởi tạo. Section này không chiếm dung lượng trong file mà chỉ định rằng cần phải cấp phát một vùng nhớ khi chương trình được nạp.
        - **.idata (Import Data):** Chứa **Import Address Table (IAT)**, là một bảng liệt kê các hàm mà chương trình cần "vay mượn" từ các file DLL khác (ví dụ: các hàm API của Windows).
        - **.rsrc (Resources):** Chứa các tài nguyên của chương trình như icon, hình ảnh, menu, thông tin phiên bản.

Các chế độ mạng (Network Mode) của VMware

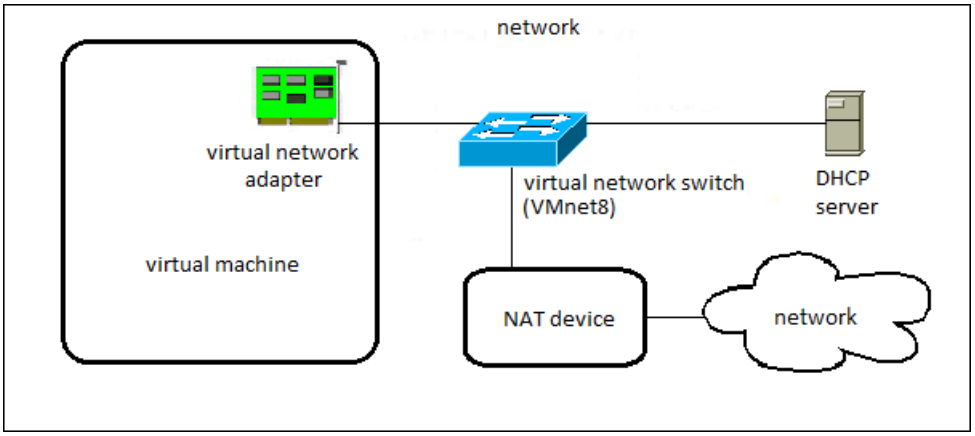
1. **Bridged Mode**



Cách hoạt động: Ở chế độ này, máy ảo được kết nối trực tiếp vào mạng vật lý mà máy Host đang sử dụng. Nó hoạt động như một máy tính độc lập trên mạng của bạn.

- **Địa chỉ IP:** Máy ảo sẽ nhận một địa chỉ IP riêng từ router của mạng vật lý (thông qua DHCP), cùng dải mạng với máy Host và các thiết bị khác.
- **Kết nối:** Máy ảo có thể truy cập các thiết bị khác trên mạng vật lý và ngược lại, các thiết bị khác cũng có thể "nhìn thấy" và truy cập vào máy ảo.
- **Tương tự như:** Cắm một chiếc máy tính thật vào router mạng của bạn.
- **Khi nào dùng:** Khi bạn cần máy ảo hoạt động như một server hoặc một thành viên đầy đủ trong mạng LAN (ví dụ: để host một website nội bộ, chạy một file server).

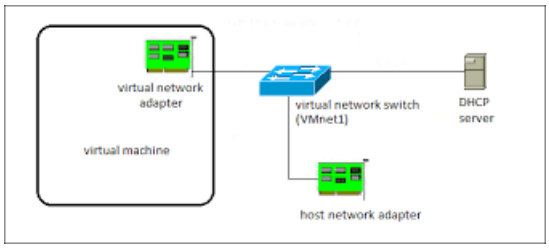
2. **NAT (Network Address Translation) Mode**



Cách hoạt động: Máy Host sẽ tạo ra một mạng riêng ảo và hoạt động như một router cho các máy ảo. Khi máy ảo muốn kết nối ra ngoài (internet), máy Host sẽ "dịch" địa chỉ IP riêng của máy ảo thành địa chỉ IP của chính nó.

- **Địa chỉ IP:** Máy ảo sẽ nhận một địa chỉ IP từ một DHCP server ảo do VMware tạo ra, nằm trong một dải mạng riêng.
- **Kết nối:**
  - **Từ trong ra ngoài:** Máy ảo có thể truy cập internet và các thiết bị trong mạng vật lý một cách dễ dàng.
  - **Từ ngoài vào trong:** Mặc định, các thiết bị bên ngoài **không thể** truy cập trực tiếp vào máy ảo (vì nó đang "nấp" sau máy Host).
- **Tương tự như:** Các thiết bị trong nhà bạn kết nối vào router Wi-Fi. Router chỉ có một địa chỉ IP công khai, nhưng tất cả các thiết bị đều có thể dùng chung để ra internet.
- **Khi nào dùng:** Đây là chế độ mặc định và phổ biến nhất. Rất tiện lợi khi bạn chỉ cần máy ảo có kết nối internet mà không cần phải cấu hình phức tạp.

3. **Host-Only Mode**



- **Cách hoạt động:** VMware tạo ra một mạng ảo hoàn toàn riêng tư, chỉ bao gồm máy Host và các máy ảo được cấu hình ở chế độ này.
- **Địa chỉ IP:** Máy ảo nhận IP từ DHCP server ảo của VMware.
- **Kết nối:** Máy ảo có thể kết nối với máy Host và các máy ảo khác trong cùng mạng Host-Only, nhưng **hoàn toàn không thể** kết nối ra mạng vật lý bên ngoài hay internet.
- **Tương tự như:** Dùng một sợi cáp mạng nối trực tiếp máy ảo và máy Host với nhau, tạo thành một mạng cô lập.
- **Khi nào dùng:** Rất hữu ích để tạo một môi trường lab an toàn, cô lập để thử nghiệm (ví dụ: phân tích malware, kiểm thử các dịch vụ mạng client-server) mà không sợ ảnh hưởng đến mạng bên ngoài.