МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

ПО КУРСОВОЙ РАБОТЕ
по дисциплине «Программирование»

Тема: Обработка бинарных файлов

Студент гр. 9381 _____ Чернышев Г.В.

Преподаватель _____ Берленко Т.А.

Санкт-Петербург
2020

ЗАДАНИЕ

НА КУРСОВУЮ РАБОТУ

Студент Чернышев Г.В.

Группа 9381

Тема работы : Обработка бинарных файлов

Исходные данные:

-ОС Linux

-Язык программирования С

-CLI

Содержание пояснительной записки:

-Содержание

-Введение

-Формулировка задания

-Разработка программы

-Заключение

-Тестирование

Предполагаемый объем пояснительной записки:

Не менее 20 страниц.

Дата выдачи задания: 01.03.2020

Дата сдачи реферата: 01.10.2020

Дата защиты реферата: 01.10.2020

Студент _____ Чернышев Г.В.

Преподаватель _____ Берленко Т.А.

АННОТАЦИЯ

Цель курсовой работы заключается в написании приложения для обработки ихображений формата PNG. В качестве интерфейса программы был выбран интерфейс командной строки. Для его реализации использовалась библиотека getopt. Для считывания, обработки и сохранения файлов формата PNG была использована библиотека libpng. В ходе работы было спешно разработано приложение для работы с файлами формата PNG. При разработке использовался язык программирования C.

SUMMARY

The purpose of the course work is to write an application for processing PNG images. The command line interface was chosen as the program interface. The getopt library was used to implement it. The libpng library was used to read, process and save PNG files. In the course of the work, an application for working with PNG files was hastily developed. The development used the C programming language.

СОДЕРЖАНИЕ

	Введение	4
1.	Наименования разделов	5
1.1.		0
1.2.		0
2.		0
2.1.		0
2.2.		0
3.		0
3.1.		0
3.2.		0
	Заключение	0
	Список использованных источников	0
	Приложение А. Название приложения	0

ВВЕДЕНИЕ

Цель работы.

Разработка программы для считывания, обработки и сохранения изображения в формате PNG.

Задачи.

-Изучение языка программирования C;

-Изучение библиотеки getopt для реализации интерфейса командной строки;

-Изучение сигнатуры формата PNG;

-Изучение библиотеки libpng для работы с изображениями формата PNG;

-Написание исходного кода программы;

-Сборка программы;

- Тестирование программы.

1. ФОРМУЛИРОВКА ЗАДАНИЯ

Вариант 13.

Программа должна иметь CLI или GUI. Более подробно тут: http://se.moevm.info/doku.php/courses:programming:rules_extra_kurs

Программа должна реализовывать весь следующий функционал по обработке png-файла

Общие сведения

Формат картинки PNG (рекомендуем использовать библиотеку libpng)

без сжатия

файл всегда соответствует формату PNG

обратите внимание на выравнивание; мусорные данные, если их необходимо дописать в файл для выравнивания, должны быть нулями.

все поля стандартных PNG заголовков в выходном файле должны иметь те же значения что и во входном (разумеется кроме тех, которые должны быть изменены).

Программа должна реализовывать следующий функционал по обработке PNG-файла

(1) Рисование окружности. Окружность определяется:

либо координатами левого верхнего и правого нижнего угла квадрата, в который она вписана, либо координатами ее центра и радиусом

толщиной линии окружности

цветом линии окружности

окружность может быть залитой или нет

цветом которым залита сама окружность, если пользователем выбрана залитая окружность

(2) Отражение заданной области. Этот функционал определяется:

выбором оси относительно которой отражать (горизонтальная или вертикальная)

Координатами левого верхнего угла области

Координатами правого нижнего угла области

(3) Копирование заданной области. Функционал определяется:

Координатами левого верхнего угла области-источника

Координатами правого нижнего угла области-источника

Координатами левого верхнего угла области-назначения

Разработка программы

Для удобства разработки интерфейса командной строки была использована библиотека getopt, позволяющая удобно обрабатывать массив данных подаваемых пользователем на вход программе. Для считывания и записи файлов в формате PNG была использована библиотека libpng. Разработка программы началась с написания структур и функции для работы с PNG-файлом. Так же были написаны перечисления и вспомогательные функции.

Перечисление Command

```
typedef enum
{
   COMMAND_NONE = 0,
   COMMAND_DRAW_CIRCLE = 1,
   COMMAND_REFLECTION_IMAGE = 2,
   COMMAND_COPY_IMAGE = 3,
} Command;
```

Хранит тип комманды.

Структура Color

```
typedef struct {
   int r;
```

```
    int g;
    int b;
    int a;
} Color;
```

Используется для храения цвета пикселя.

Структура Options

```
typedef struct
{
  Command command;
  int info;//флаг информации об изображении
  int circle_center_x;
  int circle_center_y;//координаты центра круга
  int radius;//радиус круга
  int circle_xl, circle_yl;
  int circle_xr, circle_yr;//координаты описанного квадрата
  int lineWidth;//ширина линии окружности
  Color lineColor;//цвет линии окружности
  int fill;//флаг заливки окружности
  Color fillColor;//цвет заливки окружности

  char axis;//ость отражения заданной области
  int reflection_xl, reflection_yl;
  int reflection_xr, reflection_yr;//координаты заданной области для отражения

  int mirror_xl, mirror_yl;
  int mirror_xr, mirror_yr;//координаты области-источника для копирования
```

int mirror_x_result, mirror_y_result;//координаты области, куда производится копирование

char imageInputPath;//путь для чтения изображения*

char imageOutputPath;//путь для сохранения изображения*

} Options;

Используется для хранения данных о переданных пользователем параметрах.

Структура Image

typedef struct{

char path;//путь к изображению*

int width;//размер в пискелях изображения по Ox

int height;//размер в пикселях изображения по Oy

png_byte colorType;//цветовая модель

png_byte bitDepth;//глубина цвета

png_byte interlaceType;//тип interlace

png_byte filterMethod;//тип фильтрации

png_byte compressionType;//тип сжатия

png_bytepp pixels;//двумерный массив пикселей

} Image;

Хранит основную информацию об изображении

Функция printHelp()

выводит на экран справку о программе.

Функция initOptionStruct()

Возвращает экземпляр структуры Option, поля которого заполнены по умолчанию.

Функция destroyOptionStruct()

В качестве аргумента получает указатель на экземпляр структуры Option. Освобождает память, выделенную динамически под экземпляр.

Функция InitImage()

Функция создает в динамической памяти экземпляр структуры Image. Заполняет поля экземпляра значениями по умолчанию. Функция возвращает указатель на созданный экземпляр.

Функция DestroyImage()

Принимает указатель на экземпляр структуры подлежащий удалению. С помощью функции free(). Освобождает память, выделенную под экземпляр сруктуры, а так же его поля (двумерный массив хранящй информацию о пикселях и строку содержащюю путь к файлу на компьютере.

Функция readImage()

Считывает изображение формата PNG из файла, расположенного по пути path, и сохраняет информацию о нем в структуру Image. На вход принимает два аргумента: image - указатель на экземпляр структуры Image и path - путь до файла изображения. Если при считывании изображения возникнет ошибка, то функция вернет код ошибки, равный 1. В случае, если изображение будет успешно считано, то функция вернет 0.

Функция saveImage()

Функция принимает на вход строку, содержащюю путь к файлу, в который необходимо сохранить файл.Сохраняет изображение формата PNG в файл, расположенный по пути path (информация о изображении берется из структуры Image). На вход принимает два аргумента: image - указатель на экземпляр структуры Image и path - путь до файла изображения. Если при сохранении изображения возникнет ошибка, то функция вернет код ошибки,

равный 1. В случае, если изображение будет успешно сохранено, то функция вернет 0.

Функция getImageColorType()

Функция принимает на фход указатель на экземпляр структуры Image и возвращает тип цветовой модели.

Функция getImagePixelSize()

Функция принимает на фход указатель на экземпляр структуры Image и возвращает значение — количество цветов в палитке.

Функция printImageInfo()

Функция выводит на экран подробную информацию об изображении.

Функция copePartOfImage()

Функция осуществляет копирование пикселей из одной заданной области изображения в другую. На вход подаются координаты областей и указатель на само изобрадение. Сначала Область для копирования сохраняется в буффер, массив пикселей, затем переностится в область, предназначенную для копирования.

Функция reflectionOfPartOfImage()

Функция осуществляет отражение заданной области по оси Ox или Oy. Аналогично предыдущей функции сначала уже отображенная область сохраняется в буфер. Затем копируется обратно. В случает корректной работы функция вернет 0. Если входные данные оказались некорректны, функция вернет целое значение, отличное от нуля.

Функция setPixel()

Функция принимает на вход три параметра, указатель на само изображение, указатель на массив, характеризующий пиксель и экземпляр структуры Color, храняший информацию о цвете, на который надо заменить цвет пикселя.

Функция makeColor()

Создает экземпляр структуры Color. Входные параметры помещаются в поля экземпляра структуры.

Функция checkColor()

Возвращает 1, если значения полей экземпляра структуры Color (входного параметра) корректны. В противном случает функция возвращает 0.

Функция CheckPoint()

Принимает на вход экземпляр структуры Image и проверяет, корректно ли заданы координаты точки (входные параметры). Функция возвращает результат логического выражения.

Функция circleDrawingWithRadiusAndCenter()

Функция генерирует окружность, на вход подаются координаты центра, радиус, ширина линии окружности, флаг заливки, цвет заливки (экземпляр структуры Colour) и указатель на само изображение. Функция определяет пиксели, которые необходимо закрасить, проверкой по формуле окружности (x-x0)^2 + (y — y0)^2 =R^2. Если окружность необходим закрасить, также происхожит проверка со знаком <.

Функция circleDrawingWithCircumscribedSquare()

Аналогично предыдущей функции данная рисует окружность. Входными параметрами являются координаты квадрата, в который данная окружность вписана. Все вычисления аналогичны работе предыдущей функции, кроме вычисления центра окружности и радиуса.

Функция main()

Данная функция является точкой входа в программу. На вход принимает два аргумента: argc - количество переданных опций (включая название самой программы) и argv — опции, переданные пользователем. В начале происходит обработка переданных в программу опций с помощью библиотеки getopt_long и Switch() и сохранение их в структуру Options. Если при передачи опций возникли какие-либо ошибки, то будет отображена справка программы. Если все прошло успешно, то происходит считывание изображения из файла, далее следует его обработка и сохранение обработанного изображения в файл.

Запись в экземпляр структуры Options происходит в этом фрагменте кода.

```
int  option = getopt_long(argc, argv, optionsString, longOptions, &longIndex);
   while (option != -1)
   {
      switch (option)
      {
         case 'I':
            options -> info = 1;
            break;
         case 'C':
            if(options -> command == COMMAND_NONE)
               options -> command = COMMAND_DRAW_CIRCLE;
            else
               printf("command has already been selected, Skipping.\n");
            break;
         case 'R':
            if(options -> command == COMMAND_NONE)
               options -> command = COMMAND_REFLECTION_IMAGE;
            else
               printf("command has already been selected, Skipping.\n");
            break;
         case 'M':
```

```
                if(options -> command == COMMAND_NONE)
                    options -> command = COMMAND_COPY_IMAGE;
                else
                    printf("command has already been selected, Skipping.\n");
                break;
            case 'c':
                if(sscanf(optarg, "%d:%d", &(options -> circle_center_x), &(options ->
circle_center_y)) != 2)
                    printf("Failed to read center coordinate %s, try 'x:y'.\n", optarg);
                break;
            case 'r':
                if(sscanf(optarg,"%d", &(options -> radius)) != 1)
                    printf("Failed to read radius %s, try -r of --radius <int>.\n", optarg);
                break;
            case 'u':
                if(sscanf(optarg, "%d:%d", &(options -> circle_xl), &(options -> circle_yl)) != 2)
                    printf("Failed to read coordinate %s. try 'x:y'.\n", optarg);
                break;
            case 'd':
                if(sscanf(optarg, "%d:%d", &(options -> circle_xr), &(options -> circle_yr)) !=
2)
                    printf("Failed to read coordinate %s. try 'x:y'.\n", optarg);
                break;
            case 'w':
                if(sscanf(optarg, "%d", &(options -> lineWidth)) != 1)
                    printf("Failed to read width of line %s. tre '-w or --width <int>.\n", optarg);
                break;
            case 'l':
                if(sscanf(optarg,"%d:%d:%d", &(options -> lineColor.r), &(options ->
lineColor.g), &(options -> lineColor.b)) != 3)
                    printf("Recognize color %s fail, try 'r:g:b'.\n", optarg);
                break;
            case 'p':
                if(sscanf(optarg,"%d:%d:%d", &(options -> fillColor.r), &(options ->
fillColor.g), &(options -> fillColor.b)) != 3)
```

```
                printf("Recognize color %s fail, try 'r:g:b'.\n", optarg);
            options -> fill = 1;
            break;
        case 'a':
            if(strcmp(optarg, "x") == 0)
                options -> axis = 'x';
            else if (strcmp(optarg, "y") == 0)
                options -> axis = 'y';
            else
                printf("Axis set incorrectly try 'x or y'.\n");
            break;
        case 's':
            if(sscanf(optarg, "%d:%d", &(options -> reflection_xl), &(options ->
reflection_yl)) != 2)
                printf("Failed to read coordinate %s. try 'x:y'.\n", optarg);
            break;
        case 'f':
            if(sscanf(optarg, "%d:%d", &(options -> reflection_xr), &(options ->
reflection_yr)) != 2)
                printf("Failed to read coordinate %s. try 'x:y'.\n", optarg);
            break;
        case 'b':
            if(sscanf(optarg, "%d:%d", &(options -> mirror_xl), &(options -> mirror_yl)) !=
2)
                printf("Failed to read coordinate %s. try 'x:y'.\n", optarg);
            break;
        case 'e':
            if(sscanf(optarg, "%d:%d", &(options -> mirror_xr), &(options -> mirror_yr)) !=
2)
                printf("Failed to read coordinate %s. try 'x:y'.\n", optarg);
            break;
        case 'm':
            if(sscanf(optarg, "%d:%d", &(options -> mirror_x_result), &(options ->
mirror_y_result)) != 2)
                printf("Failed to read coordinate %s. try 'x:y'.\n", optarg);
```

```
            break;
        case 'i':
            options -> imageInputPath = (char*)calloc(strlen(optarg)+1, sizeof(char));
            strcpy(options -> imageInputPath, optarg);
            break;
        case 'o':
            options -> imageOutputPath = (char*)calloc(strlen(optarg)+1, sizeof(char));
            strcpy(options -> imageOutputPath, optarg);
            break;
        case 'h':
        case '?':
            printHelp();
            destroyOptionStruct(options);
            return 0;
        default:
            printf("Parameter '%s' not recognized.\n", argv[optind]);
            printHelp();
            destroyOptionStruct(options);
            return 0;

    }

    option = getopt_long(argc, argv, optionsString, longOptions, &longIndex);
}
```

Массив longOptions[] - Хранит данные о длинных опциях (название, сокращение, наличие аргументов.

Строка OptionsString — Хранит данные о сокращениях опций.

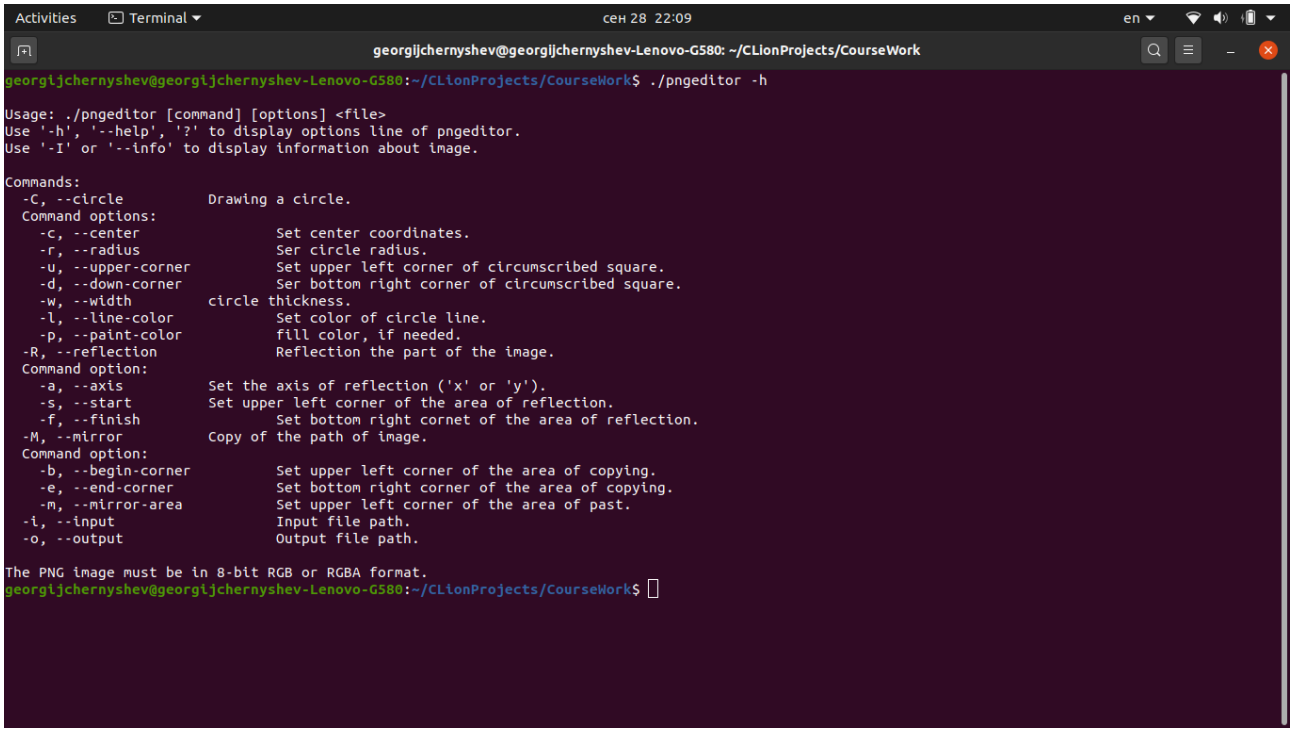
Далее с помощью Switch() просматриваются все переданные в программу опции. И сохраняются в экземпляр структуры Options.

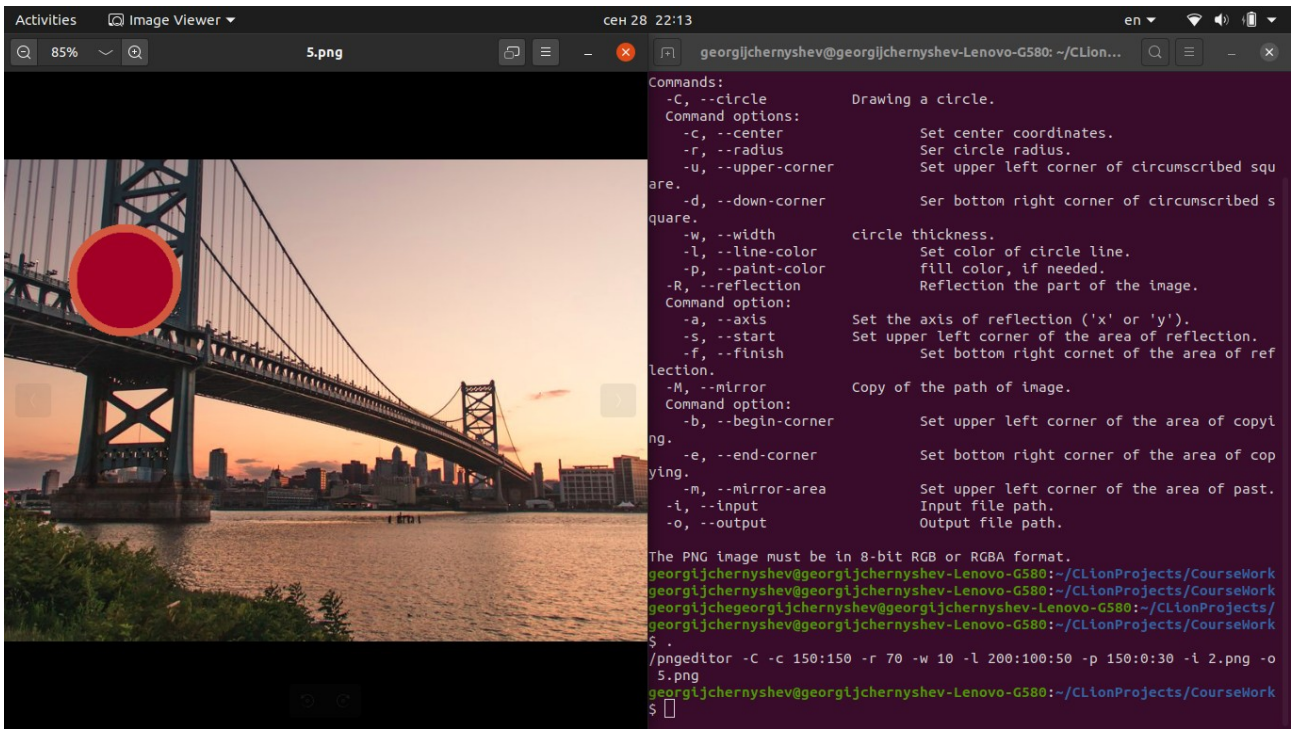
После этого Создается экземпляр структуры Image. И считывается бинарный файл, затем, с помощью Switch(). Определяется переданная пользователем опция. Происходит ее выполнение, в случае успеха

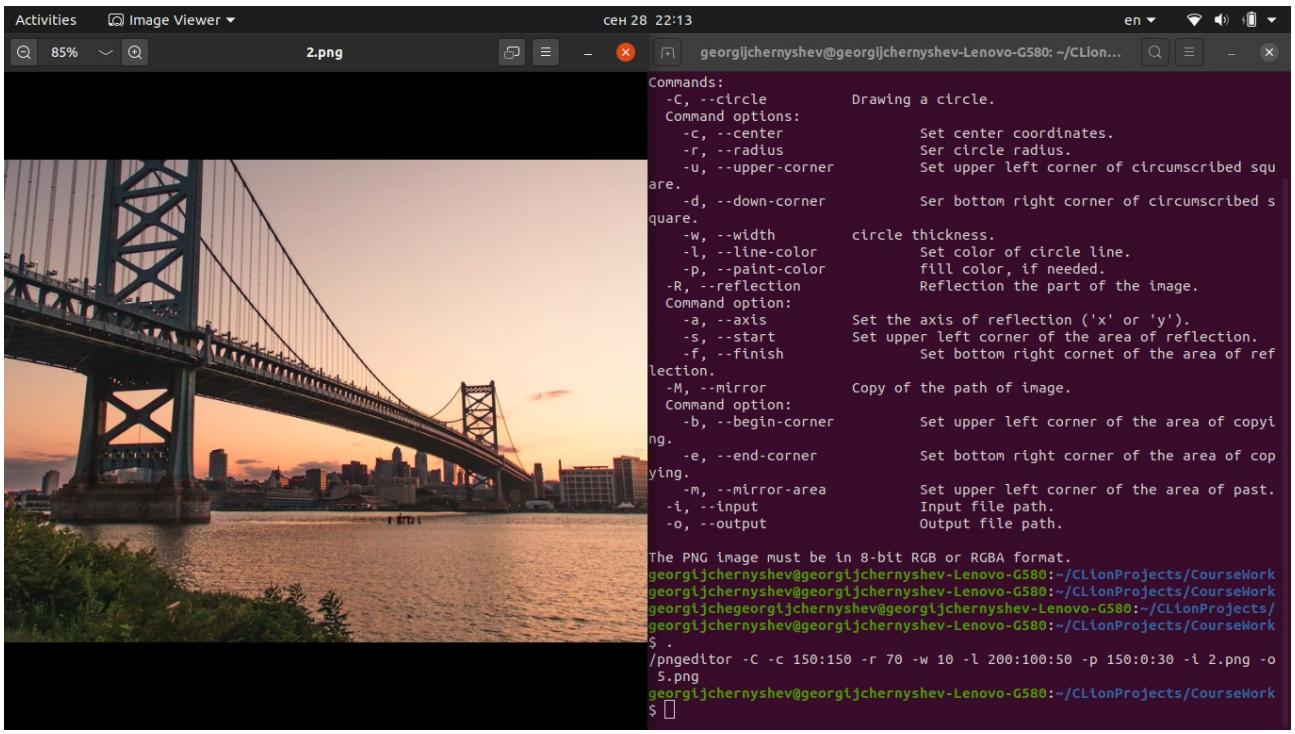
редактированный файл будет сохранен либо вместо переданного, либо сохранен в файле, указанном пользователем. В случае ошибки, пользователю будет выведена справка о приложении и\или информация об ошибке.

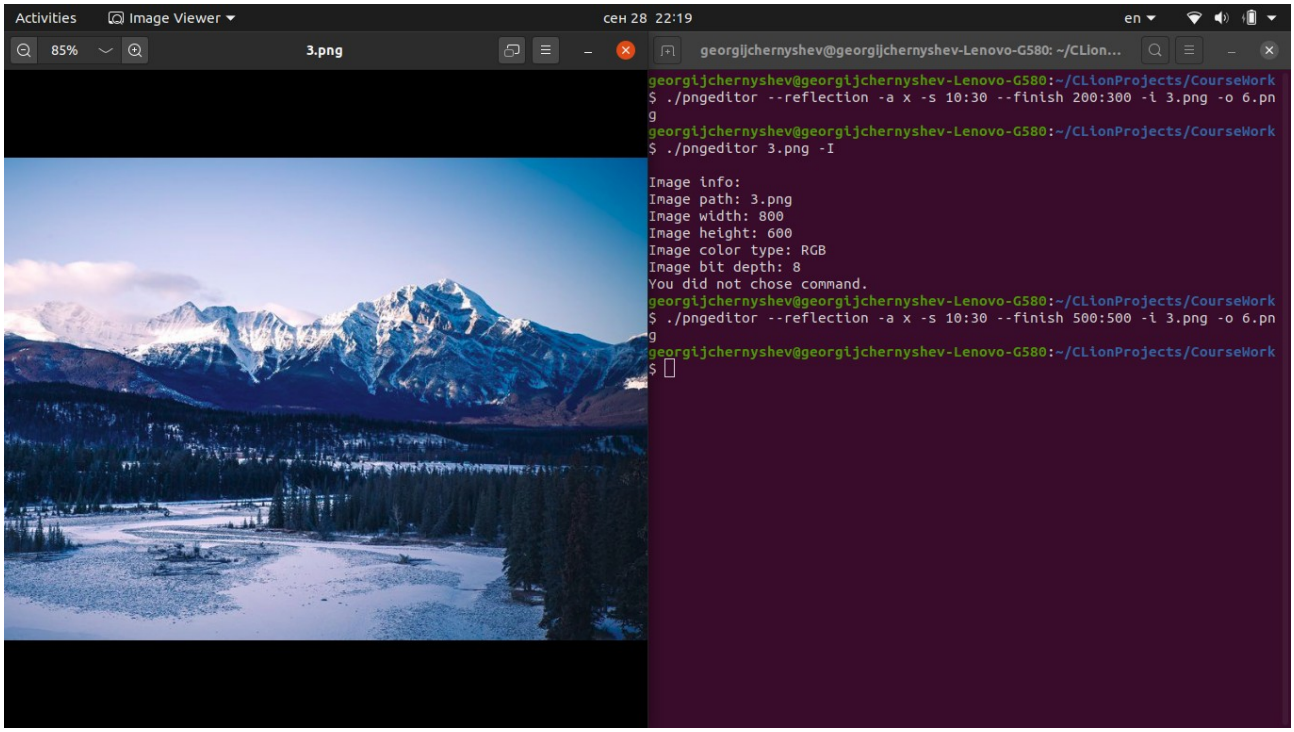
Тестирование работы программы.

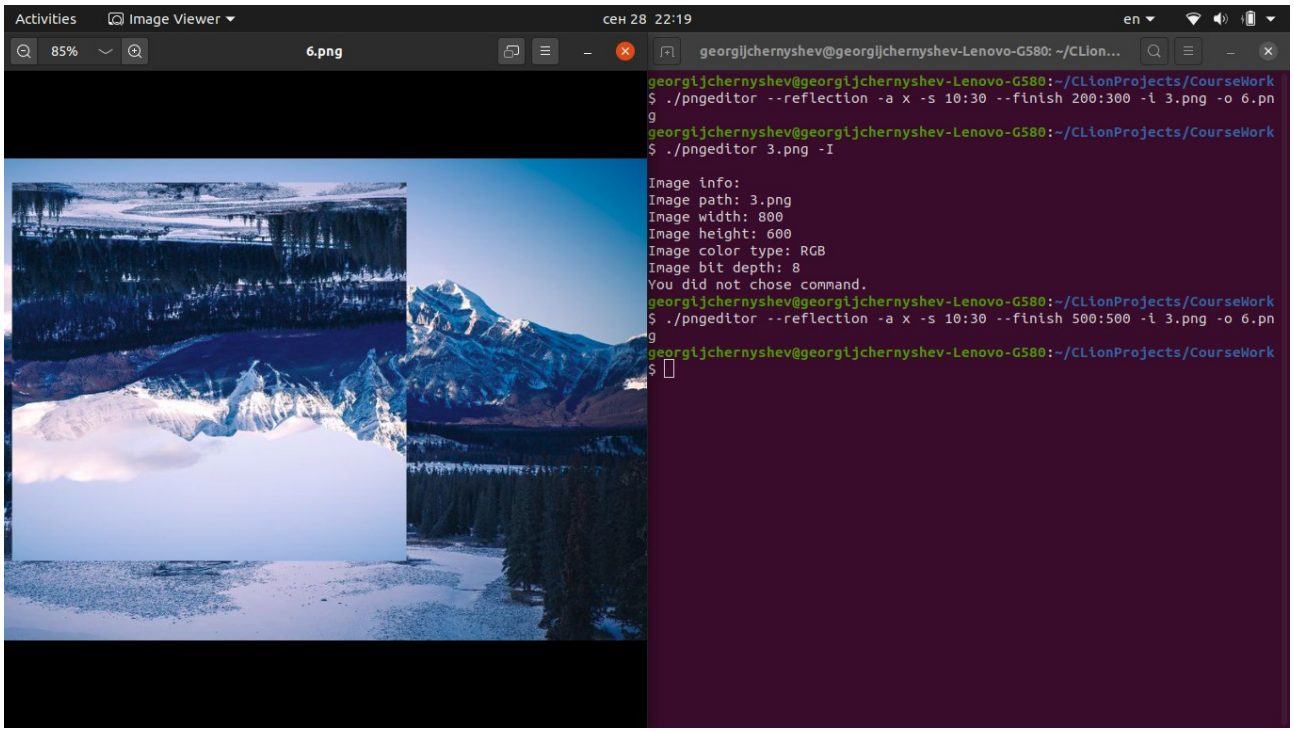
Тестирование работы функций.




Activities
Image Viewer
сен 28 22:13
en
85%
5.png
georgijchernyshev@georgijchernyshev-Lenovo-G580: ~/CLion...
Commands:
-C, --circle Drawing a circle.
Command options:
-c, --center Set center coordinates.
-r, --radius Ser circle radius.
-u, --upper-corner Set upper left corner of circumscribed square.
-d, --down-corner Ser bottom right corner of circumscribed square.
-w, --width circle thickness.
-l, --line-color Set color of circle line.
-p, --paint-color fill color, if needed.
-R, --reflection Reflection the part of the image.
Command option:
-a, --axis Set the axis of reflection ('x' or 'y').
-s, --start Set upper left corner of the area of reflection.
-f, --finish Set bottom right cornet of the area of reflection.
-M, --mirror Copy of the path of image.
Command option:
-b, --begin-corner Set upper left corner of the area of copying.
-e, --end-corner Set bottom right corner of the area of copying.
-m, --mirror-area Set upper left corner of the area of past.
-i, --input Input file path.
-o, --output Output file path.
The PNG image must be in 8-bit RGB or RGBA format.
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
georgijchegeorgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$ ./pngeditor -C -c 150:150 -r 70 -w 10 -l 200:100:50 -p 150:0:30 -i 2.png -o 5.png
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$


Activities
Image Viewer
сен 28 22:13
en
85%
2.png
georgijchernyshev@georgijchernyshev-Lenovo-G580: ~/CLion...
Commands:
-C, --circle Drawing a circle.
Command options:
-c, --center Set center coordinates.
-r, --radius Ser circle radius.
-u, --upper-corner Set upper left corner of circumscribed square.
-d, --down-corner Ser bottom right corner of circumscribed square.
-w, --width circle thickness.
-l, --line-color Set color of circle line.
-p, --paint-color fill color, if needed.
-R, --reflection Reflection the part of the image.
Command option:
-a, --axis Set the axis of reflection ('x' or 'y').
-s, --start Set upper left corner of the area of reflection.
-f, --finish Set bottom right cornet of the area of reflection.
-M, --mirror Copy of the path of image.
Command option:
-b, --begin-corner Set upper left corner of the area of copying.
-e, --end-corner Set bottom right corner of the area of copying.
-m, --mirror-area Set upper left corner of the area of past.
-i, --input Input file path.
-o, --output Output file path.
The PNG image must be in 8-bit RGB or RGBA format.
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
georgijchegeorgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$ ./pngeditor -C -c 150:150 -r 70 -w 10 -l 200:100:50 -p 150:0:30 -i 2.png -o 5.png
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$


Activities
Image Viewer
сен 28 22:19
en
85%
3.png
georgijchernyshev@georgijchernyshev-Lenovo-G580: ~/CLion...
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$ ./pngeditor --reflection -a x -s 10:30 --finish 200:300 -i 3.png -o 6.png
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$ ./pngeditor 3.png -I
Image info:
Image path: 3.png
Image width: 800
Image height: 600
Image color type: RGB
Image bit depth: 8
You did not chose command.
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$ ./pngeditor --reflection -a x -s 10:30 --finish 500:500 -i 3.png -o 6.png
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$


Activities
Image Viewer
сен 28 22:19
en
85%
6.png
georgijchernyshev@georgijchernyshev-Lenovo-G580: ~/CLion...
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$ ./pngeditor --reflection -a x -s 10:30 --finish 200:300 -i 3.png -o 6.png
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$ ./pngeditor 3.png -I
Image info:
Image path: 3.png
Image width: 800
Image height: 600
Image color type: RGB
Image bit depth: 8
You did not chose command.
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$ ./pngeditor --reflection -a x -s 10:30 --finish 500:500 -i 3.png -o 6.png
georgijchernyshev@georgijchernyshev-Lenovo-G580:~/CLionProjects/CourseWork
$

Заключение

ходе работы над поставленным заданием удалось разработать программу с интерфейсом командной строки, способную считывать, обрабатывать и сохранять изображения формата PNG. Программа обладает следующим функционалом:

-Генерация круга, его заливка

-Копиравоние заданной области в другую

-Отображение облати изображения относительно координатных осей.

Программа была успешно протестирована на работоспособность.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Список литературы

Кенриган Б. И Ритчи Д. Язык программирования Си М.: Вильямс, 1978 288 с.

Standart C++ Library reference // cplusplus.com. URL: http://www.cplusplus.com/reference/ (дата обращения: 20.06.2020).

libpng Home Page // libpng.org. URL: http://www.libpng.org/pub/png/libpng.html (дата обращения: 20.06.2020).

PNG — not GIF! // Хабр. URL: https://habr.com/ru/post/130472/ (дата обращения: 20.06.2020).

getoptlong(3): Parse options // Linux man page. URL: https://linux.die.net/man/3/getoptlong (дата обращения: 20.06.2020).

Разбор опций командной строки в UNIX-подобных системах // Хабр. URL: https://habr.com/ru/post/55665/ (дата обращения: 20.06.2020).

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРИВЕДЕН В РЕПОЗИТОРИИ НА GITGUB.