

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



MÔN: AN TOÀN VÀ PHỤC HỒI DỮ LIỆU – CSC15104
ĐỒ ÁN GIỮA KỲ

Giảng viên lý thuyết: Thái Hùng Văn

Giảng viên thực hành: Đặng Trần Minh Hậu

Thành viên nhóm: Trần Thanh Hoà - 20127163
Nguyễn Thanh Phong - 20127274
Lê Nguyễn Minh Quang - 20127295

Thành phố Hồ Chí Minh – 11/2023

Mục lục

I.	Thành viên nhóm.....	3
II.	Phân công công việc.....	3
1.	Phần báo cáo	3
2.	Phần code	4
III.	Giải pháp mô hình.....	5
1.	Lý thuyết.....	5
2.	Thực hành	5
IV.	Cấu trúc của Volume .HPQ	6
1.	Vùng Header	6
2.	Vùng Entry table	6
3.	Vùng data	7
V.	Trình bày thuật toán.....	8
1.	Tạo mới Volume.....	8
2.	Chép file vào Volume.....	9
3.	Xuất file khỏi Volume.....	10
4.	Tạo folder	11
5.	Chép file từ ngoài vào folder	12
6.	Xóa file (2 option: xóa phục hồi được và không phục hồi được).....	13
7.	Vùng Backup.....	14
8.	Xuất danh sách file	15
VI.	Hình ảnh chạy thử phần mềm	17
VII.	Tài liệu tham khảo.....	20

I. Thành viên nhóm

STT	Họ và tên	MSSV
1	Trần Thanh Hoà	20127163
2	Nguyễn Thanh Phong	20127274
3	Lê Nguyễn Minh Quang	20127295

II. Phân công công việc

1. Phân báo cáo

Công việc	Người thực hiện	Mức độ hoàn thành
Trình bày mô hình Volume .HPQ	Nguyễn Thanh Phong	100%
Trình bày thuật toán chép File từ ngoài vào Volume	Nguyễn Thanh Phong	100%
Trình bày thuật toán hệ thống thư mục	Nguyễn Thanh Phong	100%
Trình bày thuật toán mã hóa Volume và File	Nguyễn Thanh Phong Trần Thanh Hòa	100%
Trình bày quá trình Backup	Trần Thanh Hòa	100%
Trình bày quá trình đặt mật khẩu cho Volume và File	Trần Thanh Hòa	100%
Trình bày quá trình xuất danh sách File	Lê Nguyễn Minh Quang	100%
Trình bày quá trình xóa File	Lê Nguyễn Minh Quang	100%

2. Phần code

Công việc	Người thực hiện	Mức độ hoàn thành
Thiết kế tổng quan mô hình Volume	Nguyễn Thanh Phong	100%
Tạo mới 1 Volume	Nguyễn Thanh Phong	100%
Đọc Volume	Nguyễn Thanh Phong	100%
Mã hóa Volume	Nguyễn Thanh Phong Trần Thanh Hòa	100%
Giải mã Volume	Nguyễn Thanh Phong Trần Thanh Hòa	100%
Mã hóa File trong Volume	Nguyễn Thanh Phong Trần Thanh Hòa	100%
Giải mã File trong Volume	Nguyễn Thanh Phong Trần Thanh Hòa	100%
Viết File từ ngoài vào Volume (Viết Entry và data)	Nguyễn Thanh Phong	100%
Đọc 1 Entry trên Sector	Nguyễn Thanh Phong	100%
Xuất file từ Volume ra ngoài (Có kiểm tra mật khẩu khi xuất)	Trần Thanh Hòa	100%
Vùng backup dữ liệu quan trọng	Trần Thanh Hòa	100%
Đặt, đổi, xoá mật khẩu file	Trần Thanh Hòa	100%
Đặt, đổi, xoá mật khẩu Volume	Trần Thanh Hòa	100%
Đọc tất cả Entry trên Sector	Lê Nguyễn Minh Quang	100%
Xóa file và dữ liệu của file theo tên	Lê Nguyễn Minh Quang	100%
Xuất danh sách file	Lê Nguyễn Minh Quang	100%
Xóa file không thể phục hồi hoặc có thể	Lê Nguyễn Minh Quang	100%

III. Giải pháp mô hình

1. Lý thuyết

Yêu cầu	Giải pháp	Người thực hiện	Tiến độ
Việc bảo mật thông tin	Những dữ liệu bình thường sẽ được chuyển thành chuỗi byte để lưu trong Volume Ngoài ra còn có bảo mật nhiều lớp(mật khẩu ở Volume và ở từng Entry)	Nguyễn Thanh Phong Trần Thanh Hòa	100%
Việc bảo vệ dữ liệu	Sẽ có phần backup những dữ liệu quan trọng	Trần Thanh Hòa	100%
Việc phục hồi dữ liệu	Mỗi Entry sẽ có offset lưu trạng thái đã xóa chưa và có thể dựa vào nó để phục hồi lại	Trần Thanh Hòa	100%
Việc mã hóa dữ liệu	Dùng password để làm passkey kết hợp với AES để mã hóa tệp tin	Nguyễn Thanh Phong	100%
Tổ chức hệ thống thư mục	Có hàm tạo folder và các hàm thao tác liên quan	Nguyễn Thanh Phong	100%
Truy xuất theo từng cụm 512 byte	Có hàm đọc ghi file theo cụm 512	Nguyễn Thanh Phong	100%
Vùng dành riêng để backup	Vùng backup sẽ có offset đánh dấu bắt đầu và với mỗi kích thước Volume khác nhau thì kích thước của vùng back up cũng khác nhau (5% kích thước Volume)	Trần Thanh Hòa	100%

2. Thực hành

Yêu cầu	Người thực hiện	Tiến độ
Tạo / định dạng Volume MyFS.DRS	Nguyễn Thanh Phong	100%
Thiết lập /Đổi /Kiểm tra password truy xuất MyFS	Nguyễn Thanh Phong Trần Thanh Hòa	100%
Liệt kê danh sách các file trong MyFS	Lê Nguyễn Minh Quang	100%
Đặt /đổi password truy xuất cho 1 file trong MyFS	Nguyễn Thanh Phong Trần Thanh Hòa	100%
Chép (Import) 1 file từ bên ngoài vào trong MyFS	Nguyễn Thanh Phong	100%

Chép (Outport) 1 file trong MyFS ra ngoài	Trần Thanh Hòa	100%
Xóa 1 file trong MyFS (cho phép người dùng chọn dạng không thể phục hồi hoặc có thể)	Lê Nguyễn Minh Quang	100%

IV. Cấu trúc của Volume .HPQ

1. Vùng Header

Offset	Độ lớn (byte)	Nội dung
0	4	Tên Volume (Mặc định là .HPQ)
4	4	Kích thước Volume
10	32	Password đã được hash (Nếu có)

- Ghi chú:
 - Entry table đầu tiên sẽ là Sector thứ 2

2. Vùng Entry table

- Vùng Entry table sẽ lưu 6 Entry tương ứng với 6 file hoặc folder
- Offset 1FC độ lớn 4 bytes sẽ lưu vị trí Entry tiếp theo nếu thiếu chỗ (nếu chưa có Entry tiếp theo thì sẽ mang giá trị 00 00 00 00h)
- Mỗi Entry sẽ có cấu trúc như sau:

Offset	Độ lớn (byte)	Nội dung
0	32	Tên file
20	5	Định dạng
25	2	Ngày tạo
27	4	Vị trí bắt đầu dữ liệu
2B	4	Kích thước tệp tin
2F	1	Trạng thái của file (01: Folder, 02: File, 03: Backup File, 04: File đã xóa)
30	32	Password để đọc file đó (Nếu có)

- Ghi chú:
 - Nếu không có password thì 32 bytes sẽ bỏ trống (đều là 00h)

- Do dữ liệu sẽ được lưu theo cấu trúc link list nên không cần bản FAT
- Ngày vẫn giữ kiểu đọc 5 bits cho ngày 4 bits cho tháng và 7 bits cho năm + 1980
- Lưu dữ liệu theo định dạng “big endian” nên dữ liệu sẽ đọc từ trái sang phải
- Vậy ví dụ để tìm ngày tháng tạo, giá trị offset 23 là 57h và offset 24 là 59h thì sẽ được tính như sau:
 - $5759h = 0101011.1010.11001b \rightarrow$ ngày 25 tháng 10 năm 2023 (43 + 1980)
- Nếu Sector đầu tiên chứa quá 6 Entry thì hệ thống sẽ tìm Sector trống tiếp theo để lưu bản Entry theo kiểu link list. Vị trí Sector chứa bản Entry tiếp theo sẽ lưu trong bản Entry Table trước đó.
- 1 Folder sẽ vẫn có Entry như các tệp tin khác nhưng phần tên mở rộng sẽ được bỏ trống. Và Sector đầu tiên của Folder đó sẽ là bảng Entry để lưu Entry của các file được tạo trong folder đó (cấu trúc cây).

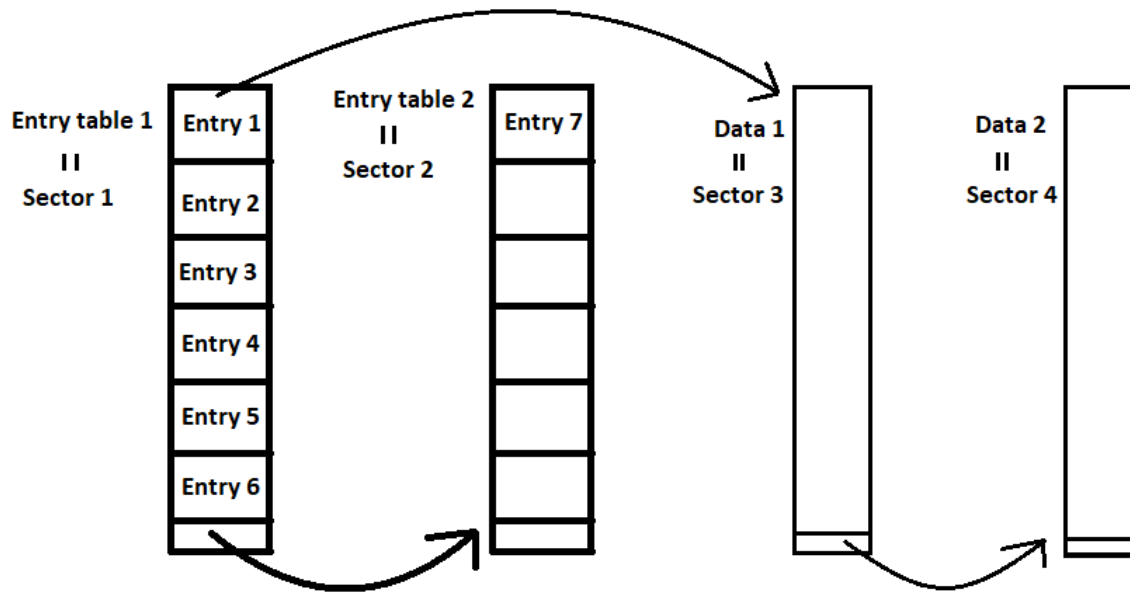
3. Vùng data

Offset	Độ lớn (byte)	Nội dung
0	508	Vùng chứa dữ liệu
1FC	4	Vị trí Sector tiếp theo của dữ liệu

- Dữ liệu sẽ được lưu theo kiểu link list nên có thể bỏ được bản FAT, nhưng điều này sẽ làm cho dữ liệu dễ bị mất nếu có hư hỏng

Tổng quan:

- Sẽ không có phân vùng cụ thể cho từng vùng mà Entry table đã đủ 6 Entry thì sẽ tìm Sector trống và lưu Entry table tiếp theo (Ví dụ như Sector 1 đã lưu 6 Entry thì sẽ tìm 1 Sector trống để lưu Entry thứ 7)
- Sau khi lưu thành công Entry lên Entry table, sẽ tìm Sector trống để lưu phần data (Entry 1 sẽ có Data gồm 2 Sector là 3 và 4)



(Tổng quan về phần Entry và Data)

V. Trình bày thuật toán

1. Tạo mới Volume

- B1: Đầu tiên tạo header rỗng
- B2: Yêu cầu người dùng chọn độ lớn của Volume
- B3: Hỏi về mật khẩu
- B4: Chuyển những thông tin của Volume thành 1 block 512 bytes rồi lưu vào Sector tương ứng

```
def createVolume(path):
    header = Header()
    header.SizeOfVolume = chooseSizeOfVolume().to_bytes(4, byteorder='big')
    option = input("1: Đặt mật khẩu\n2: bỏ qua\n")
    if option == '1':
        temp, header.Password = createNewPass()
        print("Tạo mật khẩu thành công")
    else:
        print("Không có mật khẩu")
    dataBlock = padding(padding(header.Signature + header.SizeOfVolume, 16) + header.Password, 512)

    if os.path.exists(path + "/.HPQ"):
        print("Đã tồn tại. Vui lòng chọn đường dẫn khác.")
    else:
        with open(path + "/.HPQ", "wb") as file:
            file.seek(int.from_bytes(header.SizeOfVolume, byteorder='big') - 1)
            file.write(b'\0')
            file.close()
    write512(path, dataBlock, 0)
    return header
```


2. Mã hóa, giải mã Volume

Sau khi chạy đã thao tác mọi thứ trên Volume, bước cuối cùng trước khi thoát chương trình là sẽ mã hóa toàn bộ Volume nếu Volume đó có password. Và trước khi đọc được nội dung Volume phải giải mã toàn bộ Volume

```
def encVolume(path, header):
    for i in range(1, int.from_bytes(header.SizeOfVolume, byteorder='big') // 512):
        data = read512(path, i)
        write512(path, AES.dec(padding(header.PassWord, 32), data), i)

def decVolume(path, header):
    for i in range(1, int.from_bytes(header.SizeOfVolume, byteorder='big') // 512):
        data = read512(path, i)
        write512(path, AES.enc(padding(header.PassWord, 32), data), i)
```

*Lưu ý: Do đây là đọc và ghi từng cụm 512 byte nên sẽ tốn thời gian để mã hóa và giải mã

3. Chép file vào Volume

- B1: Đọc thư mục muốn chép vào Volume và lưu các thông tin vào Entry trống
- B2: Viết Entry đó vào Entry table đầu tiên (vì theo như mô hình, ngoài việc lưu trên file thì Entry table đầu tiên sẽ là gốc, nếu nó đầy thì sẽ tìm Entry table tiếp theo đến khi nào tìm được chỗ trống)
- B3: Tìm Sector trống để lưu data

```
def copyToVolume(path, header):
    while True:
        os.system('cls')
        print("1: Lưu trong thư mục")
        print("2: Không lưu trong thư mục")
        print("3: Tạo thư mục mới")
        print("0: Thoát chương trình")
        choice = input("Nhập lựa chọn của bạn: ")
        if choice == "0":
            break
        file_path = input("Nhập đường dẫn chứa file (ví dụ E:\\report.pdf): ")
        entry = Entry()
        if os.path.exists(file_path): # lưu 1 entry
            entry.Name = os.path.splitext(os.path.basename(file_path))[0].encode('utf-8')
            entry.Extended = os.path.splitext(file_path)[1].encode('utf-8')
            creation_time = os.path.getctime(file_path)
            created_datetime = datetime.fromtimestamp(creation_time)
            created_date = created_datetime.strftime('%d-%m-%Y')
            entry.DateCreate = convert_date_to_byte(created_date)
            entry.Size = os.path.getsize(file_path).to_bytes(4, byteorder='big')
            entry.State = b'\x02'
```

```

if choice == "1":
    input_name = input("Nhập tên thư mục để lưu: ")
    pos,sector = findPosByName(path, input_name)# tìm entry theo tên
    if sector == 0:
        print("Folder không tồn tại")
    else:
        folder_entry = readOneEntry(path,sector,pos)# nếu tìm thấy entry thì đọc entry
        if is_all_zeros(folder_entry.LocationOfData):# kiểm tra xem entry đó có data chưa (data của folder là bảng entry)
            new_sector = findEmptySector(path)# chưa thì tìm sector trống để lưu
            folder_entry.LocationOfData = new_sector.to_bytes(4, byteorder='big')# lưu địa chỉ sector trống đó vào 4byte quy định
            with open(path + "/.HPQ", "rb+") as file:
                file.seek(512 * sector + pos + 39)
                file.write(folder_entry.LocationOfData)

            data_pos = writeEntry(entry,path,new_sector)# viết entry của file cần lưu vào trả ra vị trí sector trống để lưu data
            with open(file_path, "rb+") as file:# viết data vào
                data = file.read()
            if data_pos != 0:
                writeData(data,path,data_pos)
                writeBackupData(entry, data, path, header)
            secure(path, (entry.Name).decode('utf-8'))
        else:
            pos = writeEntry(entry,path,int.from_bytes(folder_entry.LocationOfData, byteorder='big'))# entry gốc
            with open(file_path, "rb+") as file:
                data = file.read()
            if pos != 0:
                writeData(data,path,pos,entry.Password)
                writeBackupData(entry, data, path, header)

elif choice == "2":
    pos = writeEntry(entry,path,1)# entry gốc
    with open(file_path, "rb+") as file:
        data = file.read()
    if pos != 0:
        writeData(data,path,pos)
        writeBackupData(entry, data, path, header)
    secure(path, (entry.Name).decode('utf-8'))
elif choice == "3":
    input_name = input("Nhập tên thư mục: ")
    pos,sector = findPosByName(path, input_name)# tìm entry theo tên
    if sector == 0:
        folder_entry = Entry()
        folder_entry.Name = input_name.encode('utf-8')
        writeEntry(folder_entry,path,1)
        secure(path, (folder_entry.Name).decode('utf-8'))
    else:
        print("Đã tồn tại")

```

4. Xuất file khỏi Volume

- B1: Nhập tên và tìm Entry chứa File trên Volume
- B2: Nhập và xác nhận mật khẩu của File để giải mã và xuất File
- B3: Đọc thông tin File và tiến hành mã hoá lại File (bảo mật)
- B4: Xuất File ra thư mục mà người dùng muốn

```

def exportFromVolume(path):
    export_path = input("Nhập đường dẫn xuất (ví dụ: /path/to/export): ")

    while True:
        os.system('cls')
        input_name = input("Nhập tên thư mục/file cần xuất (0 để thoát): ")
        if input_name == "0":
            print("Thoát chương trình.")
            break

        pos, sector = findPosByName(path, input_name) # Find the entry by name
        if sector == 0:
            print("Thư mục/file không tồn tại.")
        else:
            entry = readOneEntry(path, sector, pos) # Read the entry

    if entry.State == b'\x01': # If it's a folder
        folder_entries = readEntriesInFolder(path, int.from_bytes(entry.LocationOfData, byteorder='big'))
        print(f"Exporting folder: {entry.Name.decode()} to {export_path}")
        folder_path = os.path.join(export_path, entry.Name.decode())
        os.makedirs(folder_path, exist_ok=True)
        for folder_entry in folder_entries:
            if folder_entry.State == b'\x02': # If it's a file
                password = input(f"Enter password for {folder_entry.Name.decode() + folder_entry.Extended.decode()}: ")
                if checkFilePassword(path, folder_entry, password):
                    encrypted_data = readData(path, int.from_bytes(folder_entry.LocationOfData, byteorder='big'))
                    decrypted_data = decData(path, int.from_bytes(folder_entry.LocationOfData, byteorder='big'), password)
                    file_name = folder_entry.Name.decode() + folder_entry.Extended.decode()
                    file_path = os.path.join(folder_path, file_name)
                    with open(file_path, "wb") as file:
                        file.write(decrypted_data)
                    print(f"Exported file: {file_path}")
                else:
                    print("Incorrect password. File cannot be exported.")
            elif entry.State == b'\x02': # If it's a file
                password = input(f"Enter password for {entry.Name.decode() + entry.Extended.decode()}: ")
                if checkFilePassword(path, entry, password):
                    decData(path, int.from_bytes(entry.LocationOfData, byteorder='big'), password)
                    data = readData(path, int.from_bytes(entry.LocationOfData, byteorder='big'))
                    encData(path, int.from_bytes(entry.LocationOfData, byteorder='big'), password)
                    file_name = entry.Name.decode() + entry.Extended.decode()
                    file_path = os.path.join(export_path, file_name)
                    with open(file_path, "wb") as file:
                        file.write(data)
                    print(f"Exported file: {file_path}")
                else:
                    print("Incorrect password. File cannot be exported.")
            else:
                print("Unknown entry type.")

```

5. Tạo folder

Folder cũng là file nhưng sẽ không có phần mở rộng

- B1: Cho người dùng nhập tên folder

- B2: Tạo 1 Entry với tên folder đó
- B3: Viết Entry đó vào Entry table đầu tiên

6. Chép file từ ngoài vào folder

- B1: Đọc thư mục muốn chép vào Volume và lưu các thông tin vào Entry trống
- B2: Yêu cầu người dùng nhập tên folder
- B3: Tìm vị trí folder trong Volume dựa theo tên nhập vào trả ra vị trí của Entry Folder đó và Sector chứa nó
- B4: Dựa theo vị trí Entry và Sector tìm được vị trí phần data folder (phần data của folder sẽ lưu Entry của file) nếu vị trí đó rỗng thì tìm Sector trống để lưu
- B5: Viết Entry của thư mục muốn chép vào Entry table của folder đó
- B6: Tìm Sector trống để lưu dữ liệu của thư mục muốn chép vào

```
def copyToVolume(path, header):
    while True:
        os.system('cls')
        print("1: Lưu trong thư mục")
        print("2: Không lưu trong thư mục")
        print("3: Tạo thư mục mới")
        print("0: Thoát chương trình")
        choice = input("Nhập lựa chọn của bạn: ")
        if choice == "0":
            break
        file_path = input("Nhập đường dẫn chứa file(ví dụ E:\\report.pdf): ")
        entry = Entry()
        if os.path.exists(file_path):#lưu 1 entry
            entry.Name = os.path.splitext(os.path.basename(file_path))[0].encode('utf-8')
            entry.Extended = os.path.splitext(file_path)[1].encode('utf-8')
            creation_time = os.path.getctime(file_path)
            created_datetime = datetime.fromtimestamp(creation_time)
            created_date = created_datetime.strftime('%d-%m-%Y')
            entry.DateCreate = convert_date_to_byte(created_date)
            entry.Size = os.path.getsize(file_path).to_bytes(4, byteorder='big')
            entry.State = b'\x02'
```

```

if choice == "1":
    input_name = input("Nhập tên thư mục để lưu: ")
    pos,sector = findPosByName(path, input_name)# tìm entry theo tên
    if sector == 0:
        print("Folder không tồn tại")
    else:
        folder_entry = readOneEntry(path,sector,pos)# nếu tìm thấy entry thì đọc entry
        if is_all_zeros(folder_entry.LocationOfData):# kiểm tra xem entry đó có data chưa (data của folder là bảng entry)
            new_sector = findEmptySector(path)# chưa thì tìm sector trống để lưu
            folder_entry.LocationOfData = new_sector.to_bytes(4, byteorder='big')# lưu địa chỉ sector trống đó vào 4byte quy định
            with open(path + ".HPQ", "rb+") as file:
                file.seek(512 * sector + pos + 39)
                file.write(folder_entry.LocationOfData)

            data_pos = writeEntry(entry,path,new_sector)# viết entry của file cần lưu vào trả ra vị trí sector trống để lưu data
            with open(file_path, "rb+") as file:# viết data vào
                data = file.read()
            if data_pos != 0:
                writeData(data,path,data_pos)
                writeBackupData(entry, data, path, header)
            secure(path, (entry.Name).decode('utf-8'))
        else:
            pos = writeEntry(entry,path,int.from_bytes(folder_entry.LocationOfData, byteorder='big'))# entry gốc
            with open(file_path, "rb+") as file:
                data = file.read()
            if pos != 0:
                writeData(data,path,pos,entry.Password)
                writeBackupData(entry, data, path, header)

elif choice == "2":
    pos = writeEntry(entry,path,1)# entry gốc
    with open(file_path, "rb+") as file:
        data = file.read()
    if pos != 0:
        writeData(data,path,pos)
        writeBackupData(entry, data, path, header)
    secure(path, (entry.Name).decode('utf-8'))
elif choice == "3":
    input_name = input("Nhập tên thư mục: ")
    pos,sector = findPosByName(path, input_name)# tìm entry theo tên
    if sector == 0:
        folder_entry = Entry()
        folder_entry.Name = input_name.encode('utf-8')
        writeEntry(folder_entry,path,1)
        secure(path, (folder_entry.Name).decode('utf-8'))
    else:
        print("Đã tồn tại")

```

7. Xóa file (2 option: xóa phục hồi được và không phục hồi được)

- Cho nhập tên file cần xóa, rồi chạy hàm findPosByName(), hàm sẽ trả về vị trí của Entry đó và Sector tương ứng. Sau khi có được vị trí của Entry sẽ tiến hành đọc tên file từ Entry.Name. Nếu nó và tên file cần xóa giống nhau thì sẽ tiến hành chạy phần xóa.
- Sẽ có 2 lựa chọn xóa phục hồi được và không phục hồi được:

- xóa phục hồi được (performSoftDeletion): nó sẽ thay đổi giá trị tại offset 2F thành \x04.
- xóa không phục hồi được (performUnrecoverableDeletion): đầu tiên sẽ lấy vị trí đã lưu data của Entry đó tại offset 27h, sau đó xóa tất cả các data của vùng Sector đó. Tiếp theo sẽ lấy vị trí và Sector của Entry được yêu cầu xóa, xóa tất cả các dữ liệu của Entry đó và thay đổi giá trị của thành \x00.

```
def deleteFileFromVolume(path):
    # Search for the file in the volume
    found = False
    while True:
        os.system('cls')
        input_name = input("Nhập tên thư mục/file cần xuất (0 để thoát): ")
        if input_name == "0":
            print("Thoát chương trình.")
            break

        pos, sector = findPosByName(path, input_name) # Find the entry by name
        if sector != 0:
            entry = readOneEntry(path, sector, pos) # Read the entry
            if entry.Name.decode('utf-8') == input_name:
                found = True
                print(f"File '{input_name}' được tìm thấy trong MyFS.")
                password = input("Nhập mật khẩu để xóa file: ")
                if checkFilePassword(path, entry, password):
                    confirm = input("Bạn có chắc chắn muốn xóa? (Y/N): ")
                    if confirm.upper() == "Y":
                        if unrecoverableDeletionOption():
                            # Perform unrecoverable deletion
                            performUnrecoverableDeletion(path, entry)
                            print(f"File '{input_name}' đã bị xóa vĩnh viễn khỏi MyFS.")
                        else:
                            # Perform soft deletion
                            performSoftDeletion(path, entry)
                            print(f"File '{input_name}' đã được đánh dấu xóa khỏi MyFS.")
                    else:
                        print("Mật khẩu không chính xác. Không thể xóa file.")
                        break
            else:
                print(f"File '{input_name}' không tồn tại trong MyFS.")
        if not found:
            print(f"File '{input_name}' không tồn tại trong MyFS.")
```

8. Vùng Backup

- Chọn phần cuối cùng của Volume làm phần lưu dữ liệu Backup với kích thước bằng 5% kích thước Volume.
- Khi chép File vào Volume, người dùng được lựa chọn lưu thêm bản sao File vào phần Backup và chỉ có thể truy xuất bằng các phần mềm đọc ổ đĩa.

```

# Backup dữ liệu
def posOfBackup(header): # trả về sector bắt đầu vùng backup
    size = int.from_bytes(header.SizeOfVolume, byteorder='big') // 512
    return int(size - (size * 5 / 100))
def writeBackupData(entry, data, path, header):
    print("Lưu file backup?")
    print("1. Lưu")
    print("Khác. Thoát")
    choice = input("Nhập lựa chọn của bạn: ")
    entry.State = b'\x03'
    if choice == "1":
        backupZone = posOfBackup(header)
        pos = writeEntry(entry, path, backupZone)
        if pos != 0:
            writeData(data, path, pos)
    else:
        return

```

9. Xuất danh sách file

- Cho chạy từ Sector đầu tiên của MyFS, Đọc dữ liệu từ Sector rồi chia 80bytes cho mỗi Entry. Bắt đầu với Entry đầu tiên nếu có dữ liệu ở Entry[pos]:
- Tạo một đối tượng Entry để trích xuất các thông tin của Entry từ block dữ liệu đọc được: Thêm Entry mới tạo vào danh sách các Entry
- Di chuyển pos sang phía trước 80 bytes để đọc Entry tiếp theo.
- Tăng giá trị Sector lên 1 để chuyển sang Sector tiếp theo của MyFS.
- Trả về danh sách entries chứa các Entry được đọc từ MyFS.
- Sau khi có được danh sách các Entry tồn tại trong Volume thì sẽ xuất ra danh sách các file đã được lưu trên các Entry.

```

def readAllEntries(path):
    sector = 0 # Starting sector
    entries = []

    while True:
        data = read512(path, sector)
        pos = 0

        while pos < len(data):
            entry = Entry()
            entryBlock = data[pos: pos + 80]

            # Populating Entry attributes
            entry.Name = entryBlock[:32].replace(b'\x00', b'')
            entry.Extended = entryBlock[32:37].replace(b'\x00', b'')
            entry.DateCreate = entryBlock[37:39]
            entry.LocationOfData = entryBlock[39:43].replace(b'\x00', b'')
            entry.Size = entryBlock[43:47].replace(b'\x00', b'')
            entry.State = entryBlock[47:48].replace(b'\x00', b'')
            entry.PassWord = entryBlock[48:]

            entries.append(entry)
            pos += 80

        sector += 1 # Move to the next sector

        # Assuming the end condition, you might have a specific criteria to stop the loop
        if not data:
            break

    return entries

```

```

def printEntries(entries):
    for i, entry in enumerate(entries, start=1):
        try:
            name = entry.Name.decode('utf-8')
        except UnicodeDecodeError:
            name = entry.Name.decode('utf-8', errors='replace') # Replace invalid characters

        entry_type = "Folder" if entry.State == b'\x01' else "File" if entry.State == b'\x02' else "Backup" if entry.State == b'\x02' else "Deleted"
        if (entry_type == "File" or entry_type == "Folder"):
            print(f"Name: {name}, Type: {entry_type}")

```


VI. Hướng dẫn sử dụng và hình ảnh chạy thử phần mềm

Dùng Visual Studio Code để mở folder chứa 3 file code

- Màn hình tạo Volume

```
----- .HPQ VOLUME -----
1: Nhập đường dẫn chứa volume để mở volume
2: Tạo mới volume
0: Thoát chương trình
Nhập lựa chọn của bạn: 2
Nhập đường dẫn mới: E:
Chọn dung lượng lưu trữ (1: 512MB, 2: 1GB, 3: 2GB): 1
1: Đặt mật khẩu
2: bỏ qua
1
Nhập password mới phải nhỏ hơn 16 kí tự(Enter bỏ qua): 123
```

- Boot Sector sau khi tạo Volume .HPQ

```
2E 48 50 51 00 03 20 00 00 00 00 00 00 00 00 00 .HPQ.. ..... Sector 8,196
A6 65 A4 59 20 42 2F 9D 41 7E 48 67 EF DC 4F B8 |eY B/.A~HgüO,
A0 4A 1F 3F FF 1F A0 7E 99 8E 86 F7 F7 A2 7A E3 J.?ÿ. ~Zt÷÷÷zã
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

- Màn hình nhập file vào Volume

```
1: Lưu trong thư mục
2: Không lưu trong thư mục
3: Tạo thư mục mới
0: Thoát chương trình
Nhập lựa chọn của bạn: 2
Nhập đường dẫn chứa file(ví dụ E:\report.pdf): D:\DO_AN_GK\t1.txt
Lưu file backup?
1. Lưu
Khác. Thoát
Nhập lựa chọn của bạn:
Nhập password mới phải nhỏ hơn 16 kí tự(Enter bỏ qua): 456
```

- Boot Sector file t1.txt sau khi nhập

```
74 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 t1..... Sector 8,197
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
2E 74 78 74 00 77 57 00 00 00 02 00 00 00 07 02 .txt.wW.....
B3 A8 E0 E1 F9 AB 1B FE 3A 36 F2 31 F6 76 F7 8B "ááù«.p:6ôlöv÷<
B3 0A 51 9D 2B 21 E6 C5 30 C0 EE E8 EB B4 A5 D0 ".Q.+!æÄ0Äiëë'¥D
```

- Boot Sector data của file t1.txt đã mã hoá

[illegible]

- Màn hình liệt kê danh sách file

```

E:/HPQ VOLUME
-----
1. Đổi password volume .HPQ
2. Liệt kê danh sách file / folder
3. Đổi password file / folder
4. Import file
5. Export file
6. Xoá file
0. Quay lại
Nhập lựa chọn của bạn: 2
Name: t1, Type: File
Name: t2, Type: File
Name: t3, Type: File

```

- Boot Sector vùng Backup của Volume

```

74 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 t3..... Sector 8,576
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
2E 74 78 74 00 79 57 00 00 00 00 00 00 08 E8 03 .txt.yW.....è.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

- Màn hình xuất file

```

Nhập tên thư mục/file cần xuất (0 để thoát): t1
Enter password for t1.txt: 456
Exported file: D:\DO_AN_GK\t1.txt
Press any key to continue . . .

```

- Màn hình xoá file

```

Nhập tên thư mục/file cần xuất (0 để thoát): t3
File 't3' được tìm thấy trong MyFS.
Nhập mật khẩu để xóa file: 456
Bạn có chắc chắn muốn xóa? (Y/N): Y
Bạn muốn xóa vĩnh viễn? (Y/N): Y

```

- Màn hình thay đổi password của Volume .HPQ

```

----- E:/.HPQ VOLUME -----
1. Đổi password volume .HPQ
2. Liệt kê danh sách file / folder
3. Đổi password file / folder
4. Import file
5. Export file
6. Xóa file
0. Quay lại
Nhập lựa chọn của bạn: 1
1: Thay đổi mật khẩu
2: Xóa mật khẩu
Nhập lựa chọn của bạn: 1
Nhập password mới phải nhỏ hơn 16 kí tự(Enter bỏ qua): 123456
Nhập mật khẩu hiện tại của Volume: 3434

```

VII. Tài liệu tham khảo

Tham khảo chung mô hình:

<https://github.com/kieuconghau/file-system>

https://github.com/ngohuudang2001/mid_term_OS

Tham khảo thuật toán AES:

<https://gist.github.com/henrique-marcomini-movile>