Data Structures and Algorithms
Bitmap Project

Group members:

Rufat Huseynov
Shirin Shukurov

January 2020

1 Introduction

This project is done by Shirin Shukurov and Rufat Huseynov. There are 8 files inside the project folder: 4 of them are C files (bmp.c, convert.c, morse.c, main.c), 3 of them are header files (bmp.h, convert.h, morse.h) and 1 bmp file (sign.bmp). The functions are declared in header files, source codes are written in C files.

2 bmp.c and bmp.h

bmp.h is included in bmp.c One structure BMP and 3 functions (getImageInfo(), createOutput(), getDate()) are declared in bmp.h file.

getImageInfo() - This function takes name of the source file as argument and gets information about bitmap picture and stores them into BMP structure and returns that structure.

createOutput() - This function takes BMP structure, text, date flag(presence of date in the command), color code, position and name of output file as argument. Then it creates new file with name that user provide and writes corresponding data into that file.

getDate() -This function returns date command of linux system.(current date and time)

3 convert.c and convert.h

convert.h is included in convert.c 4 functions are declared in convert.h file: stringToStrHex(), LittleBigEndiannes(), HexToInt(), hexToStr().

stringToStrHex() - converts text to hexadecimal

LittleBigEndiannes() - receives endian as a parameter, if endian is big, it converts it to little endian, if endian is little, it converts it to big endian

HexToInt - receives hexadecimal character and changes it to an integer

hexToStr() - receives hexadecimal character and changes it to string

4 morse.c and morse.h

morse.h is included in morse.c One function(morseTable()) is declared in morse.h.

morseTable() - receives a character and converts it to its morse code

5 User Manual

To compile the program use this:
gcc main.c bmp.c convert.c morse.c -std=gnu99

Note:Do not use -std=c99. It can cause errors

If you compile the program with the command that we provide you will get a.out file. To start the program you can use this:
./a.out sign.bmp –text hello –date –pos 600,200 –o modsign.bmp
In the code that I provide above, "sign.bmp" is the source file that you want to watermark. "modsign.bmp" is the name of the output file that you want.
Note: In case of giving wrong file (not .bmp file) program will show error and automatically exit.
If you don't provide text and date it will give error and exit. You should provide one of them:
./a.out sign.bmp –text hello –pos 600,200 –o modsign.bmp

or

./a.out sign.bmp –date –pos 600,200 –o modsign.bmp

Also if there is no color code provided by the user the default code will be ABCEDF (please use real color codes not some garbage values). And if the user don't give the position it will be automatically (0,0).

Note: If the user tries to give invalid position program will automatically exit.

NOTE: This program is consider use only for English language. Using french letters that don't exist in english alphabet could cause the result that you don't want. So use only english letters.

6 Conclusion

Working on this project was very interesting. We learned how to deal with bytes, how to use big little-endian values, watermark.