

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



Thành viên nhóm

| | |
|----------|------------------|
| 20120344 | VƯƠNG TẤN PHÁT |
| 20120356 | LÊ MINH QUÂN |
| 20120369 | NGUYỄN THANH TÂN |
| 20120386 | LÊ PHƯỚC TOÀN |

ĐỒ ÁN CUỐI KỲ

A COURSE REGISTRATION SYSTEM

Giáo viên hướng dẫn: ThS Nguyễn Lê Hoàng Dũng
Học phần: Kỹ thuật lập trình

Thành phố Hồ Chí Minh – 8/2021

LỜI CẢM ƠN

Báo cáo đồ án cuối kỳ học phần Kỹ thuật lập trình với đề tài “A Course Registration System” là kết quả của quá trình cố gắng không ngừng của cả nhóm và được sự giúp đỡ, động viên khích lệ của các thầy, bạn bè và người thân. Qua trang viết này nhóm tác giả xin gửi lời cảm ơn tới những người đã giúp đỡ cả nhóm trong thời gian học tập - nghiên cứu khoa học vừa qua.

Cả nhóm xin tỏ lòng kính trọng và biết ơn sâu sắc đối với thầy Nguyễn Lê Hoàng Dũng đã trực tiếp tận tình hướng dẫn cũng như cung cấp tài liệu thông tin khoa học cần thiết cho đồ án này.

Xin chân thành cảm ơn Lãnh đạo trường Đại học Khoa học Tự nhiên, khoa Công nghệ thông tin đã tạo điều kiện cho nhóm hoàn thành tốt công việc nghiên cứu khoa học của nhóm.

Cuối cùng, xin gửi lời cảm ơn bạn bè đã giúp đỡ nhóm trong quá trình học tập và thực hiện Báo cáo đồ án.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

| | |
|---|----|
| LỜI CẢM ƠN | 2 |
| MỤC LỤC..... | 3 |
| 1. MỞ ĐẦU | 4 |
| 1.1 Giới thiệu đồ án: | 4 |
| 1.2 Yêu cầu đồ án: | 4 |
| 2. BÁO CÁO KỸ THUẬT VỀ HỆ THỐNG ĐĂNG KÝ HỌC PHẦN..... | 5 |
| 2.1. Báo cáo về cấu trúc lưu trữ dữ liệu: | 5 |
| 2.2. Báo cáo về cấu trúc dữ liệu: | 5 |
| 2.3. Báo cáo những hàm xử lý chính trong các tập tin mã nguồn:..... | 7 |
| 2.4. Báo cáo những kỹ thuật đã sử dụng trong các hàm: | 10 |
| 2.4.1. Cấu trúc dữ liệu:..... | 10 |
| 2.4.2. Đọc/ghi/xóa/đổi tên tập tin: | 10 |
| 2.4.3. Cấu trúc rẽ nhánh – cấu trúc lặp: | 11 |
| 2.4.4. Con trỏ và kỹ thuật cấp phát bộ nhớ động:..... | 11 |
| 2.4.5. Kỹ thuật sử dụng danh sách liên kết đơn:..... | 11 |
| 2.4.6. Kỹ thuật xử lý trên chuỗi ký tự:..... | 11 |
| 2.4.7. Kỹ thuật mã hóa: | 11 |
| 3. BÁO CÁO TỔNG KẾT ĐỒ ÁN | 13 |
| 3.1. Đánh giá công việc: | 13 |
| 3.2. Kết luận: | 13 |
| TÀI LIỆU THAM KHẢO..... | 14 |

1. MỞ ĐẦU

1.1 Giới thiệu đề án:

- Hệ thống đăng ký học phần là một chương trình để cho giáo vụ thao tác với các thông tin học vụ và để sinh viên thực hiện việc đăng ký học phần.

1.2 Yêu cầu đề án:

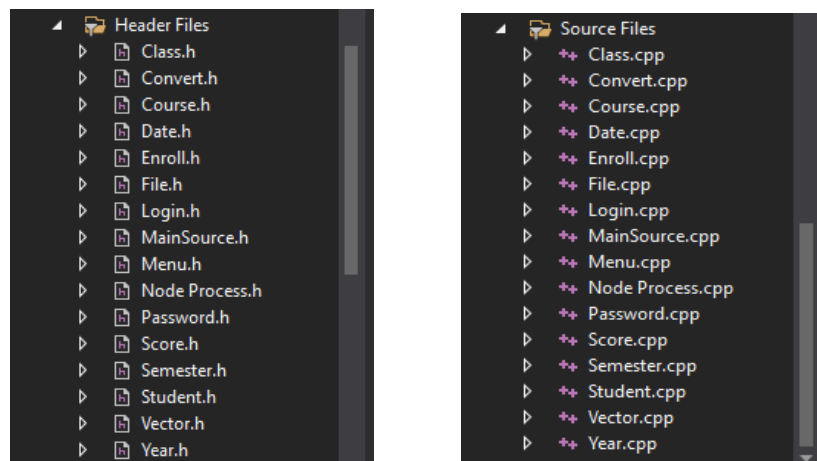
- Hệ thống có 2 người dùng: giảng viên và sinh viên.
- Có chức năng đăng nhập.
- Khi đó, người đăng nhập có thể xem thông tin đăng ký học phần của họ, thay đổi mật khẩu và đăng xuất khỏi chương trình.
- Được dùng tập tin văn bản, tập tin nhị phân, không được dùng cơ sở dữ liệu.
- Viết bằng ngôn ngữ C/C++.
- Cho phép dùng mảng động, danh sách liên kết và chuỗi kí tự.
- Dùng *github* để quản lý dự án và phiên bản code.

2. BÁO CÁO KỸ THUẬT VỀ HỆ THỐNG ĐĂNG KÝ HỌC PHẦN

2.1. Báo cáo về cấu trúc lưu trữ dữ liệu:

- Dữ liệu sẽ được lưu trữ trên tệp tin csv.
- Dữ liệu thông tin đăng nhập của giảng được lưu ở Accounts\acc_ad.csv và dữ liệu thông tin đăng nhập của sinh viên được lưu ở Accounts\acc_sv.csv bao gồm tên tài khoản, mật khẩu và đường dẫn tới lớp của đối tượng.
- Dữ liệu thông tin sinh viên được lưu ở Student\Students' Info bao gồm số thứ tự, mã số sinh viên, họ và tên, giới tính, khoa, ngày sinh, tài khoản, mật khẩu của sinh viên. Dữ liệu này được lưu trong một tập tin lớp cụ thể trong một năm nào đó. Ví dụ “CTT 2020”.
- Dữ liệu thông tin lớp học được lưu ở thư mục Classes\ bao gồm các thư mục con chứa thông tin từng lớp học theo từng niên khóa. Tập tin lớp học lưu thông tin của học sinh.
- Dữ liệu thông tin năm học được lưu ở Years\ với các thư mục con là file csv, chứa thông tin từng lớp học được mở ứng với năm của thư mục con. Trong tập tin của các năm là các tên lớp đã tạo.

2.2. Báo cáo về cấu trúc dữ liệu:



Hình 1.2.a và 1.2.b các file.h và file.cpp khai báo và chứa các hàm sử dụng

- Chương trình được chia thành các file .h để khai báo các hàm sử dụng theo từng chức năng và kèm theo đó là file .cpp để viết các mã nguồn tương ứng nhằm để chia nhỏ chương trình, dễ dàng trong việc quản lý và sửa lỗi.

+ *MainSource.cpp* chứa mã nguồn của hàm main – hàm giúp khởi chạy chương trình.

+ *MainSource.h* để chứa các khai báo toàn cục (ví dụ các struct, các thư viện của C/C++,...).

+ *Menu.h* và *Menu.cpp* chứa các khai báo và mã nguồn các hàm in các Menu và các tiêu đề ở mỗi chức năng.

+ *Node Process.h* và *Node Process.cpp* chứa các khai báo và mã nguồn hàm thao tác với danh sách liên kết.

+ *File.h* và *File.cpp* chứa các khai báo và mã nguồn hàm thao tác với tập tin dữ liệu. Ngoài ra có thêm hai hàm tạo và xóa một thư mục (Directory Create & Directory Delete).

+ *Convert.h* và *Convert.cpp* chứa các khai báo và mã nguồn hàm dùng để thay đổi kiểu dữ liệu cho phù hợp với mục đích sử dụng. Thêm vào đó còn có hàm thay đổi một ký tự cho trước trong chuỗi thành một ký tự bất kỳ (*String_Replace*).

+ *Login.h* và *Login.cpp* chứa các khai báo và mã nguồn hàm quản lý việc đăng nhập của người dùng.

+ *Password.h* và *Password.cpp* chứa các khai báo và mã nguồn hàm kiểm tra mật khẩu đăng nhập có trong dữ liệu và thay đổi mật khẩu.

+ *Enroll.h* và *Enroll.cpp* chứa các khai báo và mã nguồn hàm đăng ký học phần của sinh viên.

+ *Date.h* và *Date.cpp* chứa các khai báo và mã nguồn hàm quản lý thời gian thích hợp.

+ *Class.h* và *Class.cpp* chứa các khai báo và mã nguồn hàm quản lý và thao tác với cấu trúc dữ liệu lớp.

+ *Course.h* và *Course.cpp* chứa các khai báo và mã nguồn hàm quản lý và thao tác với dữ liệu môn học.

+ *Score.h* và *Score.cpp* chứa các khai báo và mã nguồn hàm quản lý và thao tác dữ liệu điểm số.

+ *Semester.h* và *Semester.cpp* chứa các khai báo và mã nguồn hàm quản lý và thao tác dữ liệu học kỳ.

+ *Student.h* và *Student.cpp* chứa các khai báo và mã nguồn hàm quản lý và thao tác dữ liệu học sinh.

+ *Vector.h* và *Vector.cpp* chứa các khai báo và mã nguồn hàm quản lý và thao tác với vector. Ngoài ra còn có hàm lấy dữ liệu từ tập tin và cho vào vector chuỗi cũng như chuyển dữ liệu từ vector chuỗi vào tập tin.

+ *Year.h* và *Year.cpp* chứa các khai báo và mã nguồn hàm quản lý và thao tác dữ liệu năm học.

- Khi viết mã nguồn, nếu sử dụng lại hàm ở file khác đã viết thì *#include* thư viện vào file .cpp. Việc làm trên giúp tận dụng các hàm đã có sẵn; đồng thời để tránh trường hợp khi chạy, chương trình gọi thư viện lặp đi lặp lại nhiều lần (nếu để khai báo thư viện ở file .h).

2.3. Báo cáo những hàm xử lý chính trong các tập tin mã nguồn:

Convert.cpp

string Extension(*string* name, *int* option); chuyển một tên tập tin thành có đuôi hoặc cắt đuôi định dạng.

+Tham số: chuỗi cần chuyển, option: 1 thêm đuôi, khác một cắt đuôi.

+Trả về: chuỗi sau khi thêm/cắt.

string Make_Path(*string* pre_folder, *string* file_name); chuyển một tên tập tin thành đường dẫn cụ thể

+Tham số: thư mục chứa tập tin, tên tập tin.

+Trả về: đường dẫn sau khi tạo thành công.

Date.cpp

- Bao gồm các hàm nhập và xử lý ngày tháng:

- Nhập ngày tháng và kiểm tra tính đúng đắn của nó (ví dụ ngày nhập vào là 32 sẽ không hợp lệ).

- Nhập vào khoảng ngày tháng khi tạo một tập tin học kỳ và kiểm tra khoảng này có trùng với các khoảng trước hay không.
- Nhận diện thời gian nhập vào có thuộc học kỳ hay giai đoạn đăng ký học phần của học kỳ nào hay không.

Menu.cpp

int Valid_Data(*int* limit); là hàm có chức năng nhập dữ liệu đầu vào dưới dạng số nguyên nào đó và kiểm tra tính đúng đắn của nó.

- + Tham số: số thứ tự tác vụ tối đa có thể có.
- + Trả về: số thứ tự của tác vụ đã chọn.

bool Display_Mode_Admin(*Account* info, *date* &dmy); nhận diện thời gian nhập vào thuộc khoảng thời gian thao tác nào (Trong hoặc ngoài một học kỳ nào đó). Sau đó sẽ điều hướng đến hai loại menu xử lý: mod 0 chỉ cho phép các tác vụ xem thông tin (đối với dữ liệu năm thì có thể thêm, xóa) và mod 1 cho phép thực hiện tất cả các tác vụ.

- + Tham số: Thông tin tài khoản, thời gian hiện hành.
- + Trả về: *True* nếu cần tái sử dụng hàm (hiển thị dạng vòng lặp), *False* nếu không cần sử dụng.

bool Display_Mode_Student(*Student* &info, *date* &dmy); cũng tương tự sẽ nhận diện thời gian đăng ký học phần, nếu ngoài thời gian này, sinh viên chỉ có thể xem các môn học đã đăng ký mà không thể thêm hoặc xóa.

- + Tham số: Thông tin học sinh, thời gian hiện hành.
- + Trả về: *True* nếu cần tái sử dụng hàm.

bool Admin_Proc(*int* option, *Account* info, *date* dmy, *int* mod); xử lý các tác vụ dành cho giáo vụ.

- + Tham số: số thứ tự tác vụ đã chọn, thông tin tài khoản, ngày hiện hành và chế độ thao tác.
- + Trả về: *True* nếu cần tái sử dụng hàm.

Student.cpp

void Student_Import(*string* class_path); dùng để nhập một số lượng lớn các học sinh từ tập tin có sẵn.

+ Tham số: đường dẫn đến lớp học.

void Student_Arrange(*string* class_path) để sắp xếp lại số thứ tự học sinh trong tập tin.

+ Tham số: đường dẫn đến lớp học.

void Student_inCourse(*string* semester_path, *string* faculty) để hiển thị danh sách sinh viên có trong tập tin môn học.

+ Tham số: đường dẫn đến học kỳ, tên của khoa đã chọn.

void Student_Course_Update(*string* new_course_path, *string* new_string) để cập nhật thông tin môn học có trong tập tin sinh viên sau khi môn học được sửa đổi.

+ Tham số: đường dẫn đến môn học mới, chuỗi thông tin môn học mới.

bool Student_Proc_Active(*int* option, *string* class_path, *int* mod) để xử lý các tính năng mà giáo vụ quản lý sinh viên (thêm, hiển thị sinh viên).

+ Tham số: số thứ tự tác vụ đã chọn, đường dẫn lớp học, chế độ thao tác

+ Trả về: *True* nếu cần tái sử dụng hàm.

bool Student_Proc_Passive(*int* option, *Student* info, *date* dmy, *int* mod) để xử lý các tính năng mà sinh viên có thể tự quản lý bản thân (Đăng ký môn học, xem môn học, xóa môn học).

+ Tham số: số thứ tự tác vụ đã chọn, thông tin sinh viên, ngày hiện hành, chế độ thao tác.

+ Trả về: *True* nếu cần tái sử dụng hàm.

Course.cpp

Course Course_Input(); để nhập thông tin môn học và kiểm tra tính đúng đắn (có bị trùng lặp không, số tín chỉ có lớn hơn 0 không,...)

+ Trả về: dữ liệu môn học.

int Course_Find_in_Student(*string* student_path, *string* course_name); để tìm số dòng mà môn học được lưu trong tập tin học sinh để thao tác và xử lý.

- + Tham số: đường dẫn đến file học sinh, tên môn học
- + Trả về: số dòng của môn học trong file học sinh.

void Course_Update(*string* semester_path, *string* faculty); cập nhật môn học.

- + Tham số: đường dẫn học kỳ, tên khoa đã chọn.

Enroll.cpp

bool Conflicted_Course (*string* store, *string* name); để kiểm tra môn học có bị xung đột với môn đã đăng ký của sinh viên hay không.

- + Tham số: file lưu các môn học của học sinh, tên môn học cần kiểm tra.
- + Trả về: True nếu cần tái sử dụng.

void View_Enrolled_Courses (*Student* sv); để xuất ra các môn học đã đăng ký của sinh viên.

- + Tham số: Thông tin kiểu sinh viên.

Login.cpp

check_acc_student và *check_acc_ad* để kiểm tra các tài khoản nhập vào có hợp lệ hay không.

login_as_student và *login_as_admin* để thực hiện việc đăng nhập dưới vai trò sinh viên và giáo vụ.

2.4. Báo cáo những kỹ thuật đã sử dụng trong các hàm:

2.4.1. Cấu trúc dữ liệu:

- Chương trình đã sử dụng dữ liệu cấu trúc để lưu trữ các thông tin của sinh viên, thông tin đăng nhập, điểm số, lớp học, định dạng ngày, các cấu trúc của danh sách liên kết.
- Ngoài ra, chương trình cũng sử dụng các cấu trúc để làm kiểu dữ liệu trả về.
- Điều này giúp cho việc khai báo và truy xuất các thành phần của đối tượng (sinh viên, điểm, ngày,...) dễ dàng hơn.

2.4.2. Đọc/ghi/xóa/đổi tên tập tin:

- Chương trình được thao tác thông qua thư viện `<fstream.h>` với hàm khai báo `ifstream` và `ofstream`.
- Xóa tập tin dùng hàm `[remove(const char *)]`.
- Đổi tên tập tin dùng hàm `[rename(const char*, const char*)]`.

2.4.3. Cấu trúc rẽ nhánh – cấu trúc lặp:

- Sử dụng cấu trúc for, while.
- Cấu trúc if, switch...case để tạo các Menu.

2.4.4. Con trỏ và kỹ thuật cấp phát bộ nhớ động:

- Chương trình đã sử dụng các biến con trỏ để trỏ đến các vùng nhớ trong bộ nhớ để lấy dữ liệu và thao tác.
- Chương trình có những câu lệnh kiểm tra xem con trỏ có được cấp phát bộ nhớ thành công hay chưa, nếu chưa sẽ thoát hàm.
- Kỹ thuật cấp phát bộ nhớ động để quản lý tốt hơn bộ nhớ, tránh lãng phí bộ nhớ nếu khai báo bằng các biến tĩnh thông qua toán tử *new*.
- Sau khi cấp phát bộ nhớ, chương trình đã giải phóng bộ nhớ cấp phát đó khi không còn sử dụng.
- Chương trình đã vận dụng con trỏ và viết các hàm có kiểu dữ liệu trả về là con trỏ.
- Chương trình đã sử dụng kỹ thuật cấu trúc hóa vùng nhớ thô để kết hợp giữa dữ liệu cấu trúc và con trỏ.
- Toán tử của biến con trỏ có cấu trúc là ->.

2.4.5. Kỹ thuật sử dụng danh sách liên kết đơn:

- Chương trình đã sử dụng danh sách liên kết để liên kết các ô nhớ rời rạc trong bộ nhớ để tận dụng tối đa các ô nhớ đang trống trên bộ nhớ thay vì dùng mảng (các ô nhớ phải liên tục).
- Chương trình đã khởi tạo được danh sách liên kết, thêm vào danh sách liên kết và giải phóng danh sách liên kết.

2.4.6. Kỹ thuật xử lý trên chuỗi ký tự:

- Sử dụng thư viện `<string.h>` để sử dụng các hàm liên quan đến mảng ký tự *char*.
- Áp dụng được các thao tác xử lý trên chuỗi: sao chép chuỗi, so sánh chuỗi, ghép chuỗi này vào một chuỗi khác,...

2.4.7. Kỹ thuật mã hóa:

- Áp dụng cho việc nhập mật khẩu và ẩn các ký tự đi.
- Sử dụng hàm *kbhit()* để nhận diện rằng người dùng có nhấn phím nào trên bàn phím hay không?

- Kiểm tra xem người dùng có nhấn phím *Backspace* hay không thông qua mã ASCII:

+ Nếu không sẽ in ra màn hình dấu *.

+ Nếu có sẽ không in ra dấu *Backspace*.

- Nếu người dùng bấm phím *Back* (nút xóa phía trên bàn phím) thì sẽ được xóa đi kí tự *.

- Chuỗi kí tự * sau khi nhập xong sẽ được cập nhật thành chuỗi các kí tự được nhập từ bàn phím.

3. BÁO CÁO TỔNG KẾT ĐỒ ÁN

3.1. Đánh giá công việc: Có trong file excel đính kèm và video tại link:
<https://youtu.be/qt4Qc-8Iktk>

3.2. Kết luận:

- Chương trình hoạt động tốt trên nền console.
- Các chức năng hoạt động đúng.
- Các hàm liên kết với nhau một cách chặt chẽ, chương trình được thiết kế dễ nhìn, dễ dàng sử dụng.
- Khai báo thư viện, mã nguồn hàm rõ ràng, không bị chồng chất thư viện.
- Tuy nhiên, vẫn còn một số lỗi cần khắc phục: xử lý chuỗi còn cồng kềnh, dữ liệu tổ chức còn rời rạc, dư thừa, sử dụng *github* chưa đúng cách.

TÀI LIỆU THAM KHẢO

- ThS Nguyễn Lê Hoàng Dũng:
Tài liệu “*Final project – A Course Registration System*” – Lưu hành nội bộ.
- PGS, TS Trần Đan Thư, TS Nguyễn Thanh Phương, TS Đinh Bá Tiên, PGS, TS Trần Minh Triết, ThS Đặng Bình Phương: “*Kỹ thuật lập trình*”.
- PGS, TS Trần Đan Thư, TS Nguyễn Thanh Phương, TS Đinh Bá Tiên, PGS, TS Trần Minh Triết: “*Nhập môn lập trình*”.
- *Github*: <https://github.com/>
- *StackOverflow*: <https://stackoverflow.com/>
- *Cplusplus*: <http://www.cplusplus.com/>