**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Шадринский государственный педагогический университет»

Факультет информатики математики и физики

Кафедра программирования и автоматизации бизнес-процессов

«**Разработка на языке программирования Си утилиты обработки изображения в формате PPM P6**»

Курсовая работа по дисциплине «Проектный практикум»

направление подготовки 09.03.01 Информатика и вычислительная техника

образовательная программа « Программное обеспечение вычислительной техники и автоматизированных систем»

Квалификация - бакалавр

**Выполнил**:

Пономарев М.С.

Студент 330Б группы

очная форма обучения

**Научный Руководитель**:

Канд. Пед. Наук, доцент

Слинкин Д.А

“\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Содержание**

**Введение…………………………………………………………………..3**

1. **Теоретическая часть**
   1. **Описание Формата PPM P6……………………………..…….…4**
   2. **Функции обработки изображения и методы реализации……7**

**Введение**

Редактор - это программа, предназначенная для изменения и обработки различных типов файлов или контента. Они стали неотъемлемой частью многих сфер, где требуется работа с текстом, графикой, аудио- и видеоматериалами. В данной курсовой работе рассматривается редактор изображений в формате PPM P6.

**Актуальность:** обработка изображений становится всё более востребованной. С увеличением объема графической информации возрастает потребность в программах, которые способны эффективно её обработать. Утилита на языке программирования СИ может быть полезной как для опытных разработчиков, так и для студентов.

**Цель:** Разработать утилиту для обработки изображения, которое будет иметь следующие функции: поворот изображения, масштабирование черно-белый фильтр, кадрирование.

**Задачи:**

1. Изучить спецификации формата изображений PPM P6.
2. Определить требуемые функции для редактирования изображения.
3. Реализовать рабочую программу на языке СИ со всеми требуемыми функциями редактирования изображений.
4. Протестировать и проверить работу всех функций.
5. **Теоретическая часть**
   1. **Формат PPM P6. Общая характеристика**

Формат PPM (Portable Pixmap) представляет собой простой и широко поддерживаемый формат для хранения растровых изображений, который позволяет осуществлять быструю и эффективную обработку изображений без потерь качества. Файлы формата PPM делятся на два типа: P3 (ASCII-версия, текстовый формат) и P6 (бинарный формат). Оба этих формата содержат один и тот же набор информации но различаются способом представления этих данных.

P3 (текстовый) формат представляет данные каждого пикселя в виде ASCII-чисел, что занимает больше места и обрабатывается дольше из-за необходимости преобразования текста в двоичные данные в процессе чтения файла. P6 (бинарный) формат представляет цвет каждого пикселя в виде последовательности бинарных чисел, что уменьшает размер файла и ускоряет чтение и запись данных за счёт отсутствия промежуточных преобразований.

Благодаря простой структуре формата PPM файлы могут быть легко созданы и обработаны на любой платформе с помощью стандартных средств программирования, что делает их удобными для транспортировки между различными системами и программным обеспечением.

Также этот формат используется в академических и обучающих целях для изучения обработки изображений.

В отличие от современных форматов, которые используют сложное сжатие данных для уменьшения размера файла, PPM может привести к созданию относительно больших файлов при высоком разрешении изображения.

В целом, формат PPM P6 выступает в качестве идеального кандидата для операций, где время обработки и прямой доступ к данным изображения являются приоритетными, а размер файла и поддержка сжатия не являются основными ограничениями.

Файл в формате PPM P6 состоит из нескольких основных сегментов:

Магическое число: Первая строка файла начинается с символов P6, указывающих на то, что это файл формата PPM, версии P6.

Размеры изображения: Следующая строка указывает размеры изображения, сначала ширину в пикселях, затем высоту, разделенные пробелом.

Максимальное значение цвета: Третья строка определяет максимальное значение для компонентов цвета, обычно это значение равно 255. Эта величина позволяет определить, как интерпретировать последующие данные о цвете.

Данные о пикселях: После заголовка следуют данные о пикселях, начиная с верхнего левого угла. Каждый пиксель представлен тремя байтами для каналов RGB.

**Плюсы формата PPM.**

* **Простота формата:** Одним из главных преимуществ формата PPM является его простота. Это дает возможность легко понять и реализовать поддержку формата как в новом, так и в существующем программном обеспечении.
* **Быстрая обработка данных:** Благодаря отсутствию сжатия, работы с данными в формате PPM производятся быстро, поскольку не требуется время на декомпрессию и повторное сжатие при обработке.
* **Универсальность:** Файлы PPM могут быть созданы и обработаны на любой платформе, снимая ограничения на использование в различных операционных системах и обеспечивая обмен данными между разнообразным программным обеспечением.
* **Прямой доступ к пикселям:** Формат позволяет легко получить доступ к каждому пикселю изображения для анализа или модификации, что делает его полезным для обучения и академических исследований по обработке и анализу изображений.
* **Поддержка высокого разрешения:** PPM поддерживает изображения с высоким разрешением без введения дополнительной сложности в структуру файла.

**Минусы формата PPM.**

* **Размер файла:** Важным недостатком PPM является его тенденция к созданию больших файлов из-за отсутствия сжатия. Это может ограничивать его использование для очень больших изображений или при работе с ограниченным дисковым пространством.
* **Отсутствие поддержки прозрачности:** PPM не поддерживает прозрачность изображений, что ограничивает его применение в некоторых графических задачах, где требуется альфа-канал.
* **Ограниченная поддержка:** Хотя формат является универсальным, он не всегда непосредственно поддерживается в популярном графическом и редакторском ПО без установки дополнительных плагинов или конвертации в другие форматы.
* Безопасность: Простота формата делает его потенциально уязвимым для подмены данных или выполнения вредоносного кода, если процесс обработки не соответствует определенным мерам предосторожности.
  1. **Функции обработки изображения и методы реализации.**

**Кадрирование.**

Выделение интересующей области изображения и удаление ненужных краев.

Пользователь или программа определяет координаты (x, y) левого верхнего и правого нижнего углов области, которую необходимо сохранить. После этого из изображения удаляются все пиксели за пределами указанной области.

**Масштабирование.**

Изменение размера изображения без потери важных деталей или искажения пропорций.

Методы масштабирования:

-Ближайший сосед. Это самый простой и быстрый метод, который выбирает значение ближайшего пикселя к данной точке и применяет это значение для нового пикселя.

-Билинейная интерполяция. Учитывает ближайшие 2x2 пикселя и на основании их создает новый пиксель, линейно интерполируя значения по обоим осям. Этот метод лучше сохраняет плавность переходов между цветами, чем ближайший сосед, и подходит для умеренного масштабирования.

-Бикубическая интерполяция. Еще более продвинутый метод, который использует 16 ближайших пикселей для создания нового, интерполируя значения в обеих осях кубическим способом. Бикубическая интерполяция обеспечивает еще более гладкие градиенты и переходы между цветами, но требует больше вычислительных ресурсов, чем билинейная интерполяция.

**Поворот на 90 градусов.**

Изменение ориентации изображения на 90 градусов.

Простая операция реализуемая перестановкой пикселей.

**Черно-белый фильтр.**

Преобразование изображения в градации серого, что позволяет сосредоточить внимание на текстуре и форме объектов.

-Простое усреднение. Метод вычисляет среднее значение трех цветовых компонентов для каждого пикселя и присваивает это среднее значение каждому из цветовых каналов, делая пиксель оттенком серого.

-Линейное взвешенное преобразование. Принимает во внимание разные вклады цветов к общему восприятию яркости человеческим глазом. Этот метод ближе к тому, как человеческий глаз видит цвет.

-Метод оттенков. Сочетает в себе методы усреднения и использования яркостных коэффициентов. Он вычисляет разность между самым высоким и самым низким значениями из трех цветов и делит эту разность на 2, чтобы добавить её к усреднённому значению.

-Палитра максимального декодирования. При этом алгоритме для каждого пикселя используется только самая высокая (максимальное декодирование) или самая низкая (минимальное декодирование) из трех цветовых компонентов, что создает более контрастные черно-белые изображения.

-Одноканальный метод. Можно выбрать один цветной канал (красный, зеленый или синий) и использовать его значения для создания черно-белого изображения. Этот метод подчеркивает информацию в выбранном канале, но теряет информацию в других.

**Эффект размытия**

Преобразование изображения в менее детализированное.

-Простое размытие

Использует простую форму ядра, где все значения равны между собой. Процесс включает замену каждого пикселя усредненным значением его соседей в определенном радиусе.

-Гауссово размытие Применяет ядро, основанное на функции Гаусса, для достижения более мягкого и естественного эффекта размытия. Центральные пиксели имеют больший вес по сравнению с крайними, что обеспечивает плавный переход.

-Двунаправленное размытие

Учитывает как пространственные (расстояние между пикселями), так и радиометрические (различие в интенсивности) различия, сохраняя при этом острые границы. Используется для шумоподавления, сохраняя детали краев.

-Срединное размытие

Заменяет каждый пиксель медианным значением из его окрестностей. Эффективно удаляет шум типа "соль и перец", сохраняя границы объектов.

**Эффект негатива**

Эффект при котором цвета изображения инвертируются.

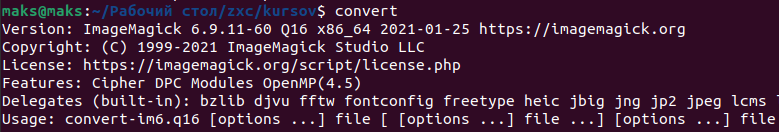
-Поэлементное инвертирование:

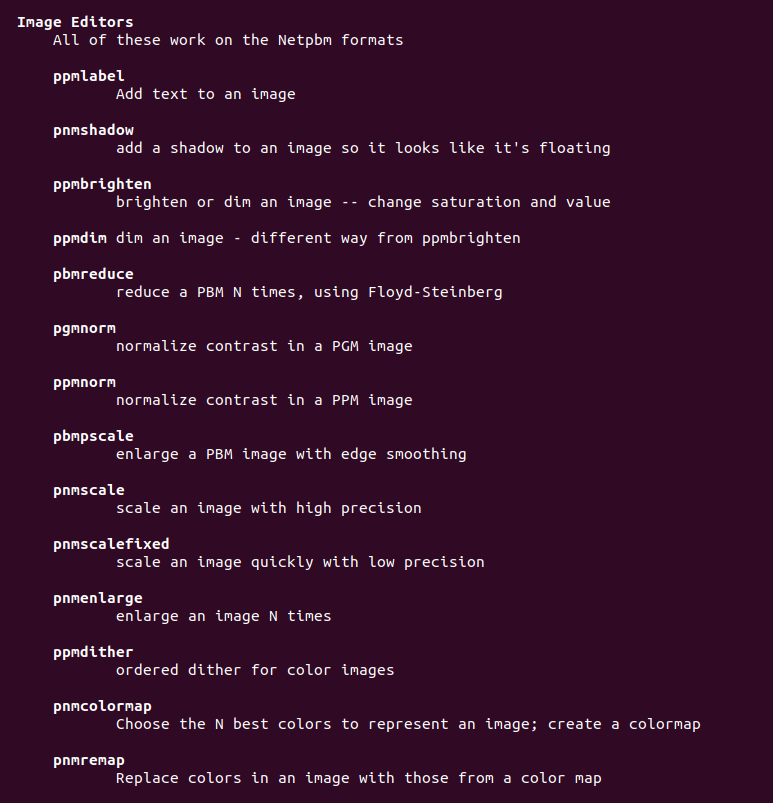
Этот подход подразумевает обработку отдельных пикселей изображения, где новое значение каждого цветового компонента пикселя (красного, зеленого, и синего) высчитывается как 255 минус текущее значение этого компонента.

1. **Реализация утилиты обработки изображения.**
   1. **Существующие программы.**

**ImageMagick**

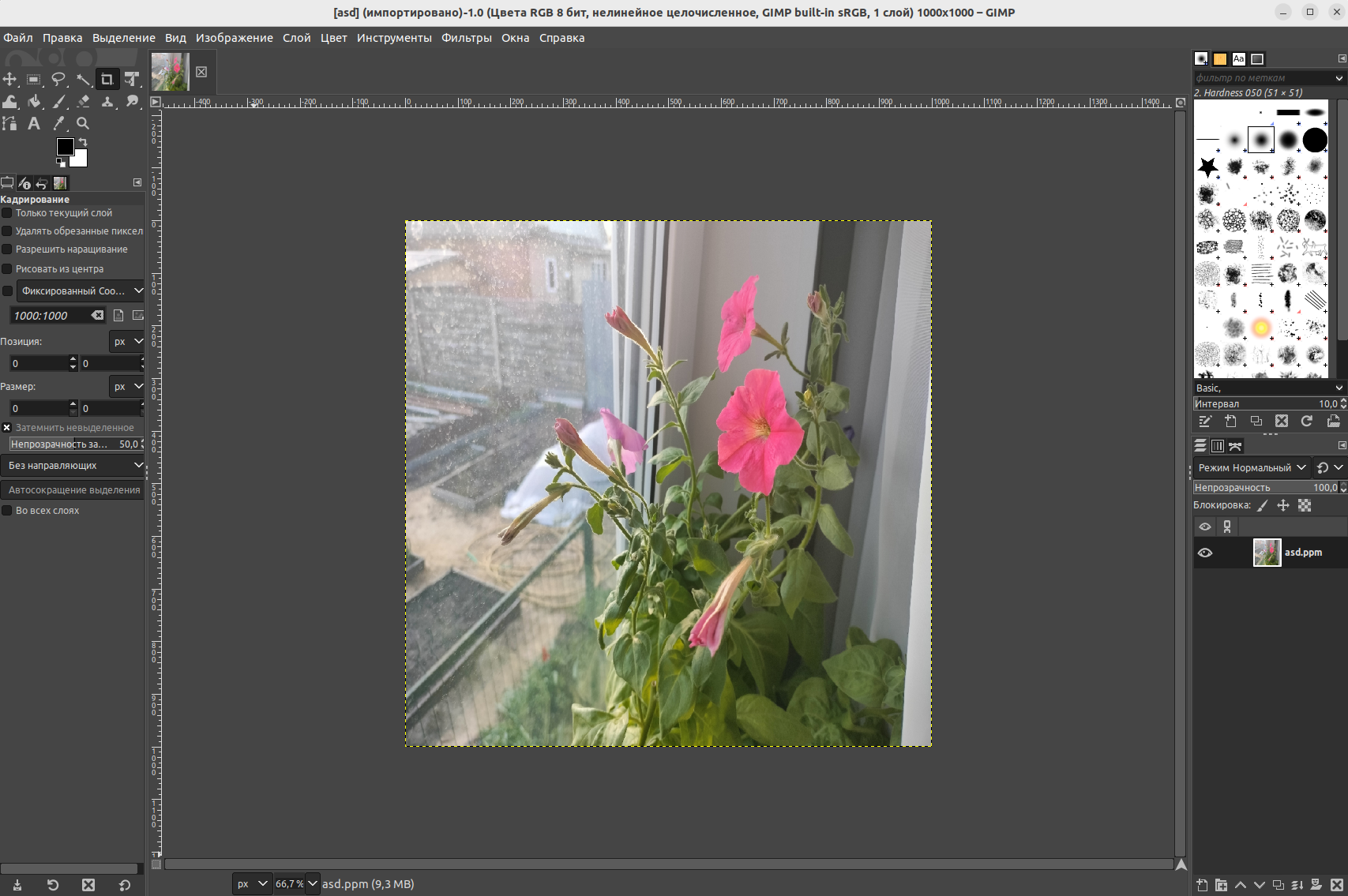
ImageMagick предоставляет широкий спектр возможностей для обработки изображений и поддерживает множество графических форматов. Она может быть использована для масштабирования, преобразования, наложения текста, вращения и многих других операций с помощью командной строки.

**Netpbm**

Netpbm – это набор простых утилит командной строки, которые могут использоваться для обработки изображений. Они могут конвертировать между различными графическими форматами, а также выполнять простые операции обработки, такие как масштабирование, поворот и изменение цветов.

**GIMP**

GIMP – это свободно распространяемый инструмент для таких задач, как ретуширование фотографий, композиция изображений и создание изображений. GIMP поддерживает автоматизацию с использованием сценариев на Scheme, Python и других скриптовых языках.

****

* 1. **Этапы реализации утилиты**

1. Считывание и загрузка изображения.

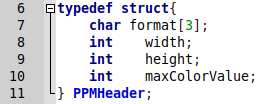
2. Применение функций обработки.

3. Сохранение изображения.

4. Взаимодействие пользователя с утилитой.

* 1. **Считывание и загрузка изображения**

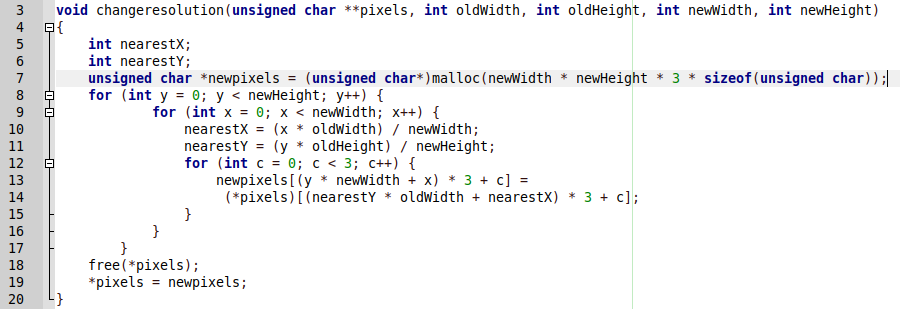
С помощью функции readimage считываются метаданные, которые записываются в структуру PPMHeader, и пиксели изображения.

****

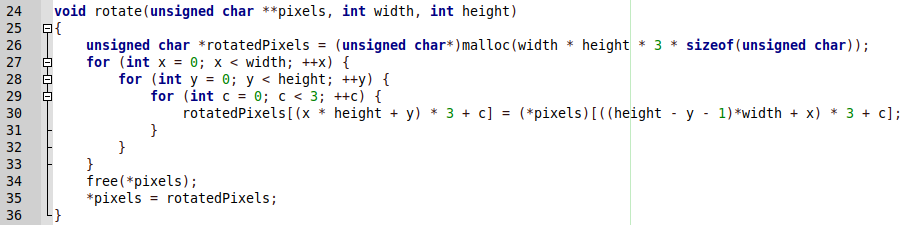
****

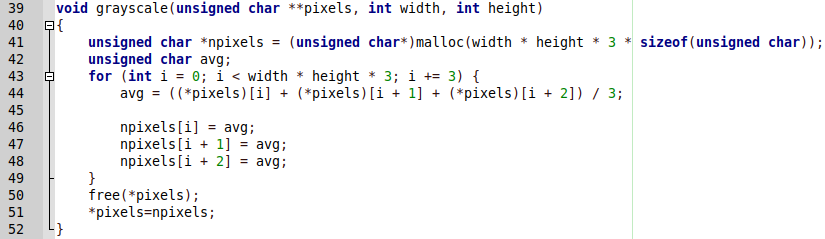
* 1. **Применение функций обработки.**

**Фунция сhangeresolution**

Изменяет разрешение изображения с использованием метода ближайшего соседа.

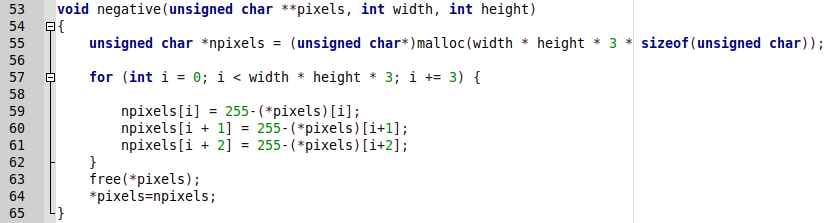
**Функция rotate**

Функция поворачивает изображение на 90 градусов по часовой стрелке.

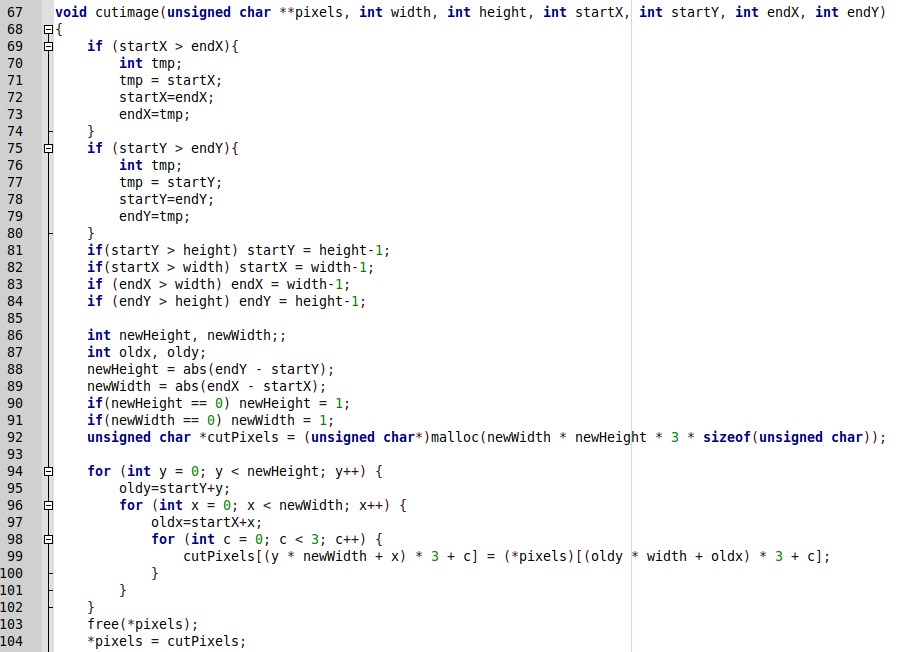
**Функция grayscale**

Функция преобразует цветное изображение в оттенки серого. Для каждого пикселя вычисляет среднее значение его цветовых компонент (RGB), заменяя ими исходные значения.

**Фунцкия negative**

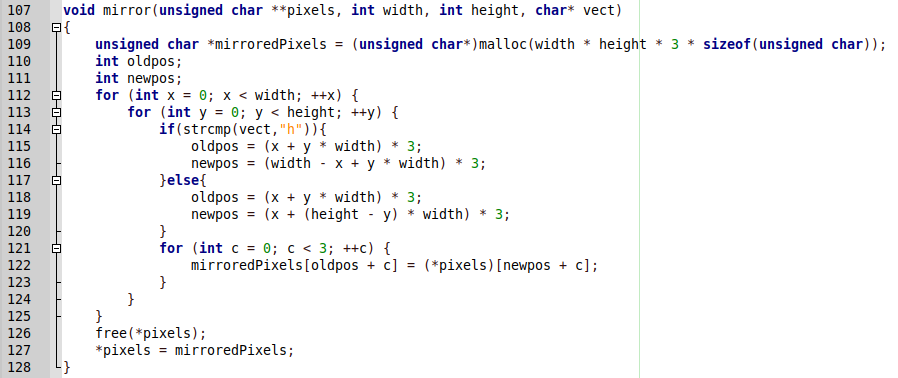


Эта функция создает негатив изображения. Для каждого пикселя вычисляет его негатив, вычитая его значение из 255.

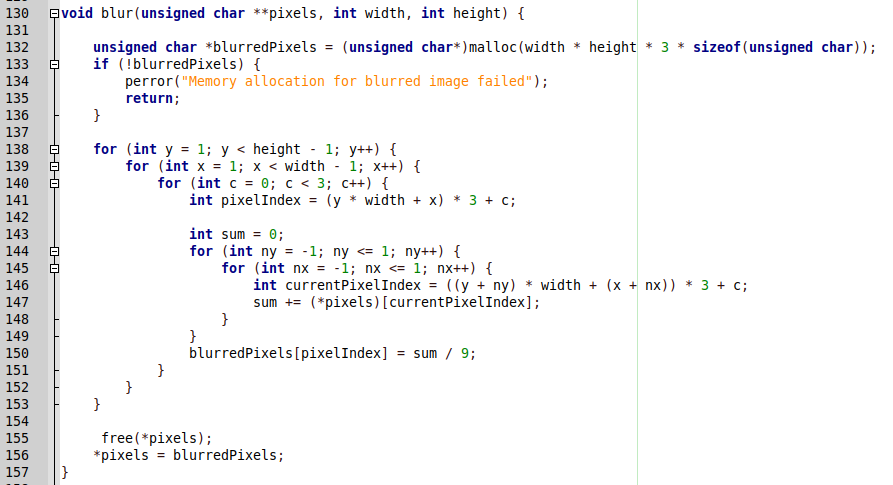
**Функция cutimmage**

Эта функция вырезает участок из изображения. Она принимает координаты начальной и конечной точек вырезаемого участка, пересчитывает координаты и копирует пиксели из старого массива в новый.

**Функция mirror**

Эта функция зеркально отражает изображение. В зависимости от переданного параметра (vect), она может отразить изображение по горизонтали или вертикали.

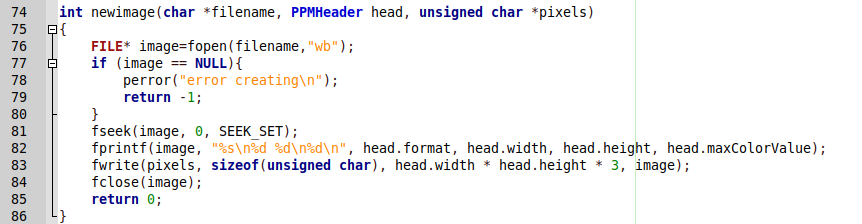
**Функция blur**

****

Эта функция применяет размытие к изображению с использованием фильтра среднего значения.

* 1. **Сохранение изображения**

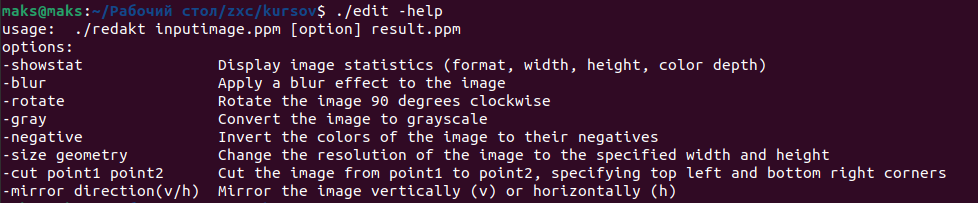
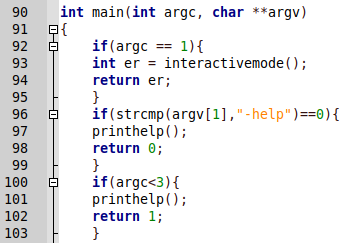
После применения функций обработки, изображение сохраняется при помощи функции newimage, которая принимает название файла, обновленный заголовок файла и обновленный массив пикселей

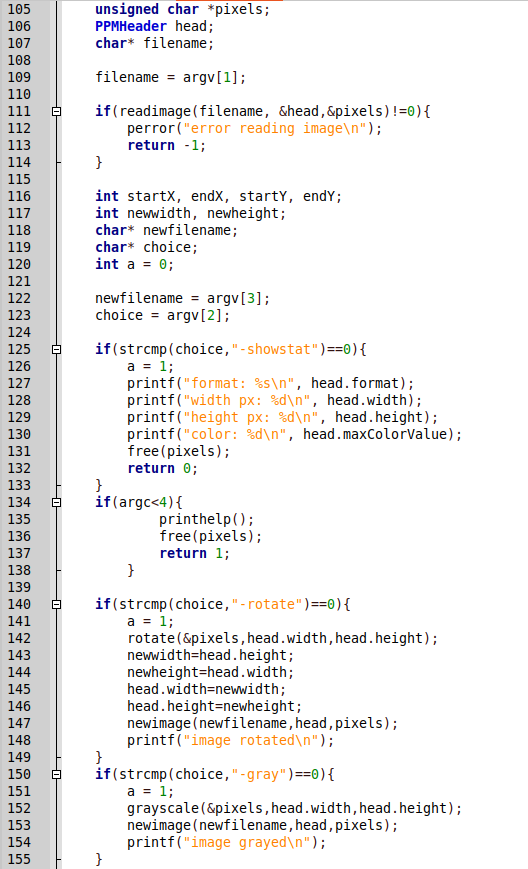
* 1. **Взаимодействие пользователя с утилитой**

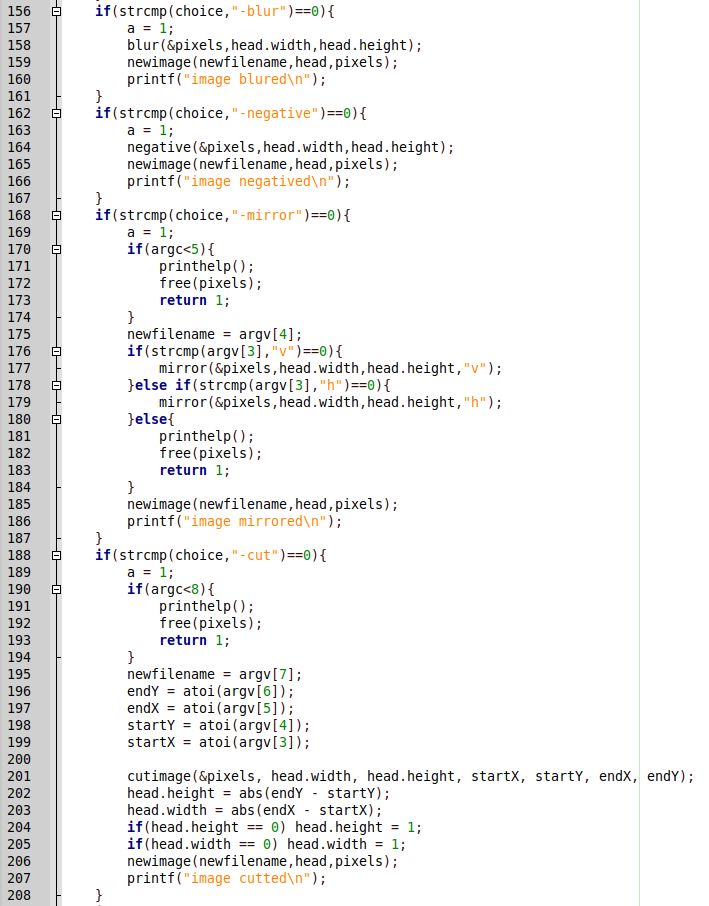
Для взаимодействия с утилитой предусмотрено два способа.

1. Взаимодействие через командную строку.

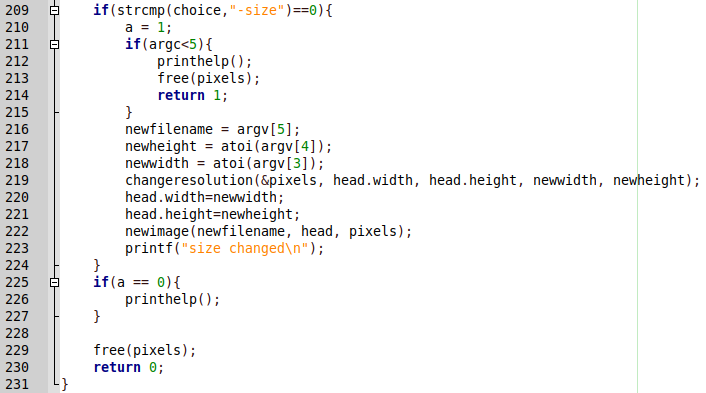
Для взаимодействия через командную строку пользователю надо ввести название картинки, которую он хочет редактировать, опцию и название результирующего файла.

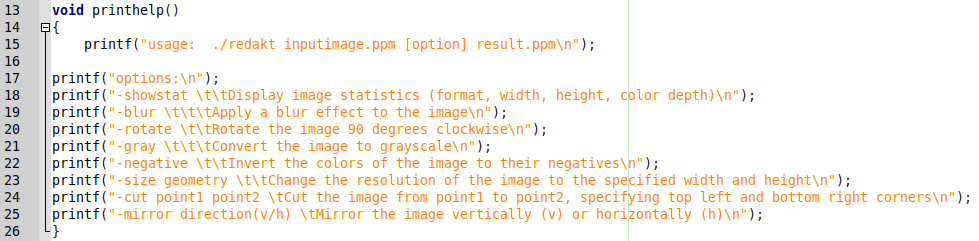


Переменная «a» нужна, чтобы определять сработала какая либо из опций или нет.

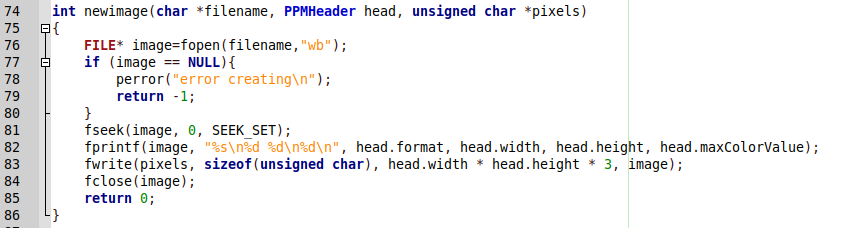


Функция atoi в строках 196-199 преобразуют строковые значения в int.

Если ни одна из функций не сработала и переменная «a» осталась равна нулю срабатывает функция printhelp.

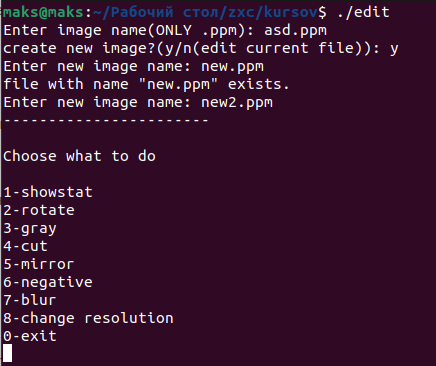
Также эта функция выводится в случае ввода неверных аргументов или если ввести опцию -help.

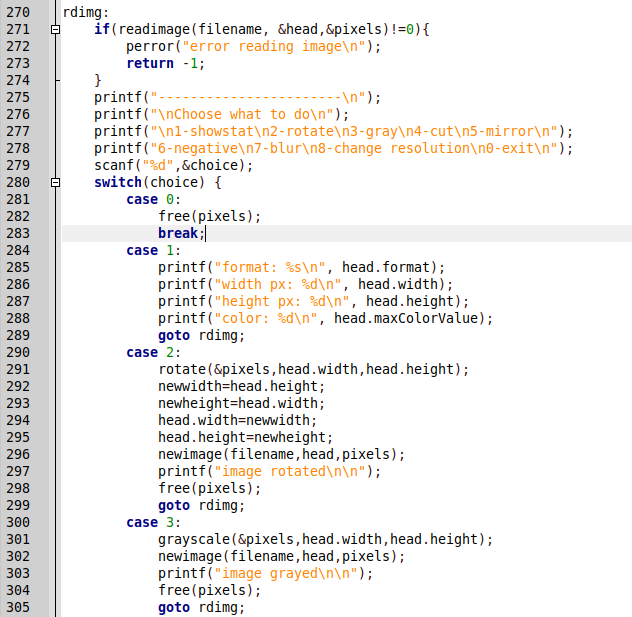
Функция newimage перезаписывает или создает новое изображение с обновленными данными.

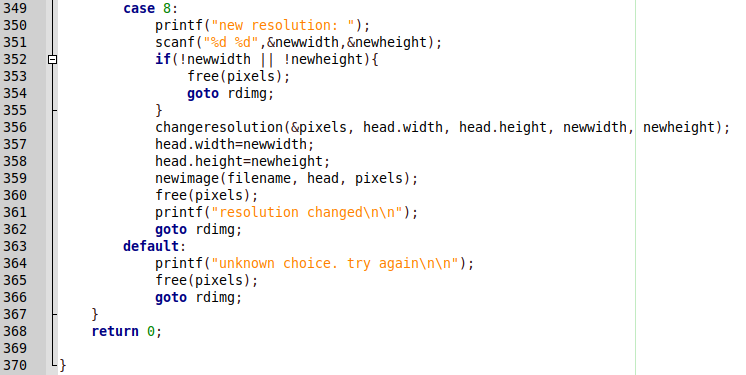
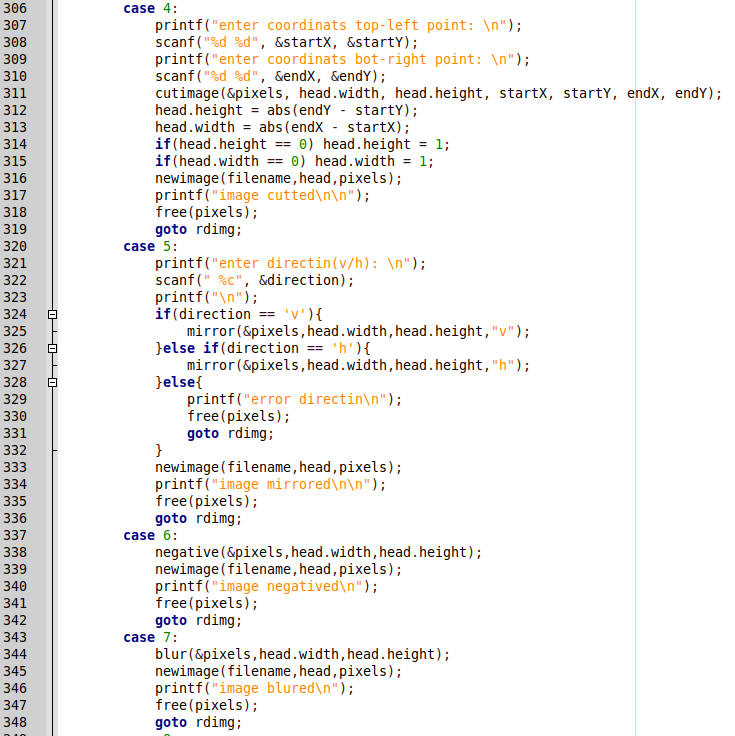


1. Взаимодействие через интерактивное меню.

В случае если программа запускается без агрументов, то утилита запустит интерактивное меню.

Все функции обработки изображения и функции чтения и создания изображений идентичны прошлому режиму. Отличие лишь во взаимодействии утилиты и пользователя.

Точка rdimg позволяет программе вернуться на момент чтения изображения после редактирования либо после какой-либо ошибки. Повторно будет считываться тот файл, в котором уже произошли необходимые



**https://netpbm.sourceforge.net/doc/ppm.html**

**https://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/cips2ed.pdf**

**https://imagemagick.org/**

**https://netpbm.sourceforge.net/**

[**https://www.gimp.org/**](https://www.gimp.org/)