Лабораторная работа №5. Разработка функций на языке Ассемблера и встраивание их в программы на языке C/C++

15 мая 2021 г.

Целью данной работы является изучение взаимодействия кода на языке C и на языке Ассемблера. Данный аспект является крайне важным, т.к. обычно программы стараются делать максимально переносимыми, при этом затрачивая на разработку минимальное время.

При разработке ПО хорошей практикой часто является следующий подход (разумеется, данные этапы могут многократно повторяться):

1. Проектирование программы.
2. Разработка программы.
3. Отладка программы.
4. Анализ производительности программы.
5. Оптимизация программы.

Подобный подход можно коротко описать с помощью следующего высказывания Кента Бека: «Make it work, make it right, make it fast».

При этом, практика показывает, что смысла разрабатывать программы на языке Ассемблера целиком практически нет, гораздо более эффективной стратегией является реализация на языке Ассемблера лишь тех фрагментов программы, на выполнение которых тратится наибольшее время.

Для выполнения данной лабораторной работы необходимо:

1. Разработать программу на языке `C/C++`, реализующую требуемую функциональность.
2. Провести анализ производительности программы, собранной со следующими настройками оптимизатора (режим отладки должен быть выключен):
   - `-O0`;
   - `-O3`;
   - `-Ofast`.
3. Реализовать основную функцию обработки изображения на языке Ассемблера.
4. Провести анализ производительности полученной программы.

Примечания:

1. Имена входного и выходного файлов должны передаваться в программу в виде аргументов командной строки.
2. Программа должна проверять наличие и корректность входных файлов, корректность задания исходных данных, и т.д.
3. Программа на языке C должна содержать функцию, выполняющую основную обработку изображения, которое находится в некотором буфере.
4. Функция, выполняющая основную обработку изображения, не может использовать никакие библиотечные функции.
5. Реализация ассемблерной функции, выполняющей основную обработку изображения, должна находиться в отдельном файле с расширением `.s`.
6. При работе с файлами (чтение/запись), содержащими графическое изображение, рекомендуется использовать готовые библиотеки на языках `C/C++`.
7. Анализ производительности должен осуществляться на нескольких примерах, его результаты должны быть представлены в виде таблицы.

Задание 1. Представить цветное изображение в оттенках серого, воспользовавшись алгоритмом «Averaging»:

$$Gray = \frac{Red + Green + Blue}{3}$$

Задание 2. Представить цветное изображение в оттенках серого, воспользовавшись алгоритмом «Luminance»:

$$Gray = Red * 0.3 + Green * 0.59 + Blue * 0.11$$

Или:

$$Gray = (Red * 0.299 + Green * 0.587 + Blue * 0.114)$$

Или:

$$Gray = (Red * 0.2126 + Green * 0.7152 + Blue * 0.0722)$$

Задание 3. Представить цветное изображение в оттенках серого, воспользовавшись алгоритмом «Desaturation»:

$$Gray = \frac{Max(Red, Green, Blue) + Min(Red, Green, Blue)}{2}$$

Задание 4. Представить цветное изображение в оттенках серого, воспользовавшись алгоритмом «Decomposition»:

$$Gray = Max(Red, Green, Blue)$$

Или:

$$Gray = Min(Red, Green, Blue)$$

Задание 5. Представить цветное изображение в оттенках серого, воспользовавшись алгоритмом «Single color channel»:

$$Gray = Red$$

Или:

$$Gray = Green$$

Или:

$$Gray = Blue$$

Задание 6. Представить цветное изображение в оттенках серого, воспользовашись алгоритмом, позволяющим указать количество оттенков серого:

$$ConversionFactor = \frac{255}{(NumberOfShades - 1)}$$

$$AverageValue = \frac{(Red + Green + Blue)}{3}$$

$$Gray = Integer(\frac{AverageValue}{ConversionFactor} + 0.5) * ConversionFactor$$

Задание 7. Инвертировать цвета в изображении.

Задание 8. Обрезать изображение по границе прямоугольной области, заданной координатами двух её углов.

Задание 9. Отмасштабировать изображение до заданного размера, воспользовашись алгоритмом «Nearest Neighbor». Для этого необходимо определить коэффициенты масштабирования по осям x и y, а затем каждому пикселю нового изображения поставить в соответствие пиксель из старого:

$$x\_ratio = \frac{width\_old}{width\_new}$$

$$y\_ratio = \frac{height\_old}{height\_new}$$

$$(x, y) \iff (floor(x * x\_ratio), floor(y * y\_ratio))$$

Задание 10. Отразить изображение по горизонтали (вокруг вертикальной оси, проходящей через центр изображения).

Задание 11. Отразить изображение по вертикали (вокруг горизонтальной оси, проходящей через центр изображения).

Задание 12. Повернуть изображение на заданный угол (вокруг центра). Для этого каждому пикселю нового изображения необходимо поставить в соответствие пиксель из старого:

$$(x, y) \iff (x * cos\alpha - y * sin\alpha, x * sin\alpha + y * cos\alpha)$$

Задание 13. Реализовать размытие изображения по Гауссу. Обработка граничных условий должна осуществляться методом дополнения, т.е. для применения фильтра необходимо сгенерировать временное изображение большего размера, скопировав крайние пиксели (это необходимо, чтобы у крайних пикселей оригинального изображения появились «соседи»).

Матрица свёртки:

$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

Задание 14. Реализовать размытие изображения по Гауссу. К граничным пикселям (тем, у которых нет «соседей») фильтр применяться не должен.

Матрица свёртки:

$$\frac{1}{256}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\ 4 & 16 & 24 & 16 & 4\\ 6 & 24 & 36 & 24 & 6\\ 4 & 16 & 24 & 16 & 4\\ 1 & 4 & 6 & 4 & 1\end{bmatrix}$$

Задание 15. Реализовать выделение границ. Обработка граничных условий должна осуществляться методом дополнения, т.е. для применения фильтра необходимо сгенерировать временное изображение большего размера, скопировав крайние пиксели (это необходимо, чтобы у крайних пикселей оригинального изображения появились «соседи»).

Матрица свёртки:

$$\begin{bmatrix}-1 & -1 & -1\\ -1 & 8 & -1\\ -1 & -1 & -1\end{bmatrix}$$

Номер варианта	Номер задачи	Формат изображения
1	1	JPEG
2	2	JPEG
3	3	JPEG
4	4	JPEG
5	5	JPEG
6	6	JPEG
7	7	JPEG
8	8	JPEG
9	9	JPEG
10	10	JPEG
11	11	JPEG
12	12	JPEG
13	13	JPEG
14	14	JPEG
15	15	JPEG
16	1	PNG
17	2	PNG
18	3	PNG
19	4	PNG
20	5	PNG
21	6	PNG

22	7	PNG
23	8	PNG
24	9	PNG
25	10	PNG
26	11	PNG
27	12	PNG
28	13	PNG
29	14	PNG
30	15	PNG
31	1	BMP
32	2	BMP
33	3	BMP
34	4	BMP
35	5	BMP
36	6	BMP
37	7	BMP
38	8	BMP
39	9	BMP
40	10	BMP
41	11	BMP
42	12	BMP
43	13	BMP
44	14	BMP
45	15	BMP