

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN
HỆ ĐIỀU HÀNH**

**ĐỀ TÀI
QUẢN LÝ HỆ THỐNG TẬP TIN
TRÊN WINDOWS**

NỘI DUNG BÁO CÁO

I.	THÔNG TIN NHÓM	3
II.	BẢNG PHÂN CÔNG CÔNG VIỆC.....	3
III.	ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH	4
1.	FAT32	4
2.	NTFS	4
IV.	CÁC BƯỚC THỰC HIỆN	5
1.	Đối với hệ thống tập tin FAT32.....	5
a)	Đọc Boot Sector	5
b)	Đọc bảng FAT.....	6
c)	Đọc RDET (Bảng thư mục gốc).....	7
d)	Đọc SDET (Bảng thư mục con).....	7
e)	Đọc cây thư mục.....	7
f)	Đọc DATA và Cây thư mục con	8
2.	Đối với hệ thống tập tin NTFS	8
a)	Đọc Partition Boot Sector (Volume Boot Record).	8
b)	Đọc vị trí Master File Table (MTF).	9
c)	Đọc MTF entry (MTF record) và Attribute.	10
d)	Đọc Index block và các Index Entry	11
e)	Đọc cây thư mục (B-Tree).	12
f)	Đọc DATA và Cây thư mục con.	13
V.	DEMO CHƯƠNG TRÌNH.....	15
1.	FAT32	15
2.	NTFS.....	21
VI.	NGUỒN KHAM KHẢO	27

I. THÔNG TIN NHÓM

STT	Họ Và Tên	MSSV
1	Hoàng Anh Quân	19120628
2	Nguyễn Vạn Thịnh	19120668
3	Mai Hồng Phúc	19120620
4	Nguyễn Hoàng Thái	19120650
5	Hoàng Văn Tiệp	19120443

II. BẢNG PHÂN CÔNG CÔNG VIỆC

STT	Họ Và Tên	Nội dung công việc	Hoàn Thành
1	Hoàng Anh Quân	<ul style="list-style-type: none">- Tổ chức cấu trúc chung và tổng hợp mã nguồn cho project.- Đọc MFT entry và các Attribute của NTFS- Cấu trúc lưu trữ cho cây thư mục.- Viết báo cáo phần demo chương trình	100%
2	Nguyễn Vạn Thịnh	<ul style="list-style-type: none">- Tổ chức cấu trúc dữ liệu của hệ thống FAT32- Đọc FAT và Các Entry và cây thư mục cho FAT32- Viết báo cáo mô tả các bước thực hiện FAT32- Tổng hợp mã nguồn hoàn thành phần FAT32	100%
3	Mai Hồng Phúc	<ul style="list-style-type: none">- Tổ chức cấu trúc dữ liệu của hệ thống NTFS- Đọc Endex Block và Index entry của NTFS- Tổng hợp mã nguồn hoàn thành phần NTFS- Viết báo cáo mô tả các bước thực hiện NTFS	100%
4	Nguyễn Hoàng Thái	<ul style="list-style-type: none">- Đọc cây thư mục cho phần hệ thống FAT32- Đọc Data của một file txt cho FAT32- Kiểm thử chương trình.- Hoàn thành nội dung báo cáo.	100%
5	Hoàng Văn Tiệp	<ul style="list-style-type: none">- Đọc Boot sector của NTFS và FAT32- Đọc cây thư mục và Data cho tập tin cho NTFS- Thiết kế cài đặt giao diện cho chương trình- Tổng hợp báo cáo.	100%

III. ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH

1. FAT32

- Đọc thông tin chi tiết của phần vùng (Boot Sector) (100%)
- Hiển thị cây thư mục của phần vùng (100%)
- Hiển thị tên tập tin / thư mục, trạng thái, kích thước, chỉ số sector(100%).
- Truy xuất thông tin trên cây thư mục và đọc tập tin có phần mở rộng txt(100%)
- Truy xuất đến thư mục thì hiển thị cây thư mục con(100%)

2. NTFS

- Đọc thông tin chi tiết của phần vùng (Partition Boot Sector) (100%)
- Hiển thị cây thư mục của phần vùng (100%)
- Hiển thị tên tập tin / thư mục, trạng thái, kích thước, chỉ số sector(100%).
- Truy xuất thông tin trên cây thư mục và đọc tập tin có phần mở rộng txt(100%)
- Truy xuất đến thư mục thì hiển thị cây thư mục con(100%)

⇒ **Mức độ hoàn thành yêu cầu trên toàn bộ project: 100%**

IV. CÁC BƯỚC THỰC HIỆN

Ý tưởng đọc dữ liệu trên C++:

Để đọc thông tin một cấu trúc n byte ví dụ: Boot sector 512byte. Ta tạo ra các cấu trúc tương ứng với số byte đó, với các thông tin được sắp xếp theo từng byte.

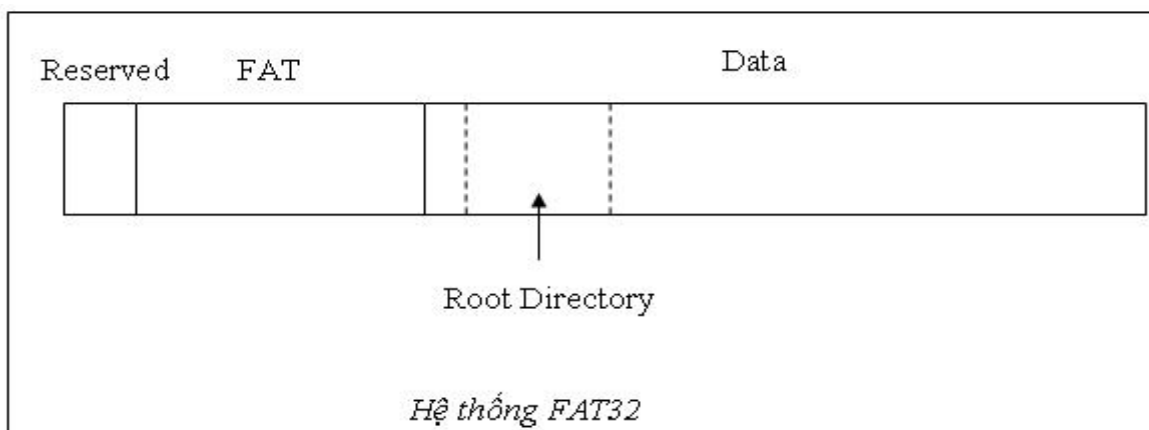
Khi đó ta chỉ cần đọc theo dạng nhị phân để đọc toàn bộ cấu trúc thay vì phải đọc riêng lẻ từng phần

Lưu ý: Vị trí của các sector được đọc trong chương trình bắt đầu tại offset 0 của một ổ đĩa logic.

(Tổ chức source code và chức năng của từng file đã nói rõ trong file **README.TXT**)

1. Đối với hệ thống tập tin FAT32

Để bắt đầu thao tác với ổ đĩa FAT32 ta cần biết tổ chức của ổ đĩa logic như sau:



a) Đọc Boot Sector

- Boot Sector là sector đầu tiên mô tả thông tin chi tiết phân vùng
- Ta tra bảng BootSector để xác định loại FAT, Số byte cho 1 sector, số sector cho 1 cluster (Sc), số sector vùng Bootsector (Sb), số bảng FAT, số sector cho RDET, số entry của cây thư mục gốc.
- Xác định được tổng sector trên một volume hay còn gọi kích thước của một đĩa và các sector đầu tiên của các vùng.

Các thuộc tính của boot sector được sử dụng trong chương trình:

Tên thuộc tính	Offset	Size (Bytes)	Ý nghĩa
BS_OEMName	3	8	Tên hoặc phiên bản HDH

BytesPerSec	B	2	Số bytes/sector
SecPerClus	D	1	Số sectors/cluster
RsvdSecCnt	E	1	Số sector trước bảng FAT
NumFATS	10	2	Số bảng FAT
SecPerTrk	18	2	Số sectors/track
NumHeads	1A	2	Số heads
HiddSec	1C	2	Số sector ẩn trước Volume
TotSec32	20	4	Số sector trong Volume
FATSz32	24	4	Số sector trong 1 bảng FAT
RootClus	2C	4	Chỉ số cluster đầu tiên của RDET
BootSig	42	1	Ký hiệu nhận biết HDH
VolLab	47	11	Volume label
FilSysType	52	8	Chuỗi nhận diện loại FAT

Dựa vào cấu trúc trên, ta xây dựng 1 struct tên là Fat32BootSector trong file FAT32_BS.h
Ta tạo 1 class FAT32_BS để thao tác với các thuộc tính trên cấu trúc đó

b) Đọc bảng FAT

Để bắt đầu đọc Volume của FAT ta cần biết được cấu trúc của bảng FAT được tổ chức ra sao:

- FAT là nơi chứa thông tin về trạng thái của các cluster trong vùng dữ liệu.
- Bảng FAT sử dụng cấu trúc dữ liệu là danh sách liên kết để lưu trữ, nên ta hàm viết hàm readListSectorFAT() trong class **VOLUME_FAT** dùng để đọc thông tin FAT cũng là một danh sách liên kết.
- Đối với FAT32: mỗi entry có 32 bit (4 byte) và đại diện cho 1 cluster

Ví dụ:

Giá trị	03 FF F0 FF	CD AB F7 EF	7F FF 00 40	FF F0 03 FF	00 40 7F FF
Phần tử FAT	0	1	2	3	4

- Vì vậy khi tiến hành đọc thông tin FAT ta sẽ đọc từng 4 byte 1 trong source code

- Để xác định thông tin cluster ta cần nắm được trạng thái của cluster :

Trạng thái	FAT32
Trống	0
Hư	0FFFFFFF=BAD
Cluster cuối file	0FFFFFFF= EOF
Chứa nội dung file	2..0FFFFFFEF

Sau khi xác định các trạng thái trên ta sẽ tiến hành đọc danh sách cluster của RDET theo bằng việc xác định cluster đầu tiên trong bảng BSector và thực hiện tra trên bảng FAT

c) Đọc RDET (Bảng thư mục gốc)

- Ta tìm đến các sector chứa nội dung của RDET đã được xác định từ trước
- Để đọc thông tin cây thư mục của phân vùng ta cần đọc và phân tích bảng RDET
- RDET là nơi tổ chức các tập tin trên volume nằm trên vùng dữ liệu (FAT32)
- Gồm một dãy các phần tử (gọi là entry), mỗi phần tử có kích thước 32 bytes chứa các thông tin của 1 tập tin hoặc một thư mục
- Có 2 loại entry là entry chính và entry phụ
- Cấu trúc của sub entry và main entry sử dụng trong source là: **MAIN_ENTRY** và **SUB_ENTRY (DirEntry.h)**, sử dụng 1 class gộp chung thành một Entry: **DIR_ENTRY**

d) Đọc SDET (Bảng thư mục con)

- Đối tượng thư mục con được xác định trong thư mục gốc ta biết được cluster bắt đầu của nó, sau đó tra cluster bắt đầu để di chuyển đến các sector của bảng SDET, và đọc các entry giống với RDET, và tiếp tục đọc như vậy cho đến nút lá

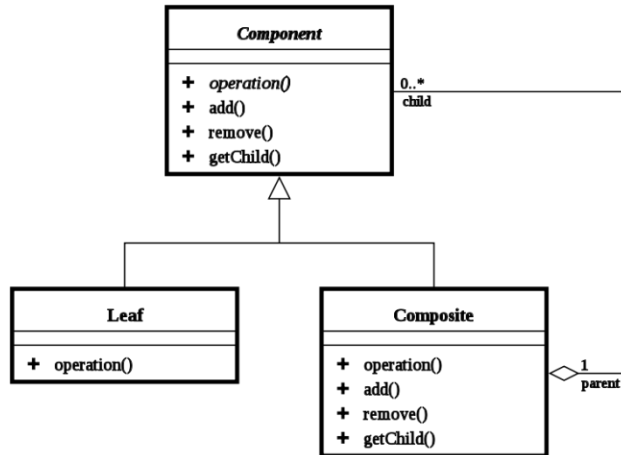
e) Đọc cây thư mục

- Sau khi đọc toàn bộ entry trên thư mục gốc và thư mục con, Ta xây dựng cấu trúc cây thư mục để lưu trữ các entry đó.
- Bằng việc sử dụng design pattern Composite trong lập trình hướng đối tượng giúp quản lý cây thư mục dễ dàng hơn.

Cấu trúc đó được tổ chức trong 3 class (DirEntry.h):

- **DIR_ENTRY** composite
- **FILE_ENTRY**: Component

- **FOLDER_ENTRY**: Leaf



Sau khi thực hiện đủ các bước trên ta đã có nội dung của cây thư mục và tập tin cũng như thông tin về boot sector, và cây thư mục được tổ chức trong class **VOLUME_FAT**

f) Đọc DATA và Cây thư mục con

Ta sẽ tổ chức tìm kiếm các entry dựa vào tên nhận được.

Sẽ có 2 trường hợp xảy ra:

- Nếu là Folder ta sẽ tiến hành in cây thư mục con giống với cây thư mục gốc dựa vào cấu trúc composite đã nêu ở trước
- Nếu là File ta sẽ đọc cluster đầu tiên xác định các cluster còn lại trên bảng FAT, và di chuyển các sector chứa data để đọc nội dung của Files

2. Đối với hệ thống tập tin NTFS

Để bắt đầu với việc thao tác NTFS ta cần phải biết tổ chức của ổ đĩa logic định dạng NTFS:

VBR	MFT	Nội dung của tập tin (loại non-resident)	MFT dự phòng	Chưa sử dụng
-----	-----	---	-----------------	-----------------

Hệ thống đĩa định dạng theo NTFS không có cơ cấu tổ chức cố định như đối với FAT.

a) Đọc Partition Boot Sector (Volume Boot Record).

Partition Boot Sector là bản ghi khởi động của ổ đĩa logic, nó luôn nằm ở vị trí đầu tiên của mỗi ổ đĩa logic.

Trong source code, Nó được biểu diễn bằng một cấu trúc 512byte có tên là **NTFSBootSector** (NTFS_BS.h) được sử dụng cho **VOLUME_NTFS** (VolumeNTFS.h)

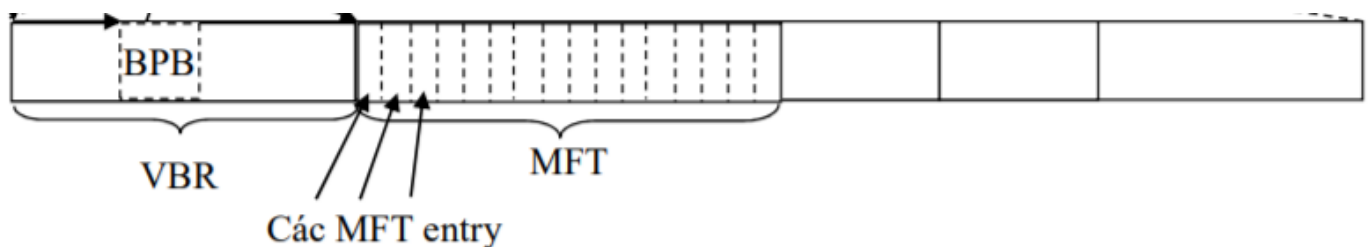
Để đọc các nội dung chính trên VBR ta cần đọc thông tin bảng BPB (Bios Parameter Block) bắt đầu tại offset 0xB tới offset 0x53 trong VBR để xác định các thông tin cần thiết cho yêu cầu project.

Địa chỉ (offset)	Kích thước	Mô tả
0Bh	2	Kích thước một sector. Đơn vị tính là byte.
0Dh	1	Số sector trong một cluster.
18h	2	Số sector/track.
1Ah	2	Số mặt đĩa (head hay side).
1Ch	4	Sector bắt đầu của ổ đĩa logic.
28h	8	Số sector của ổ đĩa logic.
30h	8	Cluster bắt đầu của MFT
38h	8	Cluster bắt đầu của MFT dự phòng (MFTMirror).
40h	1	Kích thước của một bản ghi trong MFT
44h	1	Số cluster của Index Buffer.

Sau khi xác định các thông tin trên ta lưu lại chúng vào class **Volume_NTFS** (VolumeNTFS.h) trên source code

b) Đọc vị trí Master File Table (MTF).

MFT (Master File Table) chứa thông tin về tất cả các tập tin và thư mục trong ổ đĩa logic, nó được xem là điểm bắt đầu để đi đến các tập tin trên một ổ đĩa logic.



Vị trí của MFT entry đọc tại offset 30h trên bảng VBR lưu trong **VOLUME_NTFS**, Ta tiến hành di chuyển đến sector = $\text{SecPerClus} * \text{FstMTF}$ để tiếp tục đọc nội dung trên MFT

MFT được chia nhỏ thành các phần bằng nhau gọi là MFT entry.

Có hai loại MFT entry:

-MFT entry cho tập tin siêu dữ liệu (metadata file): chứa các thông tin mô tả về cấu trúc tổ chức của hệ thống NTFS. Ví dụ: \$MFT,\$MFTMirro, \$volume ,DOT(,),..

- MFT entry cho tập tin dữ liệu (data file): chứa thông tin mô tả cho các tập tin dữ liệu, có thể chứa cả phần dữ liệu của tập tin

c) Đọc MTF entry (MTF record) và Attribute.

+MTF record

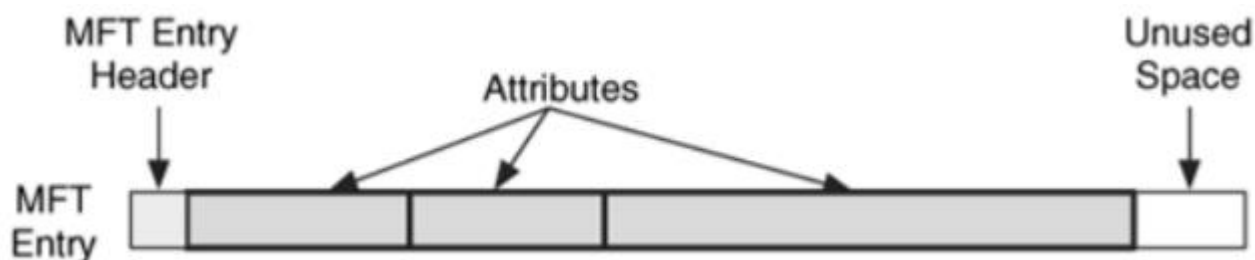
MTF record có kích thước cố định và bằng nhau thường là 1024 byte.

Các MFT record được đánh số thứ tự để xác định gọi là Index Tree được bắt đầu tại 0 (\$MFT), tại sector đầu tiên trên bảng MFT

Để đọc cây thực mục của ô đĩa. Ta cần phải bắt đầu ở thư mục gốc tại MFT record số 5 gọi là DOT entry.

Cấu trúc của MFT record:

MFT record gồm hai thành phần: header và các attribute.



Cấu trúc của MFT header được tổ chức tại struct **MFT_ENTRY_HEADER** (EntryType.h) gồm 42 byte. Trong chương trình ta chỉ cần quan tâm đến các trường cần thiết sau:

Địa chỉ (offset)	Kích thước	Mô tả
0x0	4	Dấu hiệu nhận biết MFT entry (FILE)
0x04	2	Địa chỉ (offset) của Update sequence.
0x06	2	Số phần tử của mảng Fixup
0x14	2	Địa chỉ (offset) bắt đầu của các attribute.

0x18	4	Số byte đã được sử dụng trong MFT entry.
0x28	2	Next attribute ID: mã định danh của attribute kế tiếp sẽ được thêm vào MFT entry.

Sau khi đọc xong phần header của MFT record, ta sẽ đọc lần lượt các Attribute.

+ **Attribute:**

- Attribute là một cấu trúc dữ liệu, được sử dụng để chứa nội dung của tập tin, chứa các thông tin liên quan đến tập tin, thư mục,...v.v trong hệ thống NTFS.
- Có nhiều loại Attribute, nhưng trong chương trình ta đọc 4 loại chính là:
\$FILE_NAME - \$INDEX_ROOT - \$INDEX_ALLOCATION - \$DATA
- Mỗi Attribute gồm phần header và nội dung của attribute

Đọc Attribute Header

- Được cấu trúc trong struct **MFT_HEADER_ATTRIBUTE (EntryType.h)**

Sau khi đọc phần header ta sẽ xác định được kích thước của Attribute, và vị trí của nội dung Attribute.

Đọc Attribute Content

- \$FILE_NAME: Được cài đặt trong struct **MFT_ATTR_FILENAME** và được tổ chức trong class **FILENAME_ATTRI(EntryType.h)** để biểu diễn tên unicode, ngày giờ, kiểu file
- \$INDEX_ROOT: Được cài đặt trong struct **MFT_ATTR_INDEX_ROOT**, được tổ chức trong class **INDEX_ROOT_ATTRI(EntryType.h)** Chứa nút gốc (root node) của cây chỉ mục (index tree).
- \$INDEX_ALLOCATION : được tổ chức trong class **INDEX_ALLOC_ATTRI (EntryType.h)** Chứa các nút của cây chỉ mục (index tree) nếu \$INDEX ROOT không đủ lưu trữ.
- \$Data: được tổ chức trong class **DATA_ATTRIBUTE (EntryType.h)**. Chứa nội dung của File.

Attribute được tổ chức thành resident và non-resident để giải quyết việc lưu trữ trên MFT

Sau khi đọc các thông tin trên, ta sẽ xác định được các thông tin cần thiết để truy xuất đến các vị trí của entry khác nằm trên \$INDEX_ROOT và \$INDEX_ALLOCATION.

d) Đọc Index block và các Index Entry

- Đây là một nút con của cây thư mục hiện một chỉ mục (ví dụ: một thư mục)

- Vị trí sector của Index block được đọc trong datarun của \$INDEX_ALLOCATION
- Index block bao gồm header và Index entry

Index block header

- Header của Index block được lưu trữ trong struct **INDEX_BLOCK** (EntryType.h)

```

struct INDEX_BLOCK
{
    // Index Block Header
    DWORD    Magic;           // "INDX"
    WORD     OffsetOfUS;      // Offset of Update Sequence
    WORD     SizeOfUS;        // Size in words of Update Sequence Number & Array
    ULONGLONG LSN;           // $LogFile Sequence Number
    ULONGLONG VCN;           // VCN of this index block in the index allocation
    // Index Header
    DWORD    EntryOffset;     // Offset of the index entries, relative to this address(0x18)
    DWORD    TotalEntrySize;  // Total size of the index entries
    DWORD    AllocEntrySize;  // Allocated size of index entries
    BYTE     NotLeaf;         // 1 if not leaf node (has children)
    BYTE     Padding[3];      // Padding
};

```

- Sau khi đọc xong phần header ta sẽ xác định được offset của Index entry

Đọc Index Entry

49 4E 44 58 28 00 09 00	73 DB 40 00 00 00 00 00	INDX (sŪ@
00 00 00 00 00 00 00 00	28 00 00 00 80 05 00 00	(€
E8 0F 00 00 00 00 00 00	03 00 68 00 D7 01 00 00	è h x
00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
5D 00 00 00 00 00 01 00	90 00 7A 00 00 00 00 00] z
5C 00 00 00 00 00 01 00	E6 14 FB 3F 26 CD D7 01	\ æ û?&íx
00 58 45 68 0C 19 CE 01	D0 CC EC 5B B0 2D D7 01	XEh î ðìì[°-x
E6 DF ED 07 8C CD D7 01	00 80 07 00 00 00 00 00	æßí ŒÍx €
C6 77 07 00 00 00 00 00	20 00 00 00 00 00 00 00	Æw
1C 00 43 00 54 00 54 00	31 00 30 00 32 00 2D 00	C T T 1 0 2 -
43 00 68 00 75 00 6F 00	6E 00 67 00 20 00 31 00	C h u o n g 1
2D 00 54 00 6F 00 6E 00	67 00 51 00 75 00 61 00	- T o n g Q u a
6E 00 2E 00 70 00 64 00	66 00 00 00 00 00 00 00	n . p d f

- Index Entry để xác định một \$FILE_NAME và Index tree
- Nó còn được đặt \$INDEX_ROOT nếu không tồn tại \$ALLOCATION_INDEX
- Sau khi đọc xong Index Entry. Ta sẽ lưu chúng và cây thư mục được tổ chức trong class **ENTRY_NTFS** (volumeNTFS.h)

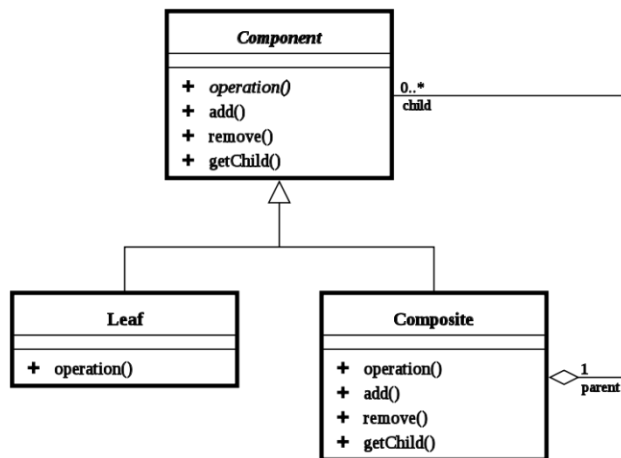
e) Đọc cây thư mục (B-Tree).

Sau khi xác định được Index tree được đọc ở Index Entry. Ta sẽ sử dụng cấu trúc đọc cây B-tree để tiếp tục tìm đến MFT entry ứng với Index tree của từng thư mục, tập tin. Ta thực hiện đọc như vậy đến node lá.

Trong source code, ta sử dụng cấu trúc Composite, giúp việc lưu trữ và tổ chức cây thư mục trở nên dễ dàng hơn.

Cấu trúc đó được tổ chức trong 3 class (VolumeNTFS.h):

- **FILE_NTFS**: composite
- **FOLDER_NTFS**: Component
- **FILE_NTFS**: Leaf



Sau khi thực hiện đủ các bước trên ta đã có nội dung của cây thư mục và tập tin cũng như thông tin về boot sector, và cây thư mục được tổ chức trong class **VOLUME_NTFS**

f) Đọc DATA và Cây thư mục con.

DATA là nội dung của một tập tin.

Đối với các tập tin có kích thước nhỏ, Nó sẽ lưu trữ trực tiếp trên \$DATA của MFT entry (resident)

Đối với tập tin lớn hơn, \$DATA sẽ cung cấp cho ta một datarun (vị trí của phần data) để chỉ đến Nội dung của tập tin đây (non-resident)

Để đọc Data hoặc cây thư mục con, ta sẽ tổ chức tìm kiếm các entry dựa vào tên nhận được.

Sẽ có 2 trường hợp xảy ra:

+ Nếu là Folder ta sẽ tiến hành in cây thư mục con giống với cây thư mục gốc dựa vào cấu trúc composite đã nêu ở trước

+ Nếu là File ta sẽ xác định File đã lưu trữ data theo kiểu nào:

Nếu là resident thì đọc trực tiếp trong Attribute \$DATA

Nếu là Non-resident ta sẽ xác định datarun Attribute \$DATA để tìm các sector chứa data của file cần đọc

⇒ ***Kết hợp lại các thông tin trên ta sẽ biết cách để thực hiện tổ chức cây thư mục và đọc dữ liệu của tập tin trên chương trình máy tính***

V. DEMO CHƯƠNG TRÌNH

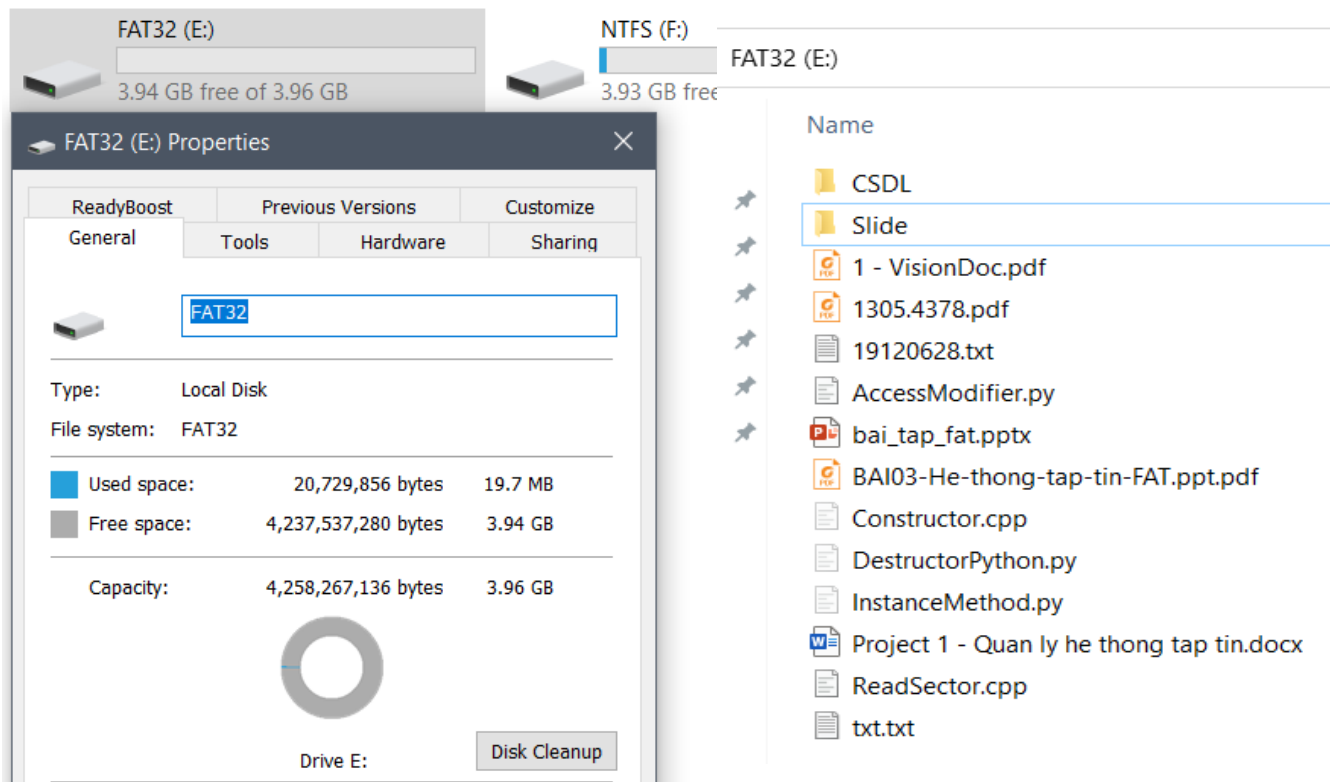
Có 2 cách để chạy chương trình

- +Có thể mở visual studio để chạy code
- +Chạy file System_File.exe trong thư mục Release

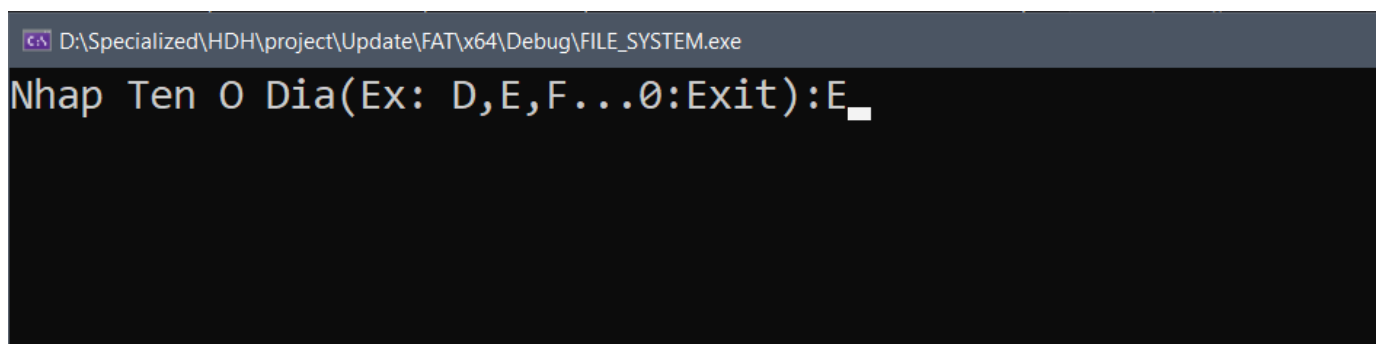
Lưu ý: Phải chạy chương trình bằng quyền quản trị (Run as administrator)

1. FAT32

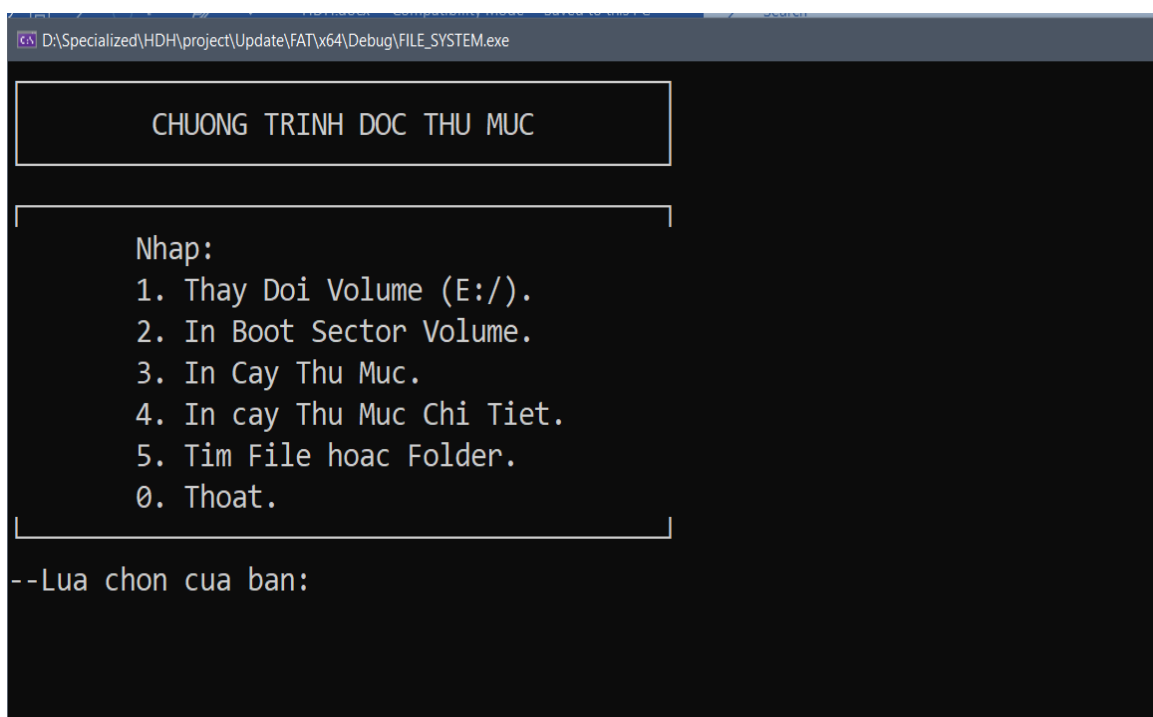
Chọn ổ đĩa E:/ có File System là FAT32 để kiểm thử.



B1: Sau khi mở chương trình lên giao diện sẽ hiển thị như hình dưới. Nhập tên ổ đĩa cần xem (nhập E để demo)

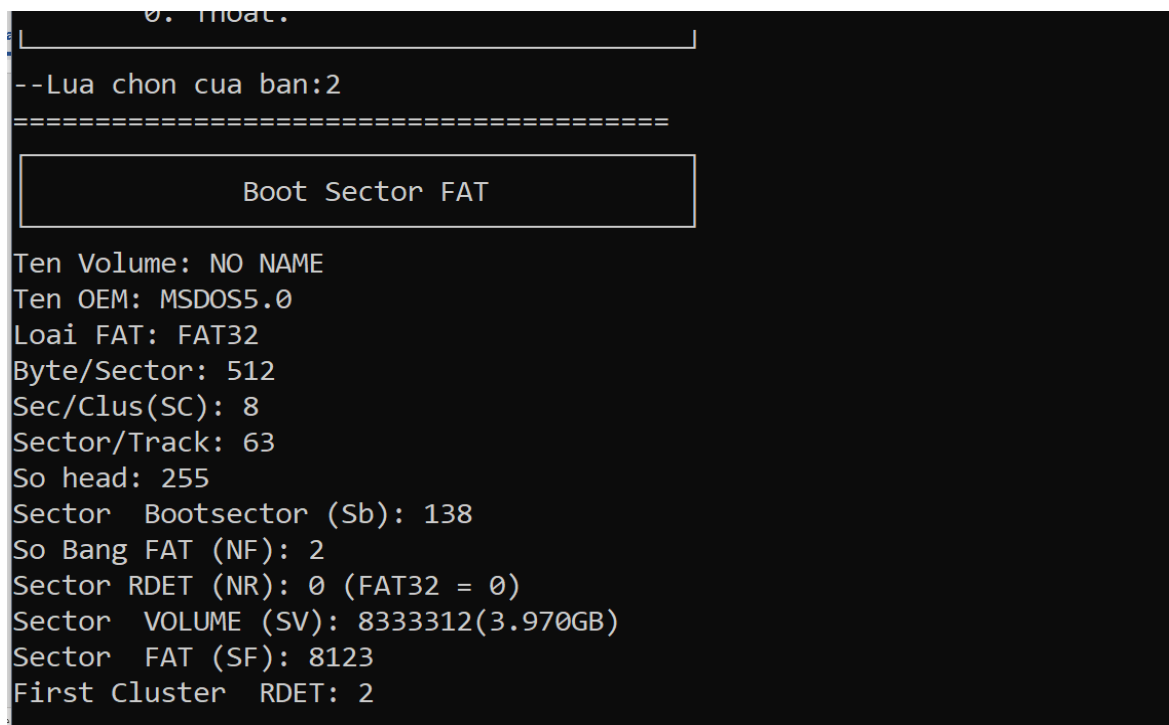


Giao diện bảng chọn được hiển thị sau khi chọn ổ đĩa:



Nếu bạn muốn thay đổi volume khác có thể nhập số 1.

B2: Nhập 2 để xem thông tin Boot sector, sau đó nhấn phím bất kỳ để trở lại menu



B3 :Nhập 3 để xem cây thư mục.

```
--Lua chon cua ban:3
=====
Cay Thu Muc

LE:
  LSystem Volume Information:
    | WPSettings.dat(12B)
    | IndexerVolumeGuid(76B)
    | EDP:
      | Recovery:
        | 1305.4378.pdf(1.360MB)
        | 19120628.TXT(1.360KB)
        | Constructor.cpp(461B)
        | AccessModifier.py(516B)
        | BAI03-He-thong-tap-tin-FAT.ppt.pdf(1.100MB)
        | DestructorPython.py(427B)
        | InstanceMethod.py(293B)
        | Project 1 - Quan ly he thong tap tin.docx(21.07KB)
    | $RECYCLEBIN:
      | DESKTOP.INI(129B)
      | $IAYHXLL.TXT(60B)
      | $IQ9V2YK.PDF(60B)
      | $I6WB3B9.PDF(62B)
      | $IBJSSFO.CPP(66B)
      | $I7E0V9K.PDF(114B)
      | $I3G4PT0.PDF(146B)
      | $IMJFM3D.PDF(128B)
      | $IAFZW4.SQL(80B)
      | $IVC3BCT.SQL(80B)
      | ReadSector.cpp(5B)
      | TXT.TXT(0B)
    | CSDL:
      | demo_tvlong.sql(1.130KB)
      | Quanlychuyenbay.txt(27.63KB)
      | quanlygiaovien.sql(47.84KB)
    | Mo ta CSDL:
      | QuanLyChuyenBay:
        | Quan Ly CHUYEN BAY.pdf(383.6KB)
      | QuanLyDeTai:
        | 19120628_QLDT.sql(1.160KB)
        | Quan Ly DE TAI - Bai tap.pdf(179.6KB)
        | Quan ly DE TAI - Mo ta du lieu.pdf(351.7KB)
    | Slide:
      | CTT102-Chuong 7-RBTU.pdf(278.9KB)
```

1

```
Project 1 - Quan ly he thong tap tin.docx(21.07KB)
$RECYCLEBIN:
  | DESKTOP.INI(129B)
  | $IAYHXLL.TXT(60B)
  | $IQ9V2YK.PDF(60B)
  | $I6WB3B9.PDF(62B)
  | $IBJSSFO.CPP(66B)
  | $I7E0V9K.PDF(114B)
  | $I3G4PT0.PDF(146B)
  | $IMJFM3D.PDF(128B)
  | $IAFZW4.SQL(80B)
  | $IVC3BCT.SQL(80B)
  | ReadSector.cpp(5B)
  | TXT.TXT(0B)
CSDL:
  | demo_tvlong.sql(1.130KB)
  | Quanlychuyenbay.txt(27.63KB)
  | quanlygiaovien.sql(47.84KB)
Mo ta CSDL:
  | QuanLyChuyenBay:
    | Quan Ly CHUYEN BAY.pdf(383.6KB)
  | QuanLyDeTai:
    | 19120628_QLDT.sql(1.160KB)
    | Quan Ly DE TAI - Bai tap.pdf(179.6KB)
    | Quan ly DE TAI - Mo ta du lieu.pdf(351.7KB)
Slide:
  | CTT102-Chuong 7-RBTU.pdf(278.9KB)
```

2

```
Quan Ly DE TAI - Bai tap.pdf(179.6KB)
Quan ly DE TAI - Mo ta du lieu.pdf(351.7KB)
Slide:
  | CTT102-Chuong 7-RBTU.pdf(278.9KB)
  | CTT102-Chuong 8-PTH&DC.pdf(503.4KB)
  | CTT102-Chuong 9- Cac van de lien quan.pdf(1.220MB)
ThucHanh:
  | Bai huong dan va bai tap:
    | CTT102-Topic01- Huong dan cai dat va su dung SQL Server .pdf(3.230MB)
    | CTT102-Topic02-Dinh nghia co so du lieu.pdf(1.110MB)
    | CTT102-Topic03-Truy van don gian.pdf(673.4KB)
    | CTT102-Topic04-Ham ket hop, gom nhom .pdf(300.5KB)
    | CTT102-Topic05-Truy van long .pdf(426.5KB)
    | CTT102-Topic06-Truy van long nang cao.pdf(418.3KB)
    | CTT102-Topic07-Mot so loai truy van khac.pdf(1.180MB)
    | CTT102-Topic08-Stored Procedure va Function.pdf(2.150MB)
    | CTT102-Topic09-RBTU va Trigger.pdf(411.2KB)
    | password is.fit.hcmus.vn.txt(15B)
  | Tuan02:
    | 19120628_HoangAnhQuan_BTUN2.rar(4.780KB)
    | 19120628_QLCB.sql(6.470KB)
    | 19120628_QLDT.sql(8.900KB)
  | Tuan03:
    | 19120628_HoangAnhQuan_BTTL3.sql(5.230KB)
    | 19120628_HoangAnhQuan_BTUN3.sql(2.780KB)
  | 19120628_QLCB.sql(10.07KB)
  | 19120628_QLDT.sql(10.70KB)
```

3

B4 :Nhập 4 để xem chi tiết cây thư mục.

Hình dưới mô tả một phần hiển thị cây thư mục chi tiết.

```
--Lua chon cua ban:4
=====
Cay Thu Muc Chi Tiet
|
E
Attribute:
First Cluster: 2
|
System Volume Information
Attribute: Hidden System Folder
First Cluster: 3
|
Name: WPSettings.dat
Attribute: Flie
Create: -Date:2021/10/11 -Time:9:30:22
Access: -Date: 2021/11/13
Modified: -Date: 2021/10/11 -Time:9:30:24
FileSize: 12 Byte
First Cluster: 4
-----
```

```
Name: 1305.4378.pdf
Attribute: Flie
Create: -Date:2021/10/20 -Time:20:14:22
Access: -Date: 2021/10/29
Modified: -Date: 2021/10/7 -Time:10:0:44
FileSize: 1431041 Byte
First Cluster: 80724
-----
Name: 19120628.TXT
Attribute: Flie
Create: -Date:2021/10/18 -Time:18:43:10
Access: -Date: 2021/11/16
Modified: -Date: 2021/10/7 -Time:18:33:52
FileSize: 1402 Byte
First Cluster: 65963
-----
Name: Constructor.cpp
Attribute: ReadOnly Hidden Flie
Create: -Date:2021/10/18 -Time:18:43:10
Access: -Date: 2021/10/30
Modified: -Date: 2021/9/29 -Time:23:38:58
FileSize: 461 Byte
First Cluster: 66246
-----
```

```
Slide
Attribute: Folder
First Cluster: 66515
|
Name: CTT102-Chuong 7-RBTV.pdf
Attribute: Flie
Create: -Date:2021/10/30 -Time:19:45:34
Access: -Date: 2021/10/30
Modified: -Date: 2013/3/5 -Time:2:16:56
FileSize: 285623 Byte
First Cluster: 67558
-----
Name: CTT102-Chuong 8-PTH&DC.pdf
Attribute: Flie
Create: -Date:2021/10/30 -Time:19:45:34
Access: -Date: 2021/10/30
Modified: -Date: 2013/3/5 -Time:2:17:34
FileSize: 515526 Byte
First Cluster: 67628
-----
Name: CTT102-Chuong 9- Cac van de lien quan.pdf
Attribute: Flie
Create: -Date:2021/10/30 -Time:19:45:34
Access: -Date: 2021/10/30
Modified: -Date: 2013/3/5 -Time:2:18:2
FileSize: 1280495 Byte
```

B5 :Nhập 5 để tìm file hoặc Folder

+ Tìm file: 19120628.txt


```
--Lua chon cua ban:5
=====
Nhap Ten Entry:19120628.txt
Name: 19120628.TXT
Attribute: Flie
Create:      -Date:2021/10/18  -Time:18:43:10
Access:      -Date: 2021/11/16
Modified:     -Date: 2021/10/7  -Time:18:33:52
FileSize: 1402 Byte
First Cluster: 65963
-----
-----DATA-----
3/
C)
RDECT: 41
FAT1: 1
FAT2: 21
DATA: tao entry moi tren SDET tai sector: 85
      tao cluster luu tru tai:16,17,18 -> chi biem cdc sector: 101,102,103,104,105,106
d)
- FAT1: 1
- FAT2: 21
- DATA: 81
e)
FAT1: 1
FAT2: 21
DATA: tao entry moi tren SDET tai sector: 85

4/
Ta cd:
SC = 32 sector/cluster
NF = 2
SB=8
SV = 1 GB=1024^3 Byte = 2097152(sector)

Bng ung th m bnc gac chi biem 512 entry = (512*32) / 512 = 32 (sector)

Thay cdc gi i tr b i -a u c vao -a ng th b c: SB + NF*SF + SR + SD = SV
      8 + 2SF + 32 + SD = 2097152 => 2SF + SD = 2097112(sector) (*)
=>SD ~ 2097112/32= 65534,75(v % Sc = 32 sector)
FAT 16 qu b un l t b ai -aa 65517 m a SD~65534,75>65517 =>SB i d b ng FAT32

Gi b u s b i SF = 1 (sector): (*) naa SD = 2097112- 2SF = 2097110(sector) = 65534,6875(cluster)
V ng d b » li b cu c 65535 cluster, n n b ung FAT ph b u c 65535 + 2 = 65537 ph b en t b i, do -a SF =
(65537 * 4) / 512 = 512.X(sector)
SF = 512 sector. M ou thu b n v b i gi b u th i t SF = 1.

Gi b u s b i SF = 512 (sector): (*) naa SD = 2097112- 2SF = 2096088(sector) = 65502,75cluster)
V ng d b » li b cu c 65502 cluster, n n b ung FAT ph b u c 65505 ph b en t b i, do -a SF = 65505 * 4) /
512 = 511.X(sector)
SF = 512 sector. Ph h b up v b i gi b u th i t SF = 512
```

1 - VisionDoc.pdf
1305.4378.pdf
19120628.txt

```
19120628.txt - Notepad
File Edit Format View Help
3/
C)
RDECT: 41
FAT1: 1
FAT2: 21
DATA: tao entry moi tren SDET tai sector: 85
      tao cluster luu tru tai:16,17,18 -> chiem cac sector: 101,102,103,
d)
- FAT1: 1
- FAT2: 21
- DATA: 81
e)
FAT1: 1
FAT2: 21
DATA: tao entry moi tren SDET tai sector: 85
      tao cluster luu tru tai:16,17,18 -> chiem cac sector: 101,102,103,
4/
Ta co:
SC = 32 sector/cluster
NF = 2
SB=8
SV = 1 GB=1024^3 Byte = 2097152(sector)

Bng thur muc goc chiem 512 entry = (512*32) / 512 = 32 (sector)
```

Ln 13, Col 8 100% Windows (CRLF) UTF-8

Phần nội
dung để đối
chiếu

+ Tìm Folder: Mo ta CSDL (E:/CSDL/Mo ta CSDL)

```
--Lua chon cua ban:5
=====
Nhap Ten Entry:Mo ta CSDL

Cay Thu Muc Con

Mo ta CSDL:
└─QuanLyChuyenBay:
    └─Quan Ly CHUYEN BAY.pdf(383.6KB)
└─QuanLyDeTai:
    └─19120628_QLDT.sql(1.160KB)
    └─Quan Ly DE TAI - Bai tap.pdf(179.6KB)
    └─Quan ly DE TAI - Mo ta du lieu.pdf(351.7KB)
=====

Cay Thu Muc Con Chi Tiet
```

1

```
Cay Thu Muc Con Chi Tiet

Mo ta CSDL
Attribute: Folder
First Cluster: 66282

└─QuanLyChuyenBay
Attribute: Folder
First Cluster: 66283

Name: Quan Ly CHUYEN BAY.pdf
Attribute: Flie
Create: -Date:2021/10/30 -Time:19:45:34
Access: -Date: 2021/10/30
Modified: -Date: 2013/3/4 -Time:15:8:22
FileSize: 392861 Byte
First Cluster: 66284
=====
```

2

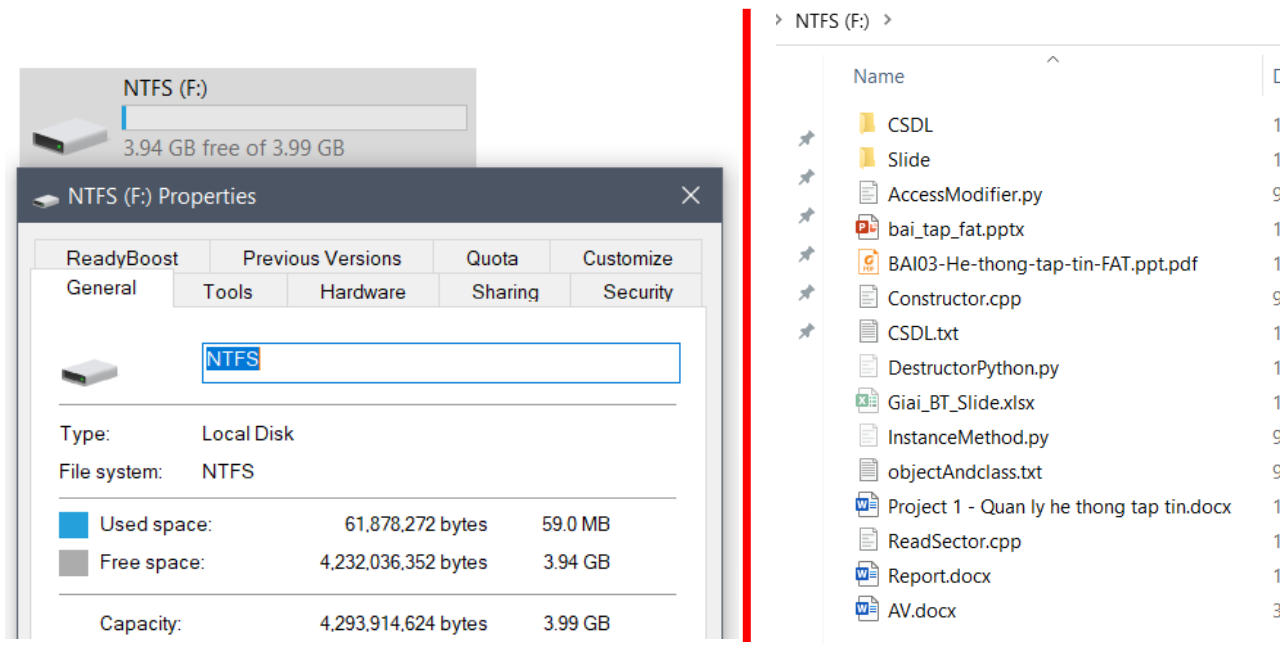
```
QuanLyDeTai
Attribute: Folder
First Cluster: 66380

Name: 19120628_QLDT.sql
Attribute: Flie
Create: -Date:2021/10/30 -Time:19:45:34
Access: -Date: 2021/10/30
Modified: -Date: 2013/10/6 -Time:1:26:52
FileSize: 1191 Byte
First Cluster: 66381
=====
Name: Quan Ly DE TAI - Bai tap.pdf
Attribute: Flie
Create: -Date:2021/10/30 -Time:19:45:34
Access: -Date: 2021/10/30
Modified: -Date: 2015/4/20 -Time:2:19:2
FileSize: 183917 Byte
First Cluster: 66382
=====
Name: Quan ly DE TAI - Mo ta du lieu.pdf
```

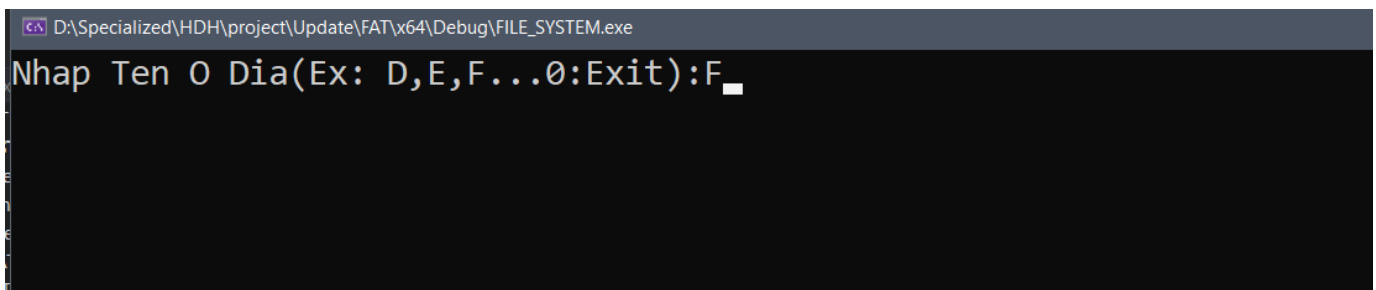
3

2. NTFS

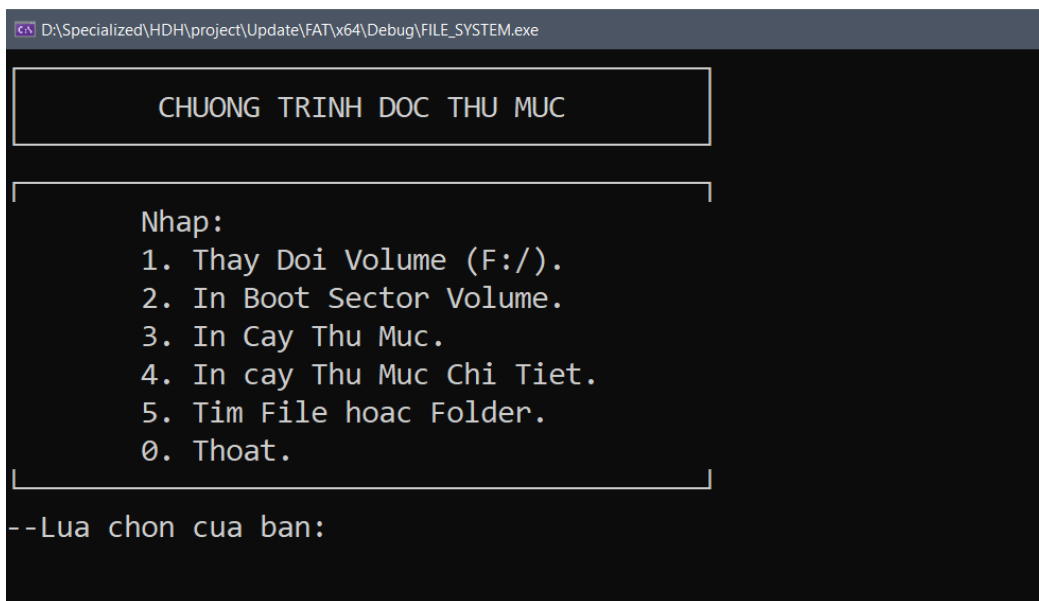
Chọn ổ đĩa F:/ có File System là NTFS để kiểm thử.



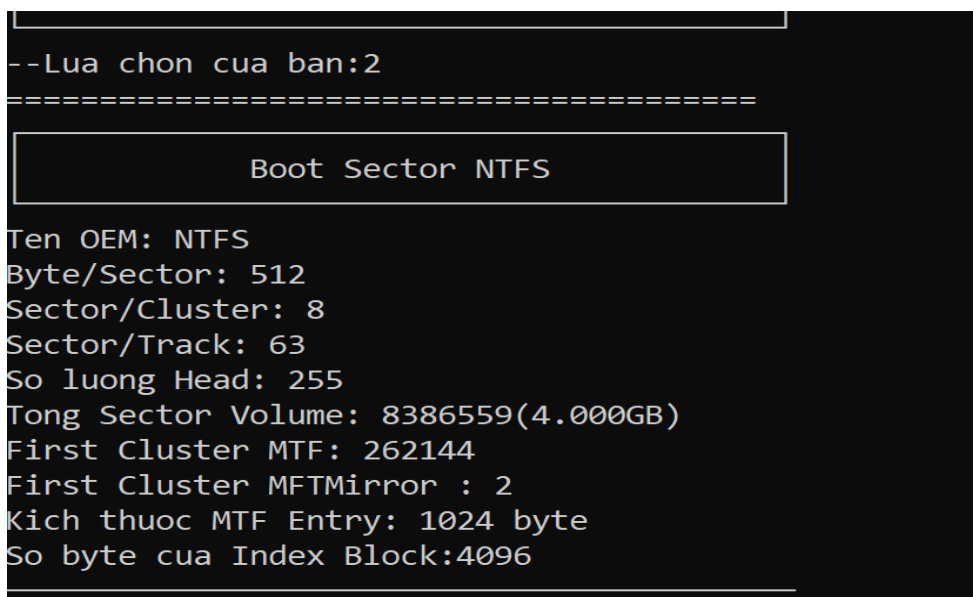
B1: Sau khi mở chương trình lên giao diện sẽ hiển thị như hình dưới. Nhập tên ổ đĩa cần xem (nhập F để demo)



Giao diện bảng chọn được hiển thị sau khi chọn ổ đĩa:



B2: Nhập 2 để xem thông tin Partition Boot sector, sau đó nhấn phím bất kỳ để trở lại menu



B3: Nhập 3 để xem cây thư mục.

```
--Lua chon cua ban:3
=====
Cay Thu Muc
L:
| bai_tap_fat.pptx(492.8KB)
| AccessModifier.py(516B)
| AV.docx(18.19KB)
| BAI03-He-thong-tap-tin-FAT.ppt.pdf(1.100MB)
| Constructor.cpp(461B)
| CSDL:
| | demo_tvlong.sql(1.130KB)
| | Mo ta CSDL:
| | | QuanLyChuyenBay:
| | | | Quan Ly CHUYEN BAY.pdf(383.6KB)
| | | | QuanLyDeTai:
| | | | | QLDT.sql(1.160KB)
| | | | | Quan Ly DE TAI - Bai tap.pdf(179.6KB)
| | | | | Quan ly DE TAI - Mo ta du lieu.pdf(351.7KB)
| | | Quanlychuyenbay.txt(27.63KB)
| | | quanlygiaovien.sql(47.84KB)
| | Slide:
| | | Quanlychuyenbay.txt(27.63KB)
| | | quanlygiaovien.sql(47.84KB)
| | Slide:
| | | CTT102-Chuong 8-PTH&DC.pdf(503.4KB)
| | | CTT102-Chuong 9- Cac van de lien quan.pdf(1.220MB)
| | ThucHanh:
| | | Bai huong dan va bai tap:
| | | | CTT102-Topic01- Huong dan cai dat va su dung SQL Se
| | | | CTT102-Topic02-Dinh nghia co so du lieu.pdf(1.110MB)
| | | | CTT102-Topic03-Truy van don gian.pdf(673.4KB)
| | | | CTT102-Topic04-Ham ket hop, gom nhom .pdf(300.5KB)
| | | | password is.fit.hcmus.vn.txt(15B)
| | | Tuan02:
| | | | BTVN2.rar(4.780KB)
| | | | _QCB.sql(6.470KB)
| | | | _QLCB2.sql(8.900KB)
| | | Tuan03:
| | | | BTTL3.sql(5.230KB)
| | | | BTN3.sql(2.780KB)
| | | Tuan04:
| | | | 19120628_HoangAnhQuan_BTVM4.rar(1.720KB)
| | | | GK_BTVM4.sql(2.260KB)
| | | | QLCB_BTVM4.sql(2.540KB)
| | CSDL.txt(1.360KB)
```

1

```

| Quanlychuyenbay.txt(27.63KB)
| quanlygiaovien.sql(47.84KB)
| Slide:
| | CTT102-Chuong 8-PTH&DC.pdf(503.4KB)
| | CTT102-Chuong 9- Cac van de lien quan.pdf(1.220MB)
| ThucHanh:
| | Bai huong dan va bai tap:
| | | CTT102-Topic01- Huong dan cai dat va su dung SQL Se
| | | CTT102-Topic02-Dinh nghia co so du lieu.pdf(1.110MB)
| | | CTT102-Topic03-Truy van don gian.pdf(673.4KB)
| | | CTT102-Topic04-Ham ket hop, gom nhom .pdf(300.5KB)
| | | password is.fit.hcmus.vn.txt(15B)
| | Tuan02:
| | | BTVN2.rar(4.780KB)
| | | _QCB.sql(6.470KB)
| | | _QLCB2.sql(8.900KB)
| | Tuan03:
| | | BTTL3.sql(5.230KB)
| | | BTN3.sql(2.780KB)
| | Tuan04:
| | | 19120628_HoangAnhQuan_BTVM4.rar(1.720KB)
| | | GK_BTVM4.sql(2.260KB)
| | | QLCB_BTVM4.sql(2.540KB)
| CSDL.txt(1.360KB)
```

2

```

| BTN3.sql(2.780KB)
| Tuan04:
| | 19120628_HoangAnhQuan_BTVM4.rar(1.720KB)
| | GK_BTVM4.sql(2.260KB)
| | QLCB_BTVM4.sql(2.540KB)
| CSDL.txt(1.360KB)
| DestructorPython.py(427B)
| Giai_BT_Slide.xlsx(34.43KB)
| InstanceMethod.py(293B)
| objectAndclass.txt(522B)
| Project 1 - Quan ly he thong tap tin.docx(21.07KB)
| ReadSector.cpp(0B)
| Report.docx(841.5KB)
| Slide:
| | CTT102-Chuong 1-TongQuan.pdf(477.9KB)
| | CTT102-Chuong 2-Mo hinh Thuc The Ket Hop.pdf(815.6KB)
Press any key to continue . . .
```

3

B4: Nhập 4 để xem cây thư mục chi tiết

Một phần của cây thư mục chi tiết

```
--Lua chon cua ban:4
=====
Cay Thu Muc Chi Tiet
-----
|
Attribute:
First Index: 5
Sector MFT entry: 2097162
-----
Name: bai_tap_fat.pptx
Attribute: Flie
Create: -Date: 2021/10/29 Time: 20:45:10
Access: -Date: 2021/11/16 Time: 11:38:47
Modified: -Date: 2021/10/10 Time: 17:37:35
FileSize: 504705 Byte
IndexEntry: 73
Sector: 2097298
-----
Name: AccessModifier.py
Attribute: Flie
Create: -Date: 2021/10/29 Time: 20:45:10
Access: -Date: 2021/11/9 Time: 19:5:38
Modified: -Date: 2021/9/26 Time: 12:19:25
FileSize: 516 Byte
```

```
Name: AccessModifier.py
Attribute: Flie
Create: -Date: 2021/10/29 Time: 20:45:10
Access: -Date: 2021/11/9 Time: 19:5:38
Modified: -Date: 2021/9/26 Time: 12:19:25
FileSize: 516 Byte
IndexEntry: 72
Sector: 2097296
-----
Name: AV.docx
Attribute: Flie
Create: -Date: 2021/11/16 Time: 16:20:13
Access: -Date: 2021/11/16 Time: 16:20:13
Modified: -Date: 2021/3/8 Time: 12:18:34
FileSize: 18635 Byte
IndexEntry: 42
Sector: 2097236
-----
Name: BAI03-He-thong-tap-tin-FAT.ppt.pdf
Attribute: Flie
Create: -Date: 2021/10/29 Time: 20:45:10
Access: -Date: 2021/10/30 Time: 20:46:37
Modified: -Date: 2021/10/16 Time: 24:21:17
FileSize: 1149373 Byte
IndexEntry: 74
Sector: 2097300
```

```
Mo ta CSDL
Attribute: Folder
First Index: 109
Sector MFT entry: 2097370
-----
|
QuanLyChuyenBay
Attribute: Folder
First Index: 110
Sector MFT entry: 2097372
-----
Name: Quan Ly CHUYEN BAY.pdf
Attribute: Flie
Create: -Date: 2021/10/30 Time: 20:44:47
Access: -Date: 2021/10/30 Time: 20:44:47
Modified: -Date: 2013/3/4 Time: 16:9:23
FileSize: 392861 Byte
IndexEntry: 111
Sector: 2097374
-----
|
QuanLyDeTai
Attribute: Folder
First Index: 112
Sector MFT entry: 2097376
```

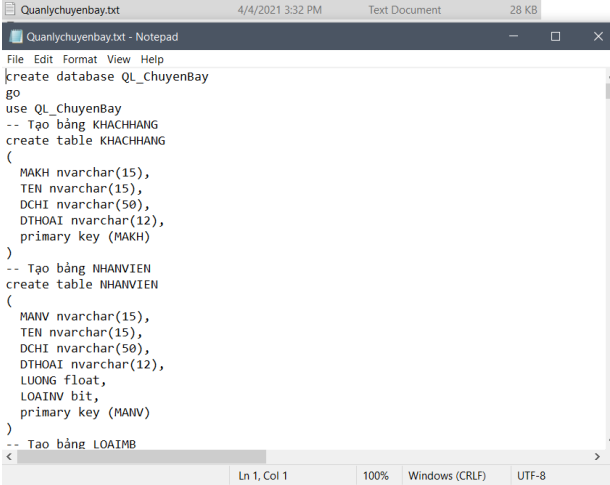
B5 :Nhập 5 để tìm file hoặc Folder

+ Tìm file: Quanlychuyenbay.txt

Một phần của nội dung do file có kích thước lớn(28302byte)

```
--Lua chon cua ban:5
=====
Nhap Ten Entry:Quanlychuyenbay.txt
---Thong Tin File---
Name: Quanlychuyenbay.txt
Attribute: Flie
Create:      -Date: 2021/10/30      Time: 20:44:47
Access:      -Date: 2021/11/16     Time: 16:18:24
Modified:     -Date: 2021/4/4      Time: 16:33:6
FileSize:     28302 Byte
IndexEntry: 107
Sector: 2097366
-----
-----DATA-----
create database QL_ChuyenBay
go
use QL_ChuyenBay
-- Tạo bảng KHACHHANG
create table KHACHHANG
(
    MAKH nvarchar(15),
    TEN nvarchar(15),
    DCHI nvarchar(50),
    DTHOAI nvarchar(12),
    primary key (MAKH)
)
```

```
insert into KHACHHANG values ('0091', 'Hai', '345 Hung Vuong', '8893223')
insert into KHACHHANG values ('0314', 'Phuong', '385 Vo Van Tuan', '8232')
insert into KHACHHANG values ('0613', 'Vu', '348 CMT8', '8343232')
insert into KHACHHANG values ('0586', 'Son', '123 Bach Dang', '8556223')
insert into KHACHHANG values ('0422', 'Tien', '75 Nguyen Thong', '833222')
--Nhập dữ liệu cho table CHUYENBAY
insert into CHUYENBAY values ('100', 'SLC', 'BOS', '08:00', '17:59')
insert into CHUYENBAY values ('112', 'DCA', 'DEN', '14:00', '18:07')
insert into CHUYENBAY values ('121', 'STL', 'SLC', '07:00', '9:13')
insert into CHUYENBAY values ('122', 'STL', 'YV', '08:30', '10:19')
insert into CHUYENBAY values ('206', 'DFW', 'STL', '09:00', '11:40')
insert into CHUYENBAY values ('330', 'JFK', 'YV', '16:00', '18:53')
insert into CHUYENBAY values ('334', 'ORD', 'MIA', '12:00', '14:14')
insert into CHUYENBAY values ('335', 'MIA', 'ORD', '15:00', '17:14')
insert into CHUYENBAY values ('336', 'ORD', 'MIA', '18:00', '20:14')
insert into CHUYENBAY values ('337', 'MIA', 'ORD', '20:30', '23:53')
insert into CHUYENBAY values ('394', 'DFW', 'MIA', '19:00', '21:30')
insert into CHUYENBAY values ('395', 'MIA', 'DFW', '21:00', '23:43')
insert into CHUYENBAY values ('449', 'CDG', 'DEN', '10:00', '19:29')
insert into CHUYENBAY values ('930', 'YV', 'DCA', '13:00', '16:10')
insert into CHUYENBAY values ('931', 'DCA', 'YV', '17:00', '18:10')
insert into CHUYENBAY values ('932', 'DCA', 'YV', '18:00', '19:10')
insert into CHUYENBAY values ('991', 'BOS', 'ORD', '17:00', '18:22')
-- Nhập dữ liệu cho table LOAIMB
insert into LOAIMB values ('A310', 'Airbus')
insert into LOAIMB values ('A320', 'Airbus')
insert into LOAIMB values ('A330', 'Airbus')
```



Quanlychuyenbay.txt 4/4/2021 3:32 PM Text Document 28 KB

Quanlychuyenbay.txt - Notepad

```
File Edit Format View Help
create database QL_ChuyenBay
go
use QL_ChuyenBay
-- Tạo bảng KHACHHANG
create table KHACHHANG
(
    MAKH nvarchar(15),
    TEN nvarchar(15),
    DCHI nvarchar(50),
    DTHOAI nvarchar(12),
    primary key (MAKH)
)
-- Tạo bảng NHANVIEN
create table NHANVIEN
(
    MANV nvarchar(15),
    TEN nvarchar(15),
    DCHI nvarchar(50),
    DTHOAI nvarchar(12),
    LUONG float,
    LOAINV bit,
    primary key (MANV)
)
-- Tạo bảng LOAIMB
```

Ln 1, Col 1 100% Windows (CRLF) UTF-8

+ Tìm Folder: CSDL (E:/CSDL)

Nhap Ten Entry:CSDL

Cay Thu Muc Con

```

CSDL:
├─ demo_tvlong.sql(1.130KB)
├─ Mo ta CSDL:
│   └─ QuanLyChuyenBay:
│       └─ Quan Ly CHUYEN BAY.pdf(383.6KB)
│       └─ QuanLyDeTai:
│           ├── QLDI.sql(1.160KB)
│           ├── Quan Ly DE TAI - Bai tap.pdf(179.6KB)
│           └─ Quan ly DE TAI - Mo ta du lieu.pdf(351.7KB)
├─ Quanlychuyenbay.txt(27.63KB)
├─ quanlygiaovien.sql(47.84KB)
├─ Slide:
│   ├── CTT102-Chuong 8-PTH&DC.pdf(503.4KB)
│   └─ CTT102-Chuong 9- Cac van de lien quan.pdf(1.220MB)
├─ ThucHanh:
│   └─ Bai huong dan va bai tap:
│       ├── CTT102-Topic01- Huong dan cai dat va su dung SQL Server .pdf(3.230MB)
│       ├── CTT102-Topic02-Dinh nghia co so du lieu.pdf(1.110MB)
│       ├── CTT102-Topic03-Truy van don gian.pdf(673.4KB)
│       ├── CTT102-Topic04-Ham ket hop, gom nhom .pdf(300.5KB)
│       └─ password is.fit.hcmus.vn.txt(15B)
└─ Tuan02:
    ├── BTVN2.rar(4.780KB)
    ├── QCB.sql(6.470KB)
    └─ QLCB2.sql(8.900KB)
└─ Tuan03:
    ├── BTTL3.sql(5.230KB)
    └─ BTN3.sql(2.780KB)
└─ Tuan04:
    ├── 19120628_HoangAnhQuan_BTVM4.rar(1.720KB)
    ├── GK_BTVM4.sql(2.260KB)
    └─ QLCB_BTVM4.sql(2.540KB)

```

Cay Thu Muc Chi Tiet

```

├─ CSDL
Attribute: Folder
First Index: 103
Sector MFT entry: 2097358

Name: demo_tvlong.sql
Attribute: File
Create: -Date: 2021/10/30 Time: 20:44:47
Access: -Date: 2021/11/13 Time: 21:4:34
Modified: -Date: 2021/4/3 Time: 14:2:5
FileSize: 1162 Byte

```

Cay Thu Muc Chi Tiet

```

├─ CSDL
Attribute: Folder
First Index: 103
Sector MFT entry: 2097358

Name: demo_tvlong.sql
Attribute: File
Create: -Date: 2021/10/30 Time: 20:44:47
Access: -Date: 2021/11/13 Time: 21:4:34
Modified: -Date: 2021/4/3 Time: 14:2:5
FileSize: 1162 Byte
IndexEntry: 106
Sector: 2097364
-----
├─ Mo ta CSDL
Attribute: Folder
First Index: 109
Sector MFT entry: 2097370

├─ QuanLyChuyenBay
Attribute: Folder

```

VI. NGUỒN KHAM KHẢO

Khảo sát hệ thống tập tin NTFS (Lê Gia Công)

Brian Carrie, *File System Forensic Analysis*, Addison Wesley Professional, 2005s

<http://legiacong.blogspot.com/2014/04/he-thong-quan-ly-tap-tin-ntfs-6-mft-mft.html>

https://www.ntfs.com/ntfs_basics.htm

http://inform.pucp.edu.pe/~inf232/Ntfs/ntfs_doc_v0.5/concepts/index_record.html

<https://homepage.cs.uri.edu/~thenry/csc487/video/>

https://flylib.com/books/en/2.48.1/ntfs_data_structures.html