

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

-----



**Xây dựng mô hình và thiết kế kiến trúc tổ chức cho một hệ thống tập tin được chứa trong file MyFS.DRS**

**MÔN HỌC : AN TOÀN VÀ PHỤC HỒI DỮ LIỆU**

*Nhóm thực hiện:*

19120667 : Nguyễn Văn Thịnh

19120700 : Phạm Ngọc Truyền

19120712 : Phan Khắc Uy

**11/2022 – TPHCM**

## Mục lục

|       |  |    |
|-------|--|----|
| 2     | Lý thuyết.....   | 1  |
| 2.1   | Sơ đồ tổng quát : .....  | 1  |
| 2.1.1 | Vùng DATA: .....   | 1  |
| 2.1.2 | Vùng Hệ thống (System area): được chia làm 3 vùng cơ bản ..... | 1  |
| 2.2   | Đáp ứng tiêu chí : .....                                       | 3  |
| 2.2.1 | Tiêu chí bảo mật thông tin: .....                              | 3  |
| 2.2.2 | Tiêu chí an toàn dữ liệu: .....                                | 3  |
| 2.2.3 | Tiêu chí phục hồi dữ liệu:.....                                | 3  |
| 2.2.4 | Tiêu chí mã hóa dữ liệu: .....                                 | 3  |
| 2.2.5 | Tiêu chí số tập tin nhiều và đa dạng: .....                    | 4  |
| 2.2.6 | Tiêu chí hàm readSector và writeSector: .....                  | 4  |
| 3     | Thực hành .....  | 4  |
| 3.1   | Tạo/ định dạng volume MyFS.DRS.....                            | 4  |
| 3.2   | Thiết lập/ Đổi/ Kiểm tra mật khẩu truy xuất MyFS: .....        | 6  |
| 3.3   | Liệt kê danh sách tập tin trong MyFS : .....                   | 7  |
| 3.4   | Chép 1 tập tin vào MyFS: .....                                 | 8  |
| 3.5   | Thiết lập/ Thay đổi mật khẩu cho tập tin: .....                | 8  |
| 3.6   | Chép tập tin ra ngoài MyFS: .....                              | 9  |
| 3.7   | Xóa tập tin trong MyFS: .....                                  | 10 |

**Phân công công việc**

| MSSV     | Họ và Tên        | Công việc<br>phân công | Chưa hoàn thành                               |
|----------|------------------|------------------------|---|
| 19120667 | Nguyễn Văn Thịnh | Phần 2                 | Mã hóa mỗi lần ra<br>mỗi kết quả khác<br>nhau |
| 19120700 | Phạm Ngọc Truyền | 3.3 đến 3.7            | Chưa thể làm trên thư<br>mục con              |
| 19120712 | Phan Khắc Uy     | Báo cáo +<br>3.1&3.2   | Đã hoàn thành cơ bản<br>yêu cầu               |

## 2 Lý thuyết

### 2.1 Sơ đồ tổng quát :

|             |           |
|-------------|-----------|
| SYSTEM AREA | DATA AREA |
|-------------|-----------|

#### 2.1.1 Vùng DATA:

|          |          |      |     |     |  |  |             |           |
|----------|----------|------|-----|-----|--|--|-------------|-----------|
| Cluster0 | Cluster1 | .... | ... | ... |  |  | Cluster n-1 | Cluster n |
|----------|----------|------|-----|-----|--|--|-------------|-----------|

- Là vùng kéo dài từ sau vùng hệ thống đến hết file MyFS.
- Được chia thành nhiều cụm được gọi là Cluster . Mỗi cluster chứa nhiều sector( ở đây là 8) và mỗi sector là 1 block gồm 512 byte.
- Chứa toàn bộ nội dung của các tập tin. Các tập tin sẽ được lưu ở các cluster riêng biệt. Nếu tập tin nhỏ hơn 1 cluster vẫn được lưu trong 1 cluster. Ngược lại, nếu tập tin lớn hơn sẽ được lưu ở nhiều cluster (không nhất thiết là phải liên tiếp).

#### 2.1.2 Vùng Hệ thống (System area): được chia làm 3 vùng cơ bản .

|      |                |             |
|------|----------------|-------------|
| Boot | ClusterManager | FileManager |
|------|----------------|-------------|

**Boot** : Là vùng chứa toàn bộ thông tin của file MyFS bao gồm :

- Tên volume (ở đây là MyFS) và tên phần mở rộng volume (ở đây là DRS).
- Kích thước của volume : Kích thước của volume có thể rất lớn nên ta dùng 5 byte để lưu bắt đầu từ offset 0h.
- Lưu giữ kích thước 1 sector (512 byte) ta sẽ dùng 10 bit, số sector của 1 cluster (8 sector) ta sẽ dùng 6 bit sau phần lưu giữ kích thước sector. Do đó tổng cộng 2 phần này được lưu 2 byte bắt đầu từ offset 5h.
- Kích thước của vùng boot, vùng ClusterManager, vùng FileManager lưu lần lượt mỗi cái 3 byte bắt đầu tại offset 7h. Có kích thước volume và 3 kích thước này ta tính được kích thước vùng Data.

- Có password volume (người dùng có thể đặt hoặc không): Dùng 4 byte bắt đầu từ offset 10h. 2 bit đầu tiên ta lưu trạng thái có cài mật khẩu hay không, 30 bit còn lại ta lưu mật khẩu dưới dạng hash.
- Lưu thông tin và số lượng các loại và tên mở rộng mà MyFS hỗ trợ (ở đây gồm các loại pdf, docx, dat, pptx, txt, xlsx,...). Mỗi tên mở rộng được lưu trong 6 byte, MyFS hỗ trợ 10 loại file nên dùng 60 byte bắt đầu từ offset 14h.

### **ClusterManager:**

- Nằm ngay sau vùng Boot, là 1 mảng có số phần tử bằng số cluster trong MyFS, mỗi phần tử quản lý trạng thái 1 cluster.
- Mỗi phần tử có ý nghĩa khác nhau: rỗng (-1), kết thúc file (-2) cluster bị xóa thường (-3), cluster bị hỏng (-4) hay số dương K là nội dung file tiếp theo chứa ở cluster K.

### **FileManager:**

- Nằm ngay sau vùng ClusterManager và trước vùng Data, chứa nhiều thông tin :
  - ⇒ Tên tập tin : gồm tên file và phần mở rộng: Tên file có kiểu chuỗi, tối đa 16 ký tự (nếu tên dài quá 16 sẽ lấy 16 ký tự đầu), dùng 16 byte để lưu. Phần mở rộng tối đa 10 ký tự. Không cho phép 2 tập tin có tên và phần mở rộng giống nhau, dùng 10 byte tiếp theo để lưu
  - ⇒ Kích thước tập tin : Ta dùng 3 byte tiếp sau phần mở rộng tập tin để lưu phần này.
  - ⇒ Thời điểm tạo ra tập tin (Ngày / tháng –/ năm giờ : phút) : Ngày từ 1 đến 31 nên dùng 5 bit để lưu, tháng từ 1 đến 12 nên dùng 4 bit, năm ta lưu 3 số cuối nên dùng 4 bit, giờ từ 0 đến 23 ta dùng 5 bit và phút từ 0 đến 60 ta dùng 6 bit. Tổng cộng ta dùng 3 byte để lưu thời điểm tạo ra tập tin.

- ⇒ Password tập tin (người dùng có thể đặt hoặc không): Dùng 4 byte. 2 bit đầu tiên ta lưu trạng thái có cài mật khẩu hay không, 30 bit còn lại ta lưu mật khẩu dưới dạng hash.
- ⇒ Cluster bắt đầu chứa nội dung tập tin : Số cluster có thể rất lớn nên ta dùng khoảng 3 byte để lưu vị trí cluster bắt đầu lưu nội dung file.
- ⇒ Tên thư mục cha : Là chuỗi tối đa 16 kí tự, ta dùng 16 byte để lưu tên thư mục cha.

## 2.2 Đáp ứng tiêu chí :

### 2.2.1 Tiêu chí bảo mật thông tin:

Volume có cung cấp mật khẩu volume và mỗi tập tin được cung cấp vùng nhớ để lưu mật khẩu tập tin (người dùng có thể đặt mật khẩu tùy sở thích).

### 2.2.2 Tiêu chí an toàn dữ liệu:

ClusterManager sẽ xác định cluster trống để lưu dữ liệu vào. Một cluster tương đối lớn (4MB) nên các nội dung tập tin sẽ được lưu liên tiếp (trừ tập tin quá lớn).

### 2.2.3 Tiêu chí phục hồi dữ liệu:

ClusterManager[K] = -3 thì file có thể khôi phục, ClusterManager[K] = -1 thì file không thể phục. Lúc tìm cluster trống, ưu tiên các cluster rỗng ClusterManager[K] = -1 nên hạn chế việc ghi đè.

### 2.2.4 Tiêu chí mã hóa dữ liệu:

- Các tập tin đã được cài mật khẩu mới có thể mã hóa.
- Trước khi mã hóa, cần xác nhận mật khẩu, mật khẩu xác nhận sẽ được hash và so sánh với mật khẩu tập tin đầu này dẫn đến không bị rò rỉ mật khẩu.

### 2.2.5 Tiêu chí số tập tin nhiều và đa dạng:

MyFS có dung lượng lớn nên có thể lưu tập tin với số lượng lớn và hỗ trợ nhiều phân mở rộng tập tin đảm bảo sự đa dạng.

### 2.2.6 Tiêu chí hàm readSector và writeSector:

Ta thiết kế 1 sector 512 byte :

- char\* readSector(fstream&file, int pos) sẽ đến vị trí pos, sau đó đọc và trả về 512 byte.
- void writeSector(fstream&file, int pos, char\* sector) sẽ đến vị trí pos sau đó ghi đè 1 sector vào file.

## 3 Thực hành

### 3.1 Tạo/ định dạng volume MyFS.DRS

Bước 1 : Thiết lập thông số quan trọng :

```
// Thiết lập các thông số để định dạng volume
#define BYTES_PER_SECTOR 512
#define SECTORS_PER_CLUSTER 2
#define SECTORS_OF_FAT 5
#define SECTORS_OF_ENTRY 2
#define CLUSTERS_OF_VOLUME 2000
#define SECTORS_BEFORE_FAT 1
#define CLUSTER_EMPTY -1
#define CLUSTER_NOT_EMPTY -2
```

Bước 2 : Tạo đối tượng để lưu thông tin volume từ struct Volume và đối tượng entry để lưu thông tin file từ struct Entry.

```

// Tạo ra đối tượng thông tin của 1 volume
Volume* createVolumeInfo(string name, string extension) {
    Volume* volume = new Volume;
    volume->name = name;
    volume->extension = extension;
    volume->password = "";
    volume->sectors_of_entry = SECTORS_OF_ENTRY;
    volume->bytes_per_sector = BYTES_PER_SECTOR;
    volume->sectors_per_cluster = SECTORS_PER_CLUSTER;
    volume->sectors_of_fat = SECTORS_OF_FAT;
    volume->clusters_of_volume = CLUSTERS_OF_VOLUME;
    return volume;
}

// Tạo ra đối tượng thông tin của 1 entry chính
Entry* createEntry(string name, string extension) {
    Entry* entry = new Entry;
    entry->name = name;
    entry->extension = extension;
    entry->password = "";
    entry->size = 0;
    entry->cluster_begin = CLUSTER_EMPTY;
    return entry;
}

```

Bước 3:

Viết thông tin volume lưu vào vùng boot :

```

// Viết thông tin volume vào file
file.seekp(0,ios::beg);
file.write((char*)vol, sizeof(Volume));

```

Bảng fat lưu vùng clusterManager :

```

// Viết thông tin bảng fat vào file
file.seekp(SECTORS_BEFORE_FAT*BYTES_PER_SECTOR, ios::beg);
file.write((char*)fat, CLUSTERS_OF_VOLUME);

```

Entry lưu vào vùng fileManager:

```

// Viết thông tin bảng entry vào file
int num_entries = SECTORS_OF_ENTRY * BYTES_PER_SECTOR / sizeof(Entry);
for (int i = 0; i < num_entries; i++) {
    Entry* entry = createEntry("", "");
    entries.push_back(entry);
    file.seekp((SECTORS_BEFORE_FAT + SECTORS_OF_FAT ) * BYTES_PER_SECTOR + (static_cast<unsigned __int64>(i) + 1) * sizeof(Entry), ios::beg);
    file.write((char*)entry, sizeof(Entry));
}


```



Và vùng data lưu vào file :

```
// Viết vùng data vào file
int system_bytes = (SECTORS_BEFORE_FAT+ SECTORS_OF_FAT + SECTORS_OF_ENTRY) * BYTES_PER_SECTOR;
for (int i = 0; i < CLUSTERS_OF_VOLUME; i++) {
    for (int j = 0; j < SECTORS_PER_CLUSTER; j++) {
        int* block = new int[BYTES_PER_SECTOR];
        int pos = system_bytes + (i * SECTORS_PER_CLUSTER + j) * BYTES_PER_SECTOR;
        file.seekp(pos, ios::beg);
        file.write((char*)block, BYTES_PER_SECTOR);
    }
}
```

Bước 4 : Kết quả:

 MyFS.DRS

11/8/2022 12:58 PM

DRS File

2,004 KB

### 3.2 Thiết lập/ Đổi/ Kiểm tra mật khẩu truy xuất MyFS:

Ta đọc vùng boot ,thiết lập mật khẩu rồi viết lại vùng boot.

- Thiết lập mật khẩu:

```
1. Thiet lap mat khai
2. Kiem tra mat khai
3. Doi mat khai
0. Thoat
--> Nhap lua chon: 1
Nhap mat khai thiet lap volume: 700
Mat khai thiet lap thanh cong!
```

- Kiểm tra mật khẩu :

```
1. Thiet lap mat khai
2. Kiem tra mat khai
3. Doi mat khai
0. Thoat
--> Nhap lua chon: 2
Nhap mat khai MyFS.DRS: 712
Mat khai khong chinh xac!
```

```
1. Thiet lap mat khai
2. Kiem tra mat khai
3. Doi mat khai
0. Thoat
--> Nhap lua chon: 2
Nhap mat khai MyFS.DRS: 700
Mat khai chinh xac!
```

- Đổi mật khẩu : nếu chưa thiết lập thì thông báo.

```
1. Thiet lap mat khau
2. Kiem tra mat khau
3. Doi mat khau
0. Thoat

--> Nhap lua chon: 3

Mat khau chua duoc thiet lap.
Xin vui long thiet lap mat khau truoc!
```

```
1. Thiet lap mat khau
2. Kiem tra mat khau
3. Doi mat khau
0. Thoat

--> Nhap lua chon: 3

Nhap mat khau hien tai: 712

Mat khau khong khop voi mat khau hien tai!
```

```
1. Thiet lap mat khau
2. Kiem tra mat khau
3. Doi mat khau
0. Thoat

--> Nhap lua chon: 3

Nhap mat khau hien tai: 700

Nhap mat khau moi: 667

Mat khau da duoc thay doi!
```

### 3.3 Liệt kê danh sách tập tin trong MyFS :

- Ta đọc vùng FileManager rồi in các file có trong MyFS
- Hiện tại chưa import nên không có file nào. Chức năng này sẽ lồng ghép vào các chức năng sau.

```
Nhap lua chon chuc nang: 3

Danh sach tap tin hien co trong volume:

Hien khong co tap tin nao trong volume!!!
```

### 3.4 Chép 1 tập tin vào MyFS:

- Bước 1: Tạo 1 entry có tên file và phần mở rộng.

```
Entry* entry_file = createEntry("", "");
pair<string, string> res = readFilePath(filepath);
entry_file->name = res.first;
entry_file->extension = res.second;
```

- Bước 2: Tìm cluster trống tại clusterManager và viết nội dung file vào cluster trống đó.
- Bước 3: Cập nhật lại vùng clusterManager và fileManager.

```
// Ghi thông tin file vào entry trống trong volume
entries[entry_pos] = entry_file;
int empty_entry_pos = (SECTORS_BEFORE_FAT + SECTORS_OF_FAT ) * BYTES_PER_SECTOR + entry_pos * sizeof(Entry);
file.seekp(empty_entry_pos, ios::beg);
file.write((char*)entries[entry_pos], sizeof(Entry));

// Cập nhật bảng fat vào file
file.seekp(SECTORS_BEFORE_FAT * BYTES_PER_SECTOR, ios::beg);
file.write((char*)fat, CLUSTERS_OF_VOLUME);
```

- Kết quả: Khi import file E:\test.txt vào

```
Nhap duong dan cua file can chep vao volume(Ex: E:\Test.txt): E:\test.txt
Import tap tin vao volume thanh cong!
```

```
Nhap lua chon chuc nang: 3
Danh sach tap tin hien co trong volume:
test.txt
```

### 3.5 Thiết lập/ Thay đổi mật khẩu cho tập tin:

Ta đọc vùng fileManager, kiểm tra tập tin có trong danh sách hay không, thiết lập mật khẩu rồi viết lại vào file.

- Thiết lập :

```
1. Thiet lap mat khau tap tin
2. Doi mat khau tap tin
0. Thoat
--> Nhap lua chon: 1
Nhap ten file can thiet lap mat khau (Ex:test.txt) : test.txt
Nhap mat khau thiet lap cho tap tin: 700
Mat khau thiet lap thanh cong!
```

- Thay đổi:

```
1. Thiet lap mat khau tap tin
2. Doi mat khau tap tin
0. Thoat

--> Nhap lua chon:
2

Nhap ten file can thay doi lap mat khau (vd:test.txt) : test.txt

Nhap mat khau hien tai: 712

Mat khau khong khop voi mat khau hien tai!
```

```
1. Thiet lap mat khau tap tin
2. Doi mat khau tap tin
0. Thoat

--> Nhap lua chon: 2

Nhap ten file can thay doi lap mat khau (vd:test.txt) : test.txt

Nhap mat khau hien tai: 700

Nhap mat khau moi: 667

Mat khau da duoc thay doi thanh cong!
```

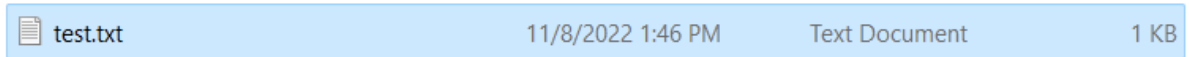
### 3.6 Chép tập tin ra ngoài MyFS:

- Bước 1: Kiểm tra file trong vùng fileManager.
- Bước 2: Có được cluster , vào cluster ghi nội dung file ra ngoài

```
for (int i = 0; i < SECTORS_PER_CLUSTER; i++) {
    char* sector = new char[BYTES_PER_SECTOR];
    int read_pos = system_bytes + (cluster_pos * SECTORS_PER_CLUSTER + i) * BYTES_PER_SECTOR;
    //file.seekg(read_pos, ios::beg);
    //file.read(sector, BYTES_PER_SECTOR);
    sector = readSector(file, read_pos);
    // Size lớn hơn số byte của 1 sector thì ghi dữ liệu 1 sector
    if (size_file > BYTES_PER_SECTOR) {
        outfile.write(sector, BYTES_PER_SECTOR);
        size_file -= BYTES_PER_SECTOR;
    }
    // Size nhỏ hơn số byte 1 sector thì ghi từng byte xong thoát khỏi vòng lặp và kết thúc
    else {
        outfile.write(sector, size_file);
        size_file = 0;
        break;
    }
}
```

- Kết quả:

```
Nhap ten file trong volume can chep ra ngoai (Ex: test.txt): test.txt
Tap tin test.txt da xuat thanh cong!
```



### 3.7 Xóa tập tin trong MyFS:

- Bước 1: Kiểm tra file có trong vùng fileManager
- Bước 2: Có vị trí cluster, đến vùng clusterManager cập nhật cluster rỗng.

```
int cluster = entries[entry_pos_file]->cluster_begin;
while (cluster != CLUSTER_EMPTY) {
    if (fat[cluster] == CLUSTER_NOT_EMPTY)
    {
        fat[cluster] = CLUSTER_EMPTY;
        break;
    }
    int next_cluster = fat[cluster];
    fat[cluster] = CLUSTER_EMPTY;
    cluster = next_cluster;
}
```

- Bước 3: Cập nhật lại vùng fileManager và vùng clusterManager

```
// Viết entry trống vào file
int entry_pos = (SECTORS_BEFORE_FAT + SECTORS_OF_FAT)*BYTES_PER_SECTOR + entry_pos_file * sizeof(Entry);
Entry* del_entry = createEntry("", "");
entries[entry_pos_file] = del_entry;
file.seekp(entry_pos, ios::beg);
file.write((char*)del_entry, sizeof(Entry));

// Cập nhật lại bảng fat
file.seekp(SECTORS_BEFORE_FAT*BYTES_PER_SECTOR, ios::beg);
file.write((char*)fat, CLUSTERS_OF_VOLUME);
```

- Kết quả:

```
Nhap ten file trong volume can xoa (Ex: test.txt): test.txt
Xoa tap tin test.txt thanh cong!

Danh sach tap tin hien co trong volume:
Hien khong co tap tin nao trong volume!!!
```