# Các bước đọc ổ đĩa dạng NTFS

## Bước 1: Lấy file cần đọc

* Đọc volume hệ thống tương ứng trong máy tính theo định dạng tên file truyền vào là [\\\\.\\X:](file:///\\\\.\\X:) với X là tên của ổ đĩa cần đọc.
* Chương trình cần phải chạy với phân quyền người quản trị để đọc được các thông tin ổ đĩa.
* Nếu không tồn tại thì thoát chương trình.

## Bước 2: Đọc Boot Sector

* Đọc 512-byte đầu tương ứng với VBR (Volume Boot Record)
* Các thông tin chính cần quan tâm:

|  |  |  |
| --- | --- | --- |
| **Offset** | **Size** | **Thông tin trong VBR** |
| 0x3 | 8 | OEM ID (Sẽ là ‘NTFS ’) |
| 0xB | 2 | Bytes/sector |
| 0xD | 1 | Sectors/cluster |
| 0xE | 2 | Số sector dự trữ |
| 0x28 | 8 | Tổng số sector |
| 0x30 | 8 | Cluster bắt đầu của $MFT |
| 0x38 | 8 | Cluster bắt đầu của $MFTMirr |
| 0x40 | 1 | Kích thước của một $MFT record |

* Kích thước của mỗi một MFT record ở trên sẽ là một con số dạng bù 2, và sẽ cần phải lấy 2 lũy thừa với trị tuyệt đối của giá trị trên để có kích thước thực tế.

VD: *(Theo ví dụ tham khảo phần giải đáp của thầy Long)*

Tại 0x40 có là 0xF6, là một con số bù 2 nên quy đổi ra hệ thập phân là -10.

Do đó, kích thước là (byte)

* Để cho thuận tiện, cluster bắt đầu của MFT đầu tiên nên được quy đổi ra byte: (Cluster bắt đầu) \* (bytes/sector) \* (sectors/cluster)

## Bước 3: Đọc MFT

* Đưa con trỏ đọc đến vị trí bắt đầu đã tính ở trên.
* Đọc MFT đầu tiên để lưu giá trị tổng số sector để lưu các MFT record.  
  (Tại offset 0x118, độ dài là 8)
* Tính tổng số MFT record mà có trong ổ đĩa.  
  (Lấy tổng số sector đó chia cho sectors/record, thường mỗi record sẽ chiếm 1024 bytes)
* Lần lượt đọc qua các MFT record tiếp theo.

## Bước 4: Đọc các MFT record header

* Quá trình đọc MFT record cần chú ý đọc qua 4 phần chính, phần header, attribute Standard Information, attribute File Name, attribute Data.
* Ở phần đầu tiên là header

|  |  |  |
| --- | --- | --- |
| **Offset** | **Size** | **Thông tin của MFT record header** |
| 0x0 | 4 | Dấu hiệu nhận biết MFT entry.  - “FILE”: MFT record bình thường.  - “BAAD”: MFT record lỗi. |
| 0x16 | 2 | Giá trị cờ báo  - 0x01: MFT record của một file.  - 0x02: MFT record của một thư mục. |
| 0x2C | 4 | Số hiệu của MFT record này. |
| 0x14 | 2 | Vị trí bắt đầu của attribute đầu tiên (Standard Information) |

* Các attribute sẽ được sắp xếp liên tiếp nhau nên vị trí bắt đầu của attribute kế tiếp sẽ là kết thúc của attribute hiện tại.
* Trong từng các attribute đều có phần header riêng của mình, các thông tin cần chú ý ở đây là:

|  |  |  |
| --- | --- | --- |
| **Offset** | **Size** | **Thông tin trong attribute header** |
| 0x0 | 4 | Mã của attribute |
| 0x4 | 4 | Kích thước của attribute này |
| 0x8 | 1 | Cờ báo giá trị resident  - 0x00: Resident  - 0x01: Non-resident |

## Bước 5: Standard Information

* Mã attribute: 0x10
* Attribute này sẽ luôn là resident.
* Chứa các thông tin liên quan đến thời gian, phiên bản, phân quyền, …

|  |  |  |
| --- | --- | --- |
| **Offset** | **Size** | **Thông tin** |
| 0x20 | 4 | Thông tin phân quyền |

* Các phân quyền cần lưu ý:

|  |  |
| --- | --- |
| **Giá trị** | **Thông tin** |
| 0x0001 | Chỉ đọc (read only) |
| 0x0002 | Ẩn (hidden) |
| 0x0004 | Hệ thống (system) |
| Ngoài ra còn các thông tin khác nữa. | |

Một file có thể có nhiều quyền, cần phải xem coi bit nào bật (phép ‘and’) để tương ứng với quyền đó (không nên dùng phép so sánh bằng).

## Bước 6: File Name

* Mã attribute: 0x30
* Attribute này sẽ luôn là resident.
* Chứa các thông tin liên quan đến tên của file, số hiệu của file cha, phân quyền, …

|  |  |  |
| --- | --- | --- |
| **Offset** | **Size** | **Thông tin** |
| 0x00 | 6 | Số hiệu của MFT record cha |
| 0x38 | 4 | Cờ báo phân quyền  Tương tự như attribute Standard Information, File Name sẽ có thêm:  - 0x10000000: thư mục (directory) |
| 0x40 | 1 | Độ dài của tên file (L) |
| 0x41 | 1 | Namespace của tên file, gồm các giá trị:  - 0: POSIX  - 1: Win32  - 2: DOS  - 3: Win32 & DOS  Từng định dạng namespace khác nhau sẽ có quy định về cách đặt tên và các kí tự đặc biệt khác nhau. |
| 0x42 | 2L | Tên – được định dạng theo chuẩn Unicode (UTF-16le)  (Một kí tự UTF-16 sẽ chiếm 2 bytes) |

* Ngoài ra, có thể sẽ có nhiều hơn một attribute File Name, điều này xảy ra khi namespace của tên file theo chuẩn MS DOS (chuẩn 8.3, còn gọi là tên ngắn).
* Tên ngắn sẽ được lưu chỉ 8 kí tự đầu của tên file trước dấu chấm, 3 kí tự sau dấu chấm, phần còn lại sẽ được quy đổi bằng cách thêm ~1 vào cuối. Nếu có một file khác có phần giống nhau tương tự, chương trình sẽ tự tăng lên thành ~2 và tương tự cho ~3, ~4, …
* Hệ thống sẽ nếu lưu tên ngắn sẽ thường đi kèm với tên dài sau đó, nằm trong 1 attribute File Name khác sau lưng attribute File Name này.

## Bước 7: Data

* Mã attribute: 0x80
* Attribute này sẽ là resident khi dữ liệu nó chứa ít hơn 700B, sẽ là non-resident khi nhiều hơn.
* Chứa các dữ liệu mà file lưu trữ (là 0 đối với thư mục)
* Nếu MFT record này là resident:

|  |  |  |
| --- | --- | --- |
| **Offset** | **Size** | **Thông tin** |
| 0x10 | 4 | Kích thước phần nội dụng (N) |
| 0x14 | 2 | Offset bắt đầu nội dung (S) |
| S | N | Nội dung |

* Nếu MFT record này là non-resident:

|  |  |  |
| --- | --- | --- |
| **Offset** | **Size** | **Thông tin** |
| 0x30 | 8 | Kích thước phần nội dụng (N) |
| 0x40 | - | Các data-runs (đề cập ở phần tiếp theo) |

## Bước 8: Xây dựng cây thư mục

* Ở từng MFT record đều có các số hiệu của MFT hiện tại và MFT cha.
* Từ đó có thể xác định thứ tự sắp xếp của các file, folder.
* Ngoài ra, có thể xác định được vị trí của MFT record gốc dựa trên mục số hiệu và số hiệu của cha là giống nhau, hoặc là đọc dựa trên attribute Index Root (mã attribute 0x90).

# Cách đọc các data-runs

* Các data-run được cấu trúc gồm 3 phần: Header – Length – Offset.
* Các data-run được xếp liền kề nhau, và phía sau data-run cuối cùng sẽ là 0x00, thông báo không còn data-run nào khác nữa.

Header có dạng 0xXY với:

* Y: số byte tiếp theo là của phần length (bắt đầu từ sau header)
* X: số byte tiếp theo của phần offset (bắt đầu sau phần length)

Ví dụ, giả sử chúng ta có các dãy byte sau:

11 03 07 32 AD 11 4C 1A 01 00 00 ….

Ở đây chúng ta có 2 data-runs:

* Data-run 1:
  + 1 byte cho phần length, 03h = 3 clusters.
  + 1 byte là phần offset, bắt đầu tại cluster thứ 07h = 7.
* Data-run 2:
  + 2 bytes cho phần length, AD 11, 11ADh = 4525 clusters.
  + 3 bytes offset, bắt đầu tại cluster 4C 1A 01, 011A4Ch = 72268.

Phía sau data-run thứ 2 là 00, tức là không còn data-run nào khác nữa.

# Các điểm hạn chế và hướng cải thiện

* Ngôn ngữ lập trình C, C++ phiên bản 2017 vẫn chưa hỗ trợ các kí tự Unicode nên khi in ra có thể sẽ xảy ra lỗi (Xuất hiện các kí tự khác).
* Chương trình cần đọc tất cả các Entry (đối với Fat32) và các MFT record (NTFS) nên khi ổ đĩa quá to, sẽ tốn một ít thời gian để chương trình có thể đọc xong.
* Đọc một cách Dynamic

# Tài liệu tham khảo

<https://flatcap.github.io/linux-ntfs/ntfs>

<https://sabercomlogica.com/en/ntfs-mft-metadata-files/>

<https://legiacong.blogspot.com/>