A picture containing logo

Description automatically generated

Khoa Công nghệ Thông tin

**Final project:**

**Thiết kế tập tin đảm bảo an toàn & phục hồi dữ liệu.**

Bộ môn An toàn và Phục hồi Dữ liệu

A picture containing outdoor, sign

Description automatically generatedGiảng viên

ThS. Thái Hùng Văn

Sinh viên

19120659 – Phạm Văn Thành

20120382 – Hoàng Thu Thủy

MỤC LỤC

[1 Đánh giá kết quả 3](#_Toc154933976)

[1.1 Đánh giá thành viên 3](#_Toc154933977)

[1.2 Đánh giá mức độ hoàn thành 4](#_Toc154933978)

[2 Nội dung chi tiết 6](#_Toc154933979)

[2.1 Phần lý thuyết 6](#_Toc154933980)

[2.1.1 Chi tiết thiết kế file 6](#_Toc154933981)

[2.1.2 Khả năng đáp ứng yêu cầu của thiết kế 10](#_Toc154933982)

[2.2 Chương trình demo 12](#_Toc154933983)

[2.2.1 Tạo file 13](#_Toc154933984)

[2.2.2 Thiết lập / Đổi mật khẩu truy xuất MyFS 15](#_Toc154933985)

[2.2.3 Đặt keyEncrypt để mã hóa dữ liệu quan trọng 17](#_Toc154933986)

[2.2.4 Nhập dữ liệu cho các danh sách 17](#_Toc154933987)

[2.2.5 Đọc thông tin từ file 19](#_Toc154933988)

[2.2.6 Xóa phần tử trong danh sách 20](#_Toc154933989)

[2.2.7 Chỉnh sửa phần tử trong danh sách 21](#_Toc154933990)

[3 Danh mục tham khảo 24](#_Toc154933991)

# Đánh giá kết quả

## Đánh giá thành viên

|  |  |  |  |
| --- | --- | --- | --- |
| Bảng phân công công việc | | | |
| MSSV | Họ và tên | Công việc | Mức độ đóng góp |
| 19120659 | Phạm Văn Thành | – Xây dựng thiết kế kiến trúc tập tin  - Thiết kế mã hóa đảm bảo an toàn dữ liệu  – Chép (Import), tổ chức dữ liệu cho danh sách Giáo Viên  – Đọc/ chỉnh sửa/ xóa dữ liệu trên danh sách Giáo viên  – Thiết kế mật khẩu động và thời gian chờ | 50% |
| 20120382 | Hoàng Thu Thủy | – Xây dựng thiết kế kiến trúc tập tin  – Đặt / Đổi mật khẩu truy xuất cho 1 tập tin  – Chép (Import), tổ chức dữ liệu cho danh sách Sinh viên  – Đọc/ chỉnh sửa/ xóa dữ liệu trên danh sách Sinh viên  – Xây dựng command line | 50% |

## Đánh giá mức độ hoàn thành

|  |  |  |  |
| --- | --- | --- | --- |
| Bảng đánh giá mức độ hoàn thành MÔ HÌNH LÝ THUYẾT | | | |
| Tiêu chí | Tên chức năng/yêu cầu | Mức độ hoàn thành | Hạn chế |
| 1 | Thiết kế bao gồm:  - DS học sinh /sinh viên (nhiều phần tử và thường có nhu cầu thêm /xóa /sửa)  - DS giáo viên (ít phần tử và ít có nhu cầu thêm /xóa /sửa) | 100% |  |
| 2 | Tránh tối đa việc dồn dữ liệu | 100% |  |
| 3 | Các phần tử được xóa sẽ không xóa hẳn để có khả năng phục hồi lại (ngoại trừ tình huống đặc biệt cần phải xóa hẳn, các phần tử đã xóa quá lâu cũng không cần phải phục hồi) | 100% |  |
| 4 | Mỗi cá nhân tổ chức minh họa vài thông tin cơ bản (Mã, Họ Tên, Ngày sinh, Ngày tham gia, Số ĐT, Số CCCD,…), trong đó số CCCD và số ĐT cần bảo mật | 100% |  |
| 5 | Có các thông tin quản lý cần thiết, như ngày tạo lập, thời điểm cập nhật, công thức /key mã hóa /giải mã, … | 100% |  |
| 6 | Cần có cơ chế kiểm tra mật khẩu động hoặc passkey mỗi khi file được mở, có khống chế thời gian (nhập sai nhiều lần thì phải đợi một thời gian sau mới có thể nhập tiếp, vẫn sai nữa thì thời gian đợi bị tăng thêm) | 100% |  |
| 7 | Xây dựng chương trình demo minh họa với ít nhất các thao tác tạo /thêm /xóa /sửa phần tử trên cả 2 DS, liệt kê 1 đoạn trong DS, đổi mật khẩu /cơ chế mã hóa. | 90% | Chương trình mới hỗ trợ liệt kê tất cả các phần tử trong danh sách 1 hoặc 2 chứ chưa hỗ trợ liệt kê một đoạn cụ thể. |

# Nội dung chi tiết

## Phần lý thuyết

### Chi tiết thiết kế file

**Tổng quát thiết kế:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FILE\_INFO  (metadata) | LIST\_TABLE | | | | | | | | | | | | LIST\_DATA | | |
| LIST\_TABLE 1 | | | | | | LIST\_TABLE 2  (BACKUP) | | | | | |
| List2\_data | Empty | List2\_data(+empty behind) |
| 96 bytes | List\_2 | | | List1 | | | List\_2 | | | List1 | | | 256\* 100 bytes | 256\*100  bytes | 256\*n  bytes |
| [0] | … | [200] | [0] | … | [1000] | [0] | … | [200] | [0] | … | [1000] |
| 16 \* 200 + 16\*1000 bytes | | | | | | 16 \* 200 + 16\*1000 bytes | | | | | |

* Mỗi *phần tử trong list\_table* =16 bytes, mỗi phần tử trong *List\_\_data* = 256 bytes.
* Mỗi List\_Table ước tính gồm khoảng 200 phần tử cho list\_2 và khoảng 1000 phần tử cho list\_1. Tổng kích thước vùng LIST\_TABLE này là 38KB
* Kích thước LIST\_DATA sẽ bao gồm vùng cho list 2 (giáo viên) khoảng 100 phần tử và phần Empty để dành khoảng trống cho list2 trong trường hợp còn thiếu, phần này cũng đủ chỗ cho 100 phần tử. Mỗi phần tử trong list dùng để mô tả thông tin về giáo viên/ học sinh sẽ được quy định là 256 bytes

**Mô tả chi tiết từng vùng:**

1. **Vùng File INFO (METADATA)**

|  |  |  |
| --- | --- | --- |
| **FILE INFO** | | |
| Địa chỉ | Kích thước (bytes) | Mô tả |
| 0 | 2 | File signature: 0x7777 |
| 2 | 16 | Password của File(đã được hash). Nếu file không dùng mật khẩu, vùng này = 0x00 |
| 18 | 2 | Trạng thái của phần tử trong list\_table bị xóa sớm nhất |
| 20 | 4 | Kích thước của file |
| 24 | 4 | Trạng thái của phần tử trong list\_Data bị xóa sớm nhất |
| 28 | 4 | Trạng thái của phần tử trong list\_Data tiếp theo sẽ xóa |
| 32 | 2 | Trạng thái của phần tử trong list\_table tiếp theo sẽ xóa |
| 34 | 8 | Ngày tạo file |
| 42 | 8 | Ngày cập nhật gần nhất của file |
| 50 | 8 | Key mã hóa |
| 58 | 38 | Có thể mã hóa toàn bộ vùng metadata này tại đây nếu cần thiết |

1. **Vùng LIST\_TABLE**

* Gồm 2 bảng list Table:

+ List Table 1 là bản chính,

+ List Table 2 là bản backup.

* Có thể linh động thay đổi vai trò tùy trường hợp
* Mỗi List Table sẽ chứa thông tin của list 1 và list 2. Mỗi phần tử trong list table sẽ mô tả thông tin của một phần tử tương ứng trong list\_data, và kích thước của phần tử mô tả này sẽ là 16 bytes

|  |  |  |
| --- | --- | --- |
| **Phần tử trong List\_table** | | |
| Địa chỉ | Kích thước (bytes) | Mô tả |
| 0 | 2 | Trạng thái của phần tử trong list   * 1: Đang sử dụng * 0: Chưa sử dụng * 2 - 1000: Đã xóa. Dựa vào “Trạng thái của phần tử list\_talbe bị xóa sớm nhất” đã lưu trữ ở *File Info* để biết danh sách phần tử nào đã được xóa (mục đích là để sau này là tìm lại một phần tử nào đó trong danh sách phần tử đã bị xóa tạm thời) |
| 2 | 1 | Phần tử này thuộc list 1 hay list 2:   * 10h: List 1 * 20h: List 2 |
| 3 | 2 | Số thứ tự trong danh sách của list , nếu chưa dùng giá trị này = 0. |
| 5 | 4 | Vị trí của phần tử trên vùng data. |
| 9 | 4 | Kích thước của phần tử |
| 13 | 3 | Checksum |

1. **Vùng LIST\_DATA**

* Vùng này sẽ chia làm 3 vùng / khu vực:

+ Vùng để lưu trữ dữ liệu cho list 2 ( giáo viên)

+ Vùng trống (empty) để backup trong trường hợp vùng cho giáo viên bị dùng hết

+ vùng để lưu trữ list 1 (học sinh): nằm ở cuối file giúp hạn chế các thao tác gộp, dồn dữ liệu

* Trong mỗi vùng này thông tin để lưu trữ giáo viên/ sinh viên ước tính sẽ rơi vào khoảng 256 bytes bao gồm: Mã, Họ Tên, Ngày sinh, Ngày tham gia, Số ĐT, Số CCCD,…
* Với kích thước định sẵn trên, ta có thể xác định được kích thước khu vực lưu trữ list 2 (giáo viên), khu vực trống backup và tìm ra vị trí để bắt đầu lưu thông tin của list 1 ( học sinh/sinh viên)
* Số lượng phần tử ước tính cho list2 là 100 phần tử và phần trống để thêm giáo viên cũng được thiết kế là 100 phần tử.

|  |  |  |
| --- | --- | --- |
| **DATA TRONG LIST** | | |
| Địa chỉ | Kích thước (bytes) | Mô tả |
| 0 | 4 | Trạng thái của block   * 1: Đang dùng * 0: Chưa dùng * 2 –> n: Đã xóa. Dựa vào “Trạng thái của phần tử bị xóa sớm nhất” đã lưu trữ ở File Info để biết phàn tử nào đã được xóa. Sau này ta có thể tạo một danh sách các phần tử bị xóa để truy xuất nếu muốn tìm một đối tượng nào đó để khôi phục) |
| 4 | 4 | Địa chỉ của phần tử tiếp theo trong list. Nếu đây là phần tử cuối của list, giá trị vùng này là 0xFFFFFFFF. |
| 8 | 248 | Nội dung phần tử trong list |

### Khả năng đáp ứng yêu cầu của thiết kế

1. Cần tránh tối đa việc dồn DL

Đáp ứng: 100%.

Lí do: Thiết kế trên đã đưa danh sách giáo viên lên đầu (ít phần tử và ít có nhu cầu thêm/xóa/sửa) và đưa danh sách sinh viên ra phía sau(nhiều phần tử và thường có nhu cầu thêm/xóa/sửa) và tạo khoảng trống phòng trường hợp thiếu không gian cho danh sách giáo viên: điều này đã giúp hạn chế tối thiểu dữ liệu bị dồn hay thay đổi địa chỉ liên tục khi làm việc với một danh sách bất kỳ trong hai danh sách đưa ra.

1. **Các phần tử được xóa sẽ không xóa hẳn để có khả năng phục hồi lại (ngoại trừ tình huống đặc biệt cần phải xóa hẳn, các phần tử đã xóa quá lâu cũng không cần phải phục hồi).**

Đáp ứng 100%.

Lí do:

* Mỗi phân tử đều thuộc một bảng quản lý phần tử (list\_table). Khi lựa chọn không xóa hẳn, bảng quản lý và list\_data chỉ thay đổi bit nhận dạng hay nói cách khác: sẽ chỉ đánh dấu đây là phần tử đã được xóa mà không xóa hẳn dữ liệu trong List\_data. Ngoài ra các phần tử bị xóa đều được gắn số thứ tự để giúp việc truy xuất lại danh sách các phần tử bị xóa tạm thời có thể nhanh chóng tìm ra và phục hồi lại.
* Khi xóa hẳn phần tử (không phục hồi): Đặt tất cả các bytes của list\_table và list\_data tương ứng của của phần tử đó về trạng thái ban đầu 0x00. Điều này khiến phần tử mất dữ liệu hoàn toàn và không thể phục hồi.

1. **Mỗi cá nhân có thể tổ chức minh họa vài thông tin cơ bản (Mã, Họ Tên, Ngày sinh, Ngày tham gia, Số ĐT, Số CCCD,…), trong đó số CCCD và số ĐT cần bảo mật.**

Đáp ứng 100%.

Lí do: Trong metadata (FileInfo) ngoài lưu trữ mật khẩu của file còn có một nơi lưu trữ key để mã hóa. Vậy nên các thông tin quan trọng khi nhập vào như CCCD, số điện thoại đều được mã hóa trước khi lưu vào file thông qua cơ chế mã hóa AES.

1. Có các thông tin quản lý cần thiết, như ngày tạo lập, thời điểm cập nhật, key mã hóa

Đáp ứng 100%.

Lí do: Metadata đã được thiết kế để lưu các thông tin cần thiết như: kích thước file, ngày tạo file, thời gian cập nhật gần nhất, key mã hóa ở đây sẽ yêu cầu người dùng nhập lúc người dùng tạo file, nếu người dùng không tạo key, key sẽ được set mặc định.

1. Cần có cơ chế kiểm tra mật khẩu động hoặc passkey mỗi khi file được mở, có khống chế thời gian (nhập sai nhiều lần thì phải đợi một thời gian sau mới có thể nhập tiếp, vẫn sai nữa thì thời gian đợi bị tăng thêm)

Đáp ứng: 100%.

Lí do:

B1: Người dùng có thể tạo mật khẩu cho file. Sau khi tạo hệ thống sẽ hash mật khẩu này là lưu lại vào metadata.

B2:*Hệ thống tạo cơ chế mật khẩu động*

Mật khẩu động có cấu trúc như sau: <hh>< password ><mm>.

*Ví dụ:* Thời gian hiện tại là 15:27 và password người dùng là **1234**

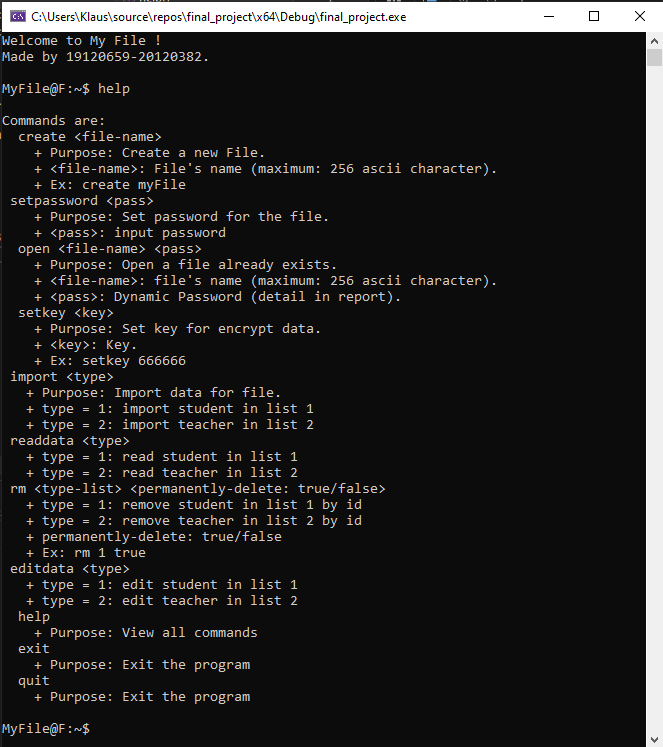
Người dùng sẽ phải nhập: **15123427.**

B3: Hệ thống kiểm tra số lần người dùng nhập sai và tang thời gian đợi để mở khóa. Người dùng nhập mật khẩu mỗi lần nhập sai, thời gian đợi sẽ tăng lên từ 2 giây, 4 giây, 8 giây, …nếu nhập sai quá 5 lần chương trình sẽ tự động đóng.

## Chương trình demo

* Phần này có những điểm tinh giản so với thiết kế mô hình lý thuyết như:
* Số lượng listTable được rút gọn lại chỉ còn 94 phần tử
* List 2 - giáo viện sẽ lưu ở các vị trí (0-> 50)
* List 1- sinh viện sẽ được lưu ở vị trí 50 trở đi.

- Người dung có thể sử dụng `help` để xem cách dùng các lệnh:



### Tạo file

– Tạo volume có tên myfile.drs:

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

File được thông báo tạo thành công kèm theo các thông tin lúc mới được khởi tạo

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, số

Mô tả được tạo tự động

Khi vừa khởi tạo, tất cả các bytes đều là 0, trừ một số vùng như:

* Signature: 0x6666
* ListTable xóa đầu tiên / tiếp theo: 0x02
* Kích thước volume : 0x0200 = 512 bytes
* ListData xóa đầu tiên/ xóa tiếp theo: 0x02, 0x02
* Ngoài ra thông tin vê ngày giờ tạo file cũng được lưu kéo dài xuống bên dưới nhưng cho cơ chế lưu của thư viện time.h nên không thể nhìn ở bản rõ

Do chưa đặt mật khẩu hay key mã hóa nên các vùng này hiện vẫn là các byte 0.

### Thiết lập / Đổi mật khẩu truy xuất MyFS

B1: Mở file bằng câu lệnh: open <tên-file>.

Do chưa thiết lập mật khẩu nên không yêu cầu nhập.

B2: Đặt mật khẩu cho file mới tạo là 123456.

+ Mật khẩu sẽ hash và được cập nhật cho file:

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

– Đổi mật khẩu thành: 19001009

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

Đổi mật khẩu thành công.

-Mật khẩu động được yêu cầu khi lần sau người dùng mở file:

+ Với mỗi lần mở sai người dùng phải đợi một khoảng thời gian nhất định:

+ ở đây mặc dù mật khẩu nhập đúng như lúc khai báo nhưng vẫn không thể vào được là do chưa tuân thử cơ chế mật khẩu động:

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

-Mật khẩu động được thiết kế theo cơ chế như đã nói ở phần 2.1.2 , cụ thể được tạo theo công thức thời gian như sau: <hh>< password ><mm>.

+ Cụ thể thời gian nhập mật khẩu lúc chạy chương trình này là 15h36. Vậy mật khẩu cần nhập sẽ phải là: 151900100936.

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, Phông chữ

Mô tả được tạo tự động

### Đặt keyEncrypt để mã hóa dữ liệu quan trọng

* Người dùng có thể đặt KeyErypt hoặc nếu quên không đặt khi nhập dữ liệu giáo viên/sinh viên vào hệ thống sẽ tự dùng key set mặc định là dãy số 123456

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

### Nhập dữ liệu cho các danh sách

-Người dùng thêm phần tử bằng câu lệnh: import <type>

+ trong đó nếu type = 1 thì có nghĩa phần tử sẽ được thêm cho học sinh

+ Type = 2: thêm danh sách cho giáo viên

1. Danh sách giáo viên

* Hệ thống yêu cầu nhập thông tin từ bàn phím:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Hiển thị thông tin sau khi được mã hóa thông tin quan trọng như số điện thoại và căn cước công dân:

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

1. Danh sách học sinh

* Tương tự đối với học sinh nhưng các sector dùng để lưu thông tin sẽ ở xa hơn trong file ở cả tableList và dataTable:

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

### Đọc thông tin từ file

* Người dùng sử dụng cú pháp: readdata <type>

+ Trong đó nếu type = 1 thì hệ thống sẽ hiển thị danh sách học sinh

+ Type = 2: Hiển thị danh sách giáo viên:

* Hiển thị danh sách giáo viên sau khi thêm một số phần tử:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

-Hiển thị danh sách sinh viên:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

### Xóa phần tử trong danh sách

1. **Xóa thường(vẫn có thể khôi phục)**

* Xóa thường thông tin của Giáo Viên “Nguyen”

Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

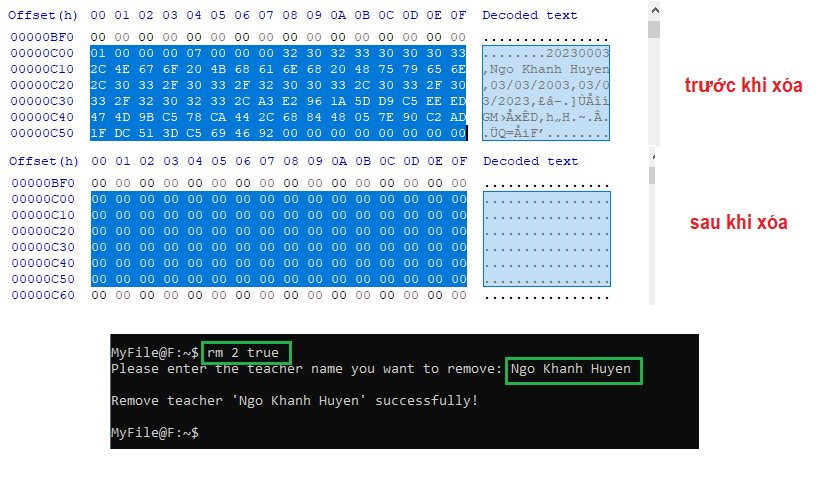
Các thông tin của người này trong file này vẫn còn, chỉ bị đánh dấu sang trạng thái khác (trạng thái đã xóa) đi cùng số thứ tự

* Hiện tại đây là Người bị xóa đầu tiên nên sẽ bắt đầu từ số 2.
* Cách này giúp cho việc phục hồi dữ liệu sau này trở nên dễ dàng hơn

1. **Xóa không phục hồi**

– Xóa không phục hồi giáo viên tên:”Ngo Khanh Huyen”

Để xóa không phục hồi, dùng lệnh rm <type> true hoặc rm <type> 1



Thông tin trên listTable và DataTable liên quan tới người này hoàn toàn biến mất và các byte được set lại thành 00.

* Sau khi xóa 2 người liên tục lúc này danh sách giáo viên(list 2) chỉ còn lại một người:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

### Chỉnh sửa phần tử trong danh sách

Thông tin trên listTable và DataTable liên quan tới người này hoàn toàn biến mất và các byte được set lại thành 00.

Tương tự các chức năng khác người dùng chọn lệnh : editdata <type> trong đó vẫn quy định type = 1 là danh sách học sinh, type 2 là danh sách giáo viên

* Dữ liệu danh sách 1 trước khi chỉnh sửa:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Người dùng nhập ID để tìm kiếm đối tượng cần sửa sau đó nhập các thông tin cần update:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Thông báo kết quả sau khi update:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* Kiểm tra lại từ chức năng đọc danh sách:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màn hình

Mô tả được tạo tự động

# Danh mục tham khảo

– C++ md5 function, zedwood.com, <http://www.zedwood.com/article/cpp-md5-function>

– GitHub Reponsitory ‘AES’, Tác giả ‘SergeyBel’, <https://github.com/SergeyBel/AES>