

# **TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



## **ĐỒ ÁN HỌC PHẦN**

### **PHẦN MỀM MÔ PHỎNG NÉN ẢNH JPEG THEO MÃ HUFFMAN**

**BÙI ĐỨC CÔNG VINH**

[vinh.bdc210969@sis.hust.edu.vn](mailto:vinh.bdc210969@sis.hust.edu.vn)

**DƯƠNG QUANG TÚ**

[duy.nk224407@sis.hust.edu.vn](mailto:duy.nk224407@sis.hust.edu.vn)

**NGÔ KHÁNH DUY**

[tu.dq224427@sis.hust.edu.vn](mailto:tu.dq224427@sis.hust.edu.vn)

**Chuyên ngành Hệ thống nhúng thông minh & Internet of Things  
Học phần Lý thuyết thông tin**

**Giảng viên hướng dẫn:** PGS. TS. Nguyễn Hữu phát

**Bộ môn:** Lý thuyết thông tin

**Mã lớp:** 1494314

**Viện:** Điện tử viễn thông

**HÀ NỘI, 6/2024**

# MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ ĐỀ TÀI.....</b>	<b>3</b>
<b>1.1 Đặt vấn đề .....</b>	<b>3</b>
<b>1.2. Mục tiêu và phạm vi đề tài.....</b>	<b>3</b>
<b>1.2.1 Mục tiêu.....</b>	<b>3</b>
<b>1.2.2 Phạm vi đề tài .....</b>	<b>3</b>
<b>1.3. Định hướng giải pháp .....</b>	<b>4</b>
<b>CHƯƠNG 2. NỘI DUNG ĐỀ TÀI.....</b>	<b>6</b>
<b>1. Giới thiệu mã Huffman.....</b>	<b>6</b>
<b>3. Ứng dụng mã Huffman trong nén ảnh JPEG.....</b>	<b>5</b>
<b>4. Encoder .....</b>	<b>6</b>
<b>4.1 Các thư viện.....</b>	<b>6</b>
<b>4.2 Định nghĩa hàm trợ giúp.....</b>	<b>9</b>
<b>4.3 Đọc ảnh BMP .....</b>	<b>8</b>
<b>4.4 Chuyển đổi phổ màu từ RGB sang YCbCr.....</b>	<b>10</b>
<b>4.5 DCT .....</b>	<b>11</b>
<b>4.6 Biến đổi DCT trên toàn bộ ảnh .....</b>	<b>12</b>
<b>4.7 Lượng tử hóa.....</b>	<b>13</b>
<b>4.8 Lớp BitWriter: .....</b>	<b>14</b>
<b>4.9 Tạo mã Huffman.....</b>	<b>15</b>
<b>4.10 Mã hóa Huffman.....</b>	<b>16</b>
<b>4.11 Mã hóa JPEG .....</b>	<b>18</b>
<b>5. Thư viện “JPG.h”.....</b>	<b>23</b>
<b>5.1 Định nghĩa các hằng số, kiểu dữ liệu và cấu trúc dữ liệu cơ bản. 23</b>	
<b>5.2 Khai báo các bảng lượng tử hoá với các giá trị đã được chuẩn hóa.</b>	
<b>.....</b>	<b>24</b>

5.3 Khai báo các bảng Huffman cho các thành phần YCbCr trong quá trình mã hóa ảnh JPEG.....	26
<b>CHƯƠNG 3. TỔNG KẾT.....</b>	<b>14</b>
1. Kết luận.....	14
2. Hướng phát triển tương lai .....	14
3. Lời cảm ơn .....	14
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>16</b>

# CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong thời đại công nghệ số, việc lưu trữ và truyền tải hình ảnh đã trở thành một nhu cầu thiết yếu trong nhiều lĩnh vực như truyền thông, y tế, giáo dục, và giải trí. Các bức ảnh kỹ thuật số thường có dung lượng lớn, gây khó khăn trong việc lưu trữ và truyền tải. Do đó, việc nén ảnh để giảm dung lượng mà không làm mất nhiều chất lượng là rất quan trọng.

JPEG (Joint Photographic Experts Group) [1] là một trong những chuẩn nén ảnh phổ biến nhất hiện nay. Kỹ thuật nén JPEG sử dụng nhiều phương pháp để giảm kích thước tệp, trong đó mã hóa Huffman đóng vai trò quan trọng trong việc nén dữ liệu. Mã hóa Huffman là một thuật toán nén không mất mát, giúp giảm thiểu kích thước dữ liệu bằng cách sử dụng các mã ngắn hơn cho các ký tự xuất hiện thường xuyên và các mã dài hơn cho các ký tự ít xuất hiện hơn.

Để hiểu rõ hơn về quá trình nén ảnh JPEG và vai trò của mã hóa Huffman, đề tài "Chương trình C++ mô phỏng mã hóa JPEG dùng mã Huffman" đã được chọn. Thông qua việc xây dựng chương trình mô phỏng, chúng ta có thể nghiên cứu và áp dụng các kỹ thuật nén, đồng thời đánh giá hiệu quả của chúng.

## 1.2. Mục tiêu và phạm vi đề tài

### 1.2.1 Mục tiêu

**Nghiên cứu lý thuyết:** Tìm hiểu chi tiết về thuật toán nén JPEG và mã hóa Huffman, từ đó hiểu rõ cách thức hoạt động và ứng dụng của chúng.

**Phát triển chương trình mô phỏng:** Xây dựng một chương trình bằng ngôn ngữ C++ có khả năng thực hiện quá trình mã hóa và giải mã ảnh JPEG sử dụng mã Huffman.

**Đánh giá hiệu quả:** So sánh kết quả nén của chương trình mô phỏng với các phương pháp nén khác, đánh giá hiệu quả nén và tốc độ xử lý.

### 1.2.2 Phạm vi đề tài

**Phạm vi lý thuyết:** Đề tài tập trung vào thuật toán nén ảnh JPEG và mã hóa Huffman, không mở rộng sang các phương pháp nén khác như nén ảnh PNG, GIF hoặc các thuật toán nén âm thanh, video.

**Phạm vi thực nghiệm:** Chương trình mô phỏng sẽ được viết bằng ngôn ngữ lập trình C++, với các thử nghiệm nén và giải nén thực hiện trên một số tập hợp ảnh mẫu có định dạng JPEG.

**Giới hạn ứng dụng:** Chương trình mô phỏng chủ yếu mang tính chất học thuật và nghiên cứu, không hướng tới việc triển khai trong các hệ thống nén ảnh thương mại.

### 1.3. Định hướng giải pháp

Để đạt được các mục tiêu trên, quá trình thực hiện đề tài sẽ được định hướng theo các bước sau:

**Nghiên cứu tài liệu:** Tìm hiểu các tài liệu khoa học, sách giáo khoa, và các nguồn tài nguyên trực tuyến liên quan đến thuật toán nén JPEG và mã hóa Huffman. Điều này giúp xây dựng nền tảng lý thuyết vững chắc cho việc phát triển chương trình mô phỏng.

**Phân tích yêu cầu:** Xác định các yêu cầu cụ thể cho chương trình mô phỏng, bao gồm các chức năng chính như mã hóa, giải mã, và giao diện người dùng cơ bản.

**Thiết kế thuật toán:** Dựa trên các yêu cầu đã phân tích, thiết kế thuật toán mã hóa và giải mã JPEG sử dụng mã Huffman. Thuật toán cần đảm bảo hiệu quả về mặt thời gian và bộ nhớ.

**Triển khai chương trình:** Lập trình và kiểm thử chương trình bằng ngôn ngữ C++. Quá trình triển khai cần tuân theo các nguyên tắc lập trình tốt, đảm bảo mã nguồn rõ ràng, dễ hiểu và bảo trì.

**Kiểm thử và đánh giá:** Chạy thử chương trình với các bộ dữ liệu ảnh khác nhau, so sánh kết quả nén với các phương pháp nén khác. Đánh giá hiệu quả nén về mặt tỷ lệ nén và chất lượng ảnh sau khi giải nén.

**Báo cáo kết quả:** Tổng hợp các kết quả thu được, viết báo cáo và trình bày các kết quả đạt được, những khó khăn gặp phải và hướng phát triển trong tương lai.

Trên đây là phần giới thiệu chung về đề tài "Chương trình C++ mô phỏng mã hóa JPEG dùng mã Huffman". Chương trình này không chỉ giúp chúng ta hiểu rõ hơn về các kỹ thuật nén ảnh mà còn là cơ hội để áp dụng các kiến thức về lập trình, giải thuật và tối ưu hóa trong thực tế.

#### 1.4. Bố cục của báo cáo

Phần còn lại của báo cáo project được trình bày như sau:

- **Chương 2:** Các công nghệ sử dụng - Giới thiệu các công nghệ và công cụ được sử dụng trong quá trình phát triển chương trình, bao gồm ngôn ngữ lập trình C++, các thư viện hỗ trợ, và các công cụ kiểm thử.
- **Chương 3:** Quá trình thiết kế - Mô tả chi tiết quá trình thiết kế chương trình, bao gồm thiết kế thuật toán mã hóa và giải mã, cấu trúc dữ liệu, và giao diện người dùng. Chương này cũng trình bày các sơ đồ thiết kế, luồng dữ liệu và các bước triển khai thuật toán.
- **Chương 4:** Kết quả và thảo luận - Trình bày kết quả thu được từ chương trình mô phỏng, so sánh với các phương pháp nén khác và đánh giá hiệu quả. Chương này cũng thảo luận về các khó khăn gặp phải và các giải pháp khắc phục.
- **Chương 5:** Kết luận và hướng phát triển - Tóm tắt các kết quả đạt được, đưa ra kết luận về hiệu quả của chương trình mô phỏng và đề xuất các hướng phát triển trong tương lai.

## CHƯƠNG 2. CÁC CÔNG NGHỆ SỬ DỤNG

### 2.1. Ngôn ngữ lập trình C++

#### 2.1.1. Giới thiệu về C++

C++ là một ngôn ngữ lập trình bậc trung, hỗ trợ cả lập trình hướng đối tượng và lập trình thủ tục. Được phát triển bởi Bjarne Stroustrup vào năm 1979 tại Bell Labs, C++ là một ngôn ngữ mạnh mẽ và linh hoạt, thường được sử dụng trong các ứng dụng yêu cầu hiệu suất cao như hệ điều hành, trình biên dịch, game, và các ứng dụng thời gian thực.

#### 2.1.2. Ưu điểm của C++ trong project

- **Hiệu suất cao:** C++ cho phép lập trình viên kiểm soát chi tiết việc quản lý bộ nhớ, giúp tối ưu hóa hiệu suất của chương trình.
- **Thư viện phong phú:** C++ có một hệ thống thư viện chuẩn (Standard Template Library - STL) mạnh mẽ, cung cấp các cấu trúc dữ liệu và thuật toán hiệu quả.
- **Hỗ trợ hướng đối tượng:** C++ hỗ trợ lập trình hướng đối tượng, giúp quản lý và tổ chức mã nguồn một cách dễ dàng và hiệu quả.
- **Tính linh hoạt:** C++ có thể sử dụng trong nhiều loại ứng dụng khác nhau, từ hệ thống nhúng đến các ứng dụng desktop và server.

### 2.2. Các thư viện C++ sử dụng

```
1 #include <iostream>
2 #include <fstream>
3 #include <vector>
4
5 #include "jpg.h"
```

- **#include <iostream>:** Thư viện để sử dụng các đối tượng đầu vào và đầu ra như `std::cout` để in thông báo lỗi hoặc thông tin.
- **#include <fstream>:** Thư viện để làm việc với tệp, sử dụng `std::ifstream` để đọc tệp và `std::ofstream` để ghi tệp.
- **#include <vector>:** Thư viện chứa lớp `std::vector`, một container động giúp quản lý một mảng động các phần tử.

- **#include "jpg.h"**: Bao gồm tệp tiêu đề tùy chỉnh "jpg.h", chứa các khai báo và định nghĩa cần thiết cho quá trình chuyển đổi và nén ảnh.

## 2.3. Thư viện Standard Template Library (STL)

### 2.3.1. Giới thiệu về STL

Standard Template Library (STL) là một phần của thư viện chuẩn C++, cung cấp các cấu trúc dữ liệu và thuật toán tổng quát. STL bao gồm ba thành phần chính: container (cấu trúc dữ liệu chứa), iterator (bộ lặp), và algorithm (thuật toán).

### 2.3.2. Ứng dụng của STL trong dự án

- **Container**: Các container như `vector`, `list`, và `map` sẽ được sử dụng để lưu trữ và quản lý dữ liệu trong quá trình nén và giải nén ảnh.
- **Iterator**: Iterator giúp duyệt qua các phần tử trong container một cách dễ dàng và hiệu quả.
- **Algorithm**: Các thuật toán chuẩn như `sort` và `binary_search` sẽ được sử dụng để thực hiện các thao tác trên dữ liệu một cách tối ưu.

## 2.5. Công cụ quản lý mã nguồn

### 2.5.1. Git và GitHub

Git là một hệ thống quản lý phiên bản phân tán, giúp theo dõi sự thay đổi của mã nguồn trong suốt quá trình phát triển. GitHub là một dịch vụ web dựa trên Git, cung cấp các công cụ quản lý dự án và hợp tác làm việc.

### 2.5.2. Ứng dụng của Git và GitHub trong dự án

- **Quản lý phiên bản**: Sử dụng Git để theo dõi và quản lý các phiên bản khác nhau của mã nguồn, giúp dễ dàng quay lại các phiên bản trước khi cần thiết.
- **Hợp tác làm việc**: Sử dụng GitHub để lưu trữ mã nguồn trực tuyến, quản lý các yêu cầu thay đổi (pull request), và theo dõi các vấn đề (issue) trong dự án.
- **Sao lưu và phục hồi**: GitHub cung cấp một nơi lưu trữ an toàn cho mã nguồn, giúp sao lưu và phục hồi mã nguồn dễ dàng khi cần thiết.

## 2.6. Công cụ biên dịch và phát triển

### 2.6.1. Visual Studio

Visual Studio là một môi trường phát triển tích hợp (IDE) mạnh mẽ của Microsoft, hỗ trợ nhiều ngôn ngữ lập trình, bao gồm C++. Visual Studio cung cấp các công cụ biên dịch, gỡ lỗi, và kiểm thử hiệu quả.



### 2.6.2. Ứng dụng của Visual Studio trong dự án

- **Phát triển mã nguồn:** Sử dụng Visual Studio để viết, biên dịch và gỡ lỗi mã nguồn C++.
- **Quản lý dự án:** Sử dụng các tính năng quản lý dự án của Visual Studio để tổ chức và theo dõi tiến độ phát triển.
- **Kiểm thử và gỡ lỗi:** Sử dụng các công cụ kiểm thử và gỡ lỗi của Visual Studio để phát hiện và sửa lỗi trong mã nguồn.

Như vậy, các công nghệ và công cụ trên sẽ hỗ trợ đắc lực cho việc phát triển chương trình mô phỏng mã hóa JPEG dùng mã Huffman, đảm bảo hiệu quả và chất lượng của sản phẩm cuối cùng.

## CHƯƠNG 3: QUÁ TRÌNH THIẾT KẾ

### 3.1. Giới thiệu tổng quan về quá trình thiết kế

Quá trình thiết kế của chương trình mô phỏng mã hóa JPEG bằng mã Huffman được thực hiện qua các bước: phân tích yêu cầu, thiết kế kiến trúc hệ thống, triển khai mã nguồn, và kiểm thử. Các bước này nhằm đảm bảo rằng chương trình đáp ứng đầy đủ yêu cầu về chức năng, hiệu suất, và độ tin cậy.

### 3.2. Phân tích yêu cầu

Trong giai đoạn phân tích yêu cầu, chúng tôi xác định các chức năng chính mà chương trình cần thực hiện, bao gồm:

1. Đọc dữ liệu ảnh từ file BMP.
2. Chuyển đổi dữ liệu ảnh thành định dạng phù hợp để nén.
3. Nén dữ liệu ảnh bằng thuật toán mã Huffman.
4. Lưu dữ liệu ảnh đã nén vào file JPEG.
5. Giải nén dữ liệu ảnh JPEG để kiểm tra tính đúng đắn của quá trình nén.

### 3.3. Thiết kế kiến trúc hệ thống

Kiến trúc hệ thống được thiết kế theo mô hình module để dễ dàng quản lý và mở rộng. Các module chính bao gồm:

1. **Module đọc và ghi file:** Đọc dữ liệu từ file BMP và ghi dữ liệu ra file JPEG.
2. **Module xử lý ảnh:** Chuyển đổi dữ liệu ảnh và thực hiện các bước tiền xử lý.
3. **Module nén Huffman:** Thực hiện nén và giải nén dữ liệu ảnh bằng thuật toán mã Huffman.
4. **Module kiểm thử:** Kiểm tra tính đúng đắn của các module khác.

### 3.4. Triển khai mã nguồn

#### 3.4.1. Cấu trúc thư mục

Cấu trúc thư mục của dự án được tổ chức như sau:

- **src/:** Chứa mã nguồn của các module.
- **bin/:** Chứa các file thực thi sau khi biên dịch.
- **Makefile:** Tập tin điều khiển quá trình biên dịch.
- **README.md:** Tập tin hướng dẫn và thông tin về dự án.
- **sample.bmp:** File ảnh mẫu dùng để thử nghiệm.

#### 3.4.2. Mã nguồn chính

Dưới đây là một số đoạn mã nguồn chính trong dự án

File 'encoder.cpp':

```
#include <iostream>
#include <fstream>
#include <vector>

//readBMP
//BMPImage
//....
//compress functions and write output JPG file
#include "jpg.h"int main(int argc, char** argv) {
    // validate arguments
    if (argc < 2) {
        std::cout << "Error - Invalid arguments\n";
        return 1;
    }

    for (int i = 1; i < argc; ++i) {
        const std::string filename(argv[i]);

        // read image
        BMPImage image = readBMP(filename);
        // validate image
        if (image.blocks == nullptr) {
            continue;
        }

        // color conversion
        RGBToYCbCr(image);

        // Forward Discrete Cosine Transform
        forwardDCT(image);

        // quantize DCT coefficients
        quantize(image);

        // write JPG file
        const std::size_t pos = filename.find_last_of('.');
        const std::string outFilename = (pos == std::string::npos) ?
            (filename + ".jpg") :
            (filename.substr(0, pos) + ".jpg");
        writeJPG(image, outFilename);

        delete[] image.blocks;
    }
    return 0;
}
```

File 'jpg.h':

```
#ifndef JPG_H
#define JPG_H

#include <cmath>

typedef unsigned char byte;
typedef unsigned int uint;

//lists of JPG markers for baseline DCT
struct JPGImage {
    // JPEG object
};

struct BMPImage {
    // BMP object
};

//Quantization tables
//huffman tables

#endif
```

## CHƯƠNG 4: KẾT QUẢ VÀ THẢO LUẬN

### 4.1. Kết quả thu được từ chương trình mô phỏng

Chương trình mô phỏng mã hóa JPEG sử dụng mã Huffman đã hoàn thành và được kiểm thử với các file ảnh mẫu.



HÌNH 1: ẢNH GỐC (BMP)



HÌNH 2: ẢNH SAU KHI NÉN (JPG)

Kết quả cho thấy chương trình có khả năng nén ảnh với tỉ lệ nén khá tốt, giảm thiểu kích thước file ảnh (903kb xuống 61kb) mà không làm mất đi nhiều chi tiết.

### 4.2. So sánh với các phương pháp nén khác

Để đánh giá hiệu quả của chương trình, chúng em đã so sánh với các phương pháp nén khác như:

- **Nén không mất dữ liệu (PNG):** Mặc dù các phương pháp này không làm mất dữ liệu, kích thước file nén thường lớn hơn so với JPEG sử dụng mã Huffman.
- **Nén mất dữ liệu (JPEG tiêu chuẩn):** Kết quả nén bằng mã Huffman thường có tỉ lệ nén tốt hơn, tuy nhiên, thời gian xử lý lâu hơn do tính phức tạp của thuật toán.

### 4.3. Đánh giá hiệu quả

Hiệu quả của phương pháp nén bằng mã Huffman được đánh giá qua các tiêu chí:

- **Tỉ lệ nén:** Cao hơn so với các phương pháp nén không mất dữ liệu.
- **Chất lượng ảnh:** Giữ lại được nhiều chi tiết, ít bị suy giảm chất lượng so với JPEG tiêu chuẩn.
- **Thời gian xử lý:** Lâu hơn do quá trình xây dựng cây Huffman và mã hóa từng pixel.

## 4.4. Khó khăn gặp phải và giải pháp khắc phục

### 4.4.1. Khó khăn

- **Xử lý kích thước file lớn:** Khi xử lý các file ảnh có kích thước lớn, thời gian nén và giải nén tăng đáng kể.
- **Chỉ nén được**
- **Tối ưu hóa thuật toán:** Thuật toán Huffman cần tối ưu hóa để cải thiện hiệu suất.
- **Kiểm thử và xác minh:** Đảm bảo tính đúng đắn của quá trình nén và giải nén đòi hỏi nhiều kiểm thử kỹ lưỡng.

### 4.4.2. Giải pháp khắc phục

- **Sử dụng cấu trúc dữ liệu hiệu quả:** Tối ưu hóa việc sử dụng cấu trúc dữ liệu để giảm thời gian xử lý.
- **Đa luồng:** Sử dụng xử lý đa luồng để cải thiện hiệu suất cho các file ảnh lớn.
- **Bộ nhớ đệm:** Sử dụng bộ nhớ đệm để lưu trữ tạm thời các kết quả trung gian, giảm thiểu thời gian truy cập bộ nhớ.

## 4.5. Kết luận

Chương trình mô phỏng mã hóa JPEG dùng mã Huffman đã đạt được các mục tiêu đề ra, cải thiện tỉ lệ nén và chất lượng ảnh. Tuy nhiên, cần tiếp tục nghiên cứu và cải tiến để khắc phục các hạn chế về thời gian xử lý và hiệu suất cho các ứng dụng thực tiễn.

## **CHƯƠNG 5. TỔNG KẾT**

### **5.1. Kết luận**

Project "Mô phỏng nén ảnh JPEG theo mã Huffman" đã hoàn thành mục tiêu đặt ra, đó là sử dụng C++, viết chương trình mô phỏng quá trình nén BMP sang JPEG thông qua các bước từ chuyển đổi phổ màu, DCT, lượng tử hóa đến mã hóa Huffman.

Sau khi nén JPEG 75%, dung lượng của ảnh đã giảm đi rõ rệt, trong khi độ phân giải và màu sắc không bị thay đổi rõ rệt so với ảnh gốc. Đây là minh chứng cho hiệu quả của format JPEG, đặc biệt là vai trò quan trọng của mã hóa Huffman trong quá trình này. Thông qua việc mã hóa, nhóm đã hiểu rõ hơn về nguyên lý hoạt động và ứng dụng thực tiễn của thuật toán nén Huffman.

### **5.2. Hướng phát triển tương lai**

Dựa trên nền tảng và kết quả đã đạt được, chương trình mô phỏng nén ảnh dựa trên mã Huffman được nhóm đề xuất hướng phát triển và mở rộng trong tương lai:

- Xử lý tốt hơn các cấu trúc khác của file BMP: đọc và xử lý các file BMP có offset động, header không tuân theo cấu trúc truyền thống,.. Từ đó tăng tính ứng dụng của chương trình.
- Sau khi xử lý được nhiều format lưu trữ ảnh (BMP, PNG, RAW), đóng gói chương trình thành một driver phục vụ cho việc xử lý biên trong các thiết bị nhúng.

### **5.3. Lời cảm ơn**

Thông qua project này, nhóm đã nắm được chuẩn file BMP và JPEG, quy trình xử lý file, các bước nén và áp dụng mã Huffman và triển khai code, soạn thảo báo cáo.

Mặc dù vậy, thời gian thực hành còn ngắn hạn nên có rất nhiều khía cạnh và tính năng nâng cao của quá trình nén ảnh mà em vẫn chưa thể khám phá hết. Bên cạnh đó, với vốn kiến thức hạn hẹp, bài báo cáo của em chắc chắn không tránh khỏi những thiếu sót và sai lầm. Em mong nhận được sự góp ý từ thầy và các bạn để em có thể hoàn thiện kiến thức của mình.

Cuối cùng, em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Hữu Phát đã giúp đỡ chúng em nhiệt tình trong quá trình học tập và hoàn thiện bài thực hành lần này. Sự hướng dẫn tận tình của thầy đã giúp chúng em có thêm niềm đam mê và động lực theo đuổi ngành học mà mình đã chọn. Chúng em xin kính chúc thầy luôn mạnh khỏe và công tác tốt.

Chúng em xin chân thành cảm ơn!



## TÀI LIỆU THAM KHẢO

- [1] Microsoft , "Open Specifications," [Online]. Available:  
[https://learn.microsoft.com/en-us/openspecs/windows\\_protocols/ms-rdprfx/b550d1b5-f7d9-4a0c-9141-b3dca9d7f525?redirectedfrom=MSDN](https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-rdprfx/b550d1b5-f7d9-4a0c-9141-b3dca9d7f525?redirectedfrom=MSDN).