C Programming

WITH NETWORK PROTOCOL

IMAGE PROCESSING

Ali Sharafian - Parsa HaghighatGoo

Professor : Dr. Ali Hamzeh

semester spring - 2023

INTRODUCTION

In this project, we designed a program using C programming language in which a photo is given by the user and 20 filters are given to the user to apply on the photo. This program supports 3 types of photo formats: jpeg - jpg - png - bmp. The user has 3 ways to give photos and input to the program:

1. by giving the address of the photo on the system
2. sending the image to app's email (parsaali0002@gmail.com)
3. by giving the url of the image

Contents

1 Input

Here we talk about how the user can give the picture to the program.
There is 3 way for doing it:

1.1 System Files

The first way is using pictures which exist in the computer.
The user should write the path of pictures.

1.2 Sending The Picture To Program's E-Mail

the second way is that the user would email the desired picture to the email address (parsaali0002@gmail.com). we use the curllib for connecting to email and fetching attachments.

1.3 Giving The Photo From Url

the third way is the user should give the URL address relevant to desired picture to the program the program downloads the picture and make it ready for future procceses.
In this case the user can choose between two way:

- Url with socket
- Url with curl

2 Photo Filters

In this program, we have designed 20 filters for the user so that the user can apply them on the desired photo.

- Hue
- Solarize
- Revert
- Grayscale
- Sepia
- Brn
- Off Grid
- Cybr
- Sharpen
- Noise
- Vin4
- Warm color
- Comic
- Oil Painting
- Water Color
- Motion Blur
- Polygon
- Cross
- Neon
- Mirror

3 Foramt Of Picture

This program supports 3 photo formats:

3.1 JPG-JPEG
3.2 PNG
3.3 BITMAP(BMP)

In this case we write a function for processing on bmp and use stblib for processing jpg and png.

---

4 Main.c

In the main function of this program, we gathered all of the functions and sections of the program together .
Then we use switch case to get input from user for choosing which format of picture he give to the program and which filte the user want to apply on picture and how the user want to give the path of picture to the program(Url ,e-mail or system file) .

4.1 Headers

Here's the C code for the headers and defines section:

```c
//HEADERS AND DIFINES
#include "stdio.h"
#include "stdlib.h"

#define STB_IMAGE_IMPLEMENTATION
#define STB_IMAGE_WRITE_IMPLEMENTATION

#include "stb_image.h"
#include "stb_image_write.h"
#include "stb_image_resize.h"
#include <conio.h>
#include "curl.h"
#include <string.h>
#include "math.h"
```

```
#include <winsock2.h>
#include <ws2tcpip.h>


#define line_len 10000

#pragma comment(lib, "ws2_32.lib")
```

4.2 Filters

This section is for applying the filters and filter's functions are written in it.

4.3 Format Functions

This section is for choosing format of the picture.
There is 3 function :
• Bmp • Jpg • Png
and each of them is about (opening the image , closing the image , applying filters , saving and writing images)

4.4 E-MAIL Section

In this section :
first . we fetched the email and attachment with this c code .

```
void fetchemail() {
    //GMAIL DL PIC
    CURL *curl;
    CURLcode res;
    FILE *file;

    // Set up libcurl
    curl_global_init(CURL_GLOBAL_DEFAULT);
    curl = curl_easy_init();
    if (curl) {
        // Set IMAP server and account details
        char email[] = "parsaali0002@gmail.com"; // Replace with your Gmail
↪   address
        char password[] = "sjkhqzzmcrjidifo"; // Replace with your Gmail
↪   password
```

```
14
15          curl_easy_setopt(curl, CURLOPT_CAINFO,
16                  "example path");
17
18          // Set the mailbox name and UID
19          char mailbox[] = "INBOX";
20          int uid = 6;
21
22          ///In this section u can add uid functions (plus or mines) for
23          ///calling and getting uid from file for saving the uid and change it
24          ///after each email
25
26
27          // Create the URL with the UID
28          char url[100];
29          sprintf(url, "imaps://imap.gmail.com/%s;UID=%d", mailbox, uid);
30
31          // Connect to the IMAP server and log in
32          curl_easy_setopt(curl, CURLOPT_USERNAME, email);
33          curl_easy_setopt(curl, CURLOPT_PASSWORD, password);
34          curl_easy_setopt(curl, CURLOPT_URL, url);
35
36          // Perform the fetch request
37          res = curl_easy_perform(curl);
38          if (res != CURLE_OK) {
39              printf("Failed to fetch email: %s\n", curl_easy_strerror(res));
40              return;
41          }
42
43          // Open a file to save the fetched email
44          file =
    ↪ fopen("C:\\Users\\parca\\Desktop\\networkingphotoshopproject\\att.txt",
    ↪ "wb");
45          if (!file) {
46              printf("Error opening file for writing.");
47              return;
48          }
49
50          // Set the write callback function to write data into the file
51          curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, writeCallback);
52          curl_easy_setopt(curl, CURLOPT_WRITEDATA, file);
53
```

```
        // Perform the fetch request and save the email content to the file
        res = curl_easy_perform(curl);
        if (res != CURLE_OK) {
            printf("Failed to fetch email: %s\n", curl_easy_strerror(res));
            return;
        }

        // Clean up and close the file
        fclose(file);

        // Clean up and close the IMAP session
        curl_easy_cleanup(curl);
    } else {
        printf("Error initializing libcurl.\n");
        return;
    }

    // Clean up libcurl
    curl_global_cleanup();

}

```

```
size_t writeCallback(void *contents, size_t size, size_t nmemb, void *userp)
↪ {
    FILE *file = (FILE *) userp;
    size_t written = fwrite(contents, size, nmemb, file);
    return written;
}

```

2.cut the desired fetched text for decoding with this c code :

```
//writing decode fetched txt to a file
void finddecodetxt(const char *text, const char *outputFileName) {
    char check[16] = "X-Attachment-Id:";
    const char *decodetxtstart = strstr(text, check);
    if (decodetxtstart == NULL) {
        printf("there ins't exist fetch decode txt!\n");
        return;
    }

```

```
    decodetxtstart += strlen("X-Attachment-Id:");

    decodetxtstart = strstr(decodetxtstart, "\n\n");
    if (decodetxtstart == NULL) {
        printf("Invalid text format\n");
        return;
    }

    decodetxtstart += strlen("\n\n");

    const char *decodetxtend = strstr(decodetxtstart, "\n--");
    if (decodetxtend == NULL) {
        printf("Invalid text format. Base64 code end not found.\n");
        return;
    }

    size_t decodetxtlen = decodetxtend - decodetxtstart;
    char *decodetxt = (char *) malloc(decodetxtlen + 1);
    strncpy(decodetxt, decodetxtstart, decodetxtlen);
    decodetxt[decodetxtlen] = '\0';

    FILE *outputFile = fopen(outputFileName, "w");
    if (outputFile == NULL) {
        printf("Failed to open the decode txt file.\n");
        return;
    }

    fprintf(outputFile, "%s", decodetxt);
    fclose(outputFile);

    printf("image fetch txt extracted and saved to '%s'.\n", outputFileName);
    free(decodetxt);
}

//extract decode fetched txt from total fetched file txt
void extractdecodetxt() {
    FILE *att;
    char line[line_len];
    const char *decode =
↪   "C:\\Users\\parca\\Desktop\\networkingphotoshopproject\\decode.txt";

```

```
    att =
↪   fopen("C:\\Users\\parca\\Desktop\\networkingphotoshopproject\\att.txt",
↪   "r");
    if (att == NULL) {
        printf("\nError in Opening the attachment txt temp");
        getch();
        return;
    }

    fseek(att, 0, SEEK_END);
    long file_size = ftell(att);
    rewind(att);

    char *temp = (char *) malloc(file_size + 1);
    fread(temp, sizeof(char), file_size, att);
    temp[file_size] = '\0';

    finddecodetxt(temp, decode);

    free(temp);
    fclose(att);
}

```

3.and after that we decode the image fetched txt
you can see the decode's code in decode section.

4.5 Url Section

In this section we talk about our 2 way for downloading from desired url

4.5.1 Socket

Here's C code for downloading photo from url with socket protocols

```
void dlfromsocket() {
    WSADATA wsaData;
    SOCKET sock;
    struct addrinfo hints, *res;
    char request[1024];
    int bytesReceived, totalBytesReceived;
    FILE *fp;
```

```
8
9       // Initialize Winsock
10      if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0) {
11          printf("WSAStartup failed\n");
12          return;
13      }
14
15      // Create socket
16      sock = socket(AF_INET, SOCK_STREAM, 0);
17      if (sock == INVALID_SOCKET) {
18          printf("Error creating socket\n");
19          WSACleanup();
20          return;
21      }
22
23      // Resolve URL to IP address
24      memset(&hints, 0, sizeof(hints));
25      hints.ai_family = AF_INET;
26      hints.ai_socktype = SOCK_STREAM;
27      if (getaddrinfo("www.cs.sjsu.edu", "80", &hints, &res) != 0) {
28          printf("Error resolving URL\n");
29          closesocket(sock);
30          WSACleanup();
31          return;
32      }
33
34      // Connect to server
35      if (connect(sock, res->ai_addr, res->ai_addrlen) == SOCKET_ERROR) {
36          printf("Error connecting to server\n");
37          freeaddrinfo(res);
38          closesocket(sock);
39          WSACleanup();
40          return;
41      }
42
43      // Send HTTP GET request
44      sprintf(request,
45              "GET /~pearce/modules/lectures/web/html/HTTP_files/image001.jpg
    ↪ HTTP/1.1\r\nHost:www.cs.sjsu.edu\r\n\r\n");
46      send(sock, request, strlen(request), 0);
47
48      // Receive response
```

```
    totalBytesReceived = 0;
    fp = fopen("./tempsocket.txt", "wb");
    while ((bytesReceived = recv(sock, request, sizeof(request), 0)) > 0) {
        fwrite(request, 1, bytesReceived, fp);
        totalBytesReceived += bytesReceived;
        printf("%c", bytesReceived);
    }


    // Clean up
    freeaddrinfo(res);
    closesocket(sock);
    WSACleanup();
    fclose(fp);

    printf("Downloaded %d bytes\n", totalBytesReceived);
    return;
}

```

This is how we download the picture from url with socket.

This C code is about when we download the image we should split the encoded text for writing the image.

```
void cuttheencodedtxtofimagefromdltxt() {
    FILE *fs, *fn;
    fs = fopen("example path", "rb");
    if (fs == NULL) {
        printf("\nError in Opening the dledtxt file");
        getch();
        return;
    }
    fn = fopen("C:\\project\\fetch\\tempcutsocket.txt", "w");
    if (fn == NULL) {
        printf("\nError in Opening the encodedtxtofimage file");
        getch();
        return;
    }
    char c;
    int count = 0, flag = 0;
    while (1) {
```

```
        if (feof(fs)) {
            break;
        }
        c = fgetc(fs);
        if (c == '\n') {
            count++;
            if (count == 2) {
                flag = 1;
                break;
            }
        } else {
            count = 0;
        }
    }
    int i = 0;
    while (flag) {
        if (feof(fs)) {
            break;
        }
        c = fgetc(fs);
        fputc(c, fn);
        i++;
        if (i == 274) {
            continue;
        }
    }
    fclose(fs);
    fclose(fn);
    printf("cut done!\n");
    return;
}

```

And the c code is below is about how we call socket's way.

```
void urlwithsocket() {
    dlfromsocket();
    cuttheencodedtxtofimagefromdltxt();
    return;
}

```

4.5.2 CURL

And Here is our curl protocol for download image from url.

```c
void download_image(const char *url, const char *filename) {
    FILE *file = fopen(filename, "wb");
    if (!file) {
        printf("Failed to open the file for writing.\n");
        return;
    }

    // Create the command to download the image using curl
    char command[256];
    snprintf(command, sizeof(command), "curl -o \"%s\" \"%s\"", filename,
    ↪ url);

    // Execute the command using the system function
    int result = system(command);

    if (result == 0) {
        printf("Image downloaded successfully.\n");
    } else {
        printf("Failed to download the image.\n");
    }

    fclose(file);
}

void url() {

    //url for example:(this is a png file)
    //const char* url = int c, i = 0, j = 0, m;
    char url[1000], path[1000];
    printf("please write your url link:");
    getchar();
    for (; (c = getchar()) != '\n'; i++) {
        url[i] = c;
        url[i + 1] = '\0';
    }

    printf("please write your path for saving downloded image :");
    for (; (c = getchar()) != '\n'; j++) {
```

```
        path[j] = c;
        path[j + 1] = '\0';
    }

    //const char* filename = "c";
    download_image(url, path);

    printf("%s", url);
    printf("\n%s", path);

    if (path[j - 1] == 'p') {
        bmp(path);
    } else if (path[j - 2] == 'n') {
        png(path);
    } else {
        jpg(path);
    }
    return;
}
```

4.6 Main Section

4.7 Rating Section

```
printf("if you are satisfied ,please enter a number between 1 and 10 for
↪  rating to the program and you dont please enter 0:");
    scanf("%d", &rate);

    printf("\nyour rate is : %d", rate);

```

This C code is for rating the program.

4.8 Contact With Us

```
printf("\nIf you want to contact with us you can send your opinion or problem
↪  to our program's e-mail (parsaali0002@gmail.com)!");

```

This C code explain how user can contact with us .

5 Decode

Here's the C code for decoding the image

```c

//decodefile func for decoding file to img
void decodeFile(const char *inputFile, char outputFile[]) {
    char command[456];
    sprintf(command, "certutil -decode %s %s", inputFile, outputFile);
    system(command);
}

//decoding sec
void decodefetchtxttoimg(char path[], int i) {
    const char *inputFileName =
↪   "C:\\Users\\parca\\Desktop\\networkingphotoshopproject\\decode.txt";

    printf("\n%s", path);
    decodeFile(inputFileName, path);

    if (path[i - 1] == 'p') {
        bmp(path);
    } else if (path[i - 2] == 'n') {
        png(path);
    } else {
        jpg(path);
    }
    return;
}

```

6 UID

in this section we write 2 function for saving uid after and change it after sending each email. These functions :

- uidmines
- uidplus

```
int uidplus() {
    char c;
    int uid = 0;
    FILE *file =
↪   fopen("C:\\Users\\parca\\Desktop\\networkingphotoshopproject\\uid.txt",
↪   "r");
    while (1) {
        if (feof(file)) {
            break;
        }
        c = fgetc(file);
        if (c >= '0' && c <= '9') {
            uid = (uid * 10) + (c - '0');
        }
    }
    rewind(file);
    fclose(file);
    file =
↪   fopen("C:\\Users\\parca\\Desktop\\networkingphotoshopproject\\uid.txt",
↪   "w");
    int i = uid + 1;
    fprintf(file, "%d", i);
    fclose(file);
    return uid;
}

```

```
int uidmines() {
    char c;
    int uid = 0;
    FILE *file =
↪   fopen("C:\\Users\\parca\\Desktop\\networkingphotoshopproject\\uid.txt",
↪   "r");
    while (1) {
        if (feof(file)) {
            break;
        }
```

```
9           c = fgetc(file);
10          if (c >= '0' && c <= '9') {
11              uid = (uid * 10) + (c - '0');
12          }
13      }
14      rewind(file);
15      fclose(file);
16      file =
    ↪   fopen("C:\\Users\\parca\\Desktop\\networkingphotoshopproject\\uid.txt",
    ↪   "w");
17      int i = uid - 1;
18      fprintf(file, "%d", i);
19      fclose(file);
20      return 0;
21  }
```

7 GUI

we desiged 2 gui for this progrm but we don't link them to our c program.
1.first one with QT and python.
2.second one with powershell gui.

8 LaTex

We Write this document with LaTex.

9 Thanks

We hope this document is useful for you.