

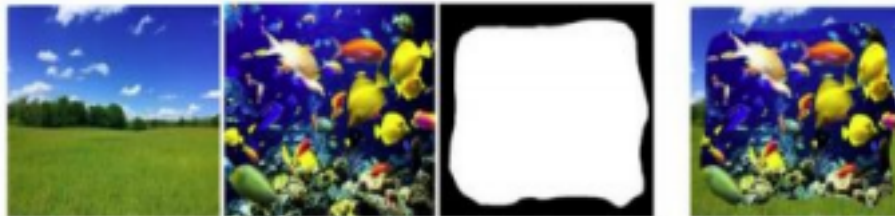
Enmascarar con SIMD

Normativa de entrega:

- Fecha de entrega: 01/11.
- Grupos de 2 integrantes.
- Se entregará un link a un repo git que contenga todo el código fuente.
- Entregar todo el código C y ASM solicitado.
- Realizar un informe en el archivo README.md Markdown en git.
- Mostrar mediante gráficos los tiempos de respuesta para las pruebas realizadas en cada contexto.
- Además de esto se deberá entregar un archivo bash .sh que compile y ejecute las pruebas que se entregan en el informe.

Recomendaciones:

- Utilizar distintos tamaños de imágenes para las pruebas.
- Es aconsejable utilizar herramientas para hacer resize a las imágenes.



Con el objetivo de comparar la performance de las instrucciones SIMD con las instrucciones aritméticas y lógicas de la ALU, se pide desarrollar una aplicación de línea de comandos que combine dos imágenes usando una máscara de selección. Implementaremos dos versiones de esta operación: primero en lenguaje C y luego utilizando instrucciones SIMD en lenguaje ensamblador.

Para independizarnos del formato de las imágenes, usaremos **GraphicsMagick** (<http://www.graphicsmagick.org/>) para convertir las imágenes a procesar en archivos rgb. Una imagen rgb es un archivo binario que almacena los valores rojo, verde y azul de cada píxel usando tres bytes consecutivos.

Trabajo Práctico 2 - Paralelismo
Organización del Computador II - UNGS
2C 2021

Implementación

La aplicación debe recibir los siguientes parámetros: `img1.rgb` , `img2.rgb`, `máscara.rgb`, ancho y alto. Las dos imágenes de entrada y la máscara deben tener las mismas dimensiones. La máscara solo debe contener los colores blanco (FFFFFF) y negro (000000) de manera que los pixels negros seleccionen el pixel correspondiente de `img1`, y los pixels blancos de la máscara seleccionen los de `img2`.

Por un lado, se debe implementar la función **`enmascarar_asm`** en lenguaje ensamblador de 32 bits usando instrucciones SIMD y respetando la convención de C de pasaje de parámetros. Al igual que **`enmascarar_c`** en el lenguaje C. Ambas funciones deben pisar el contenido del buffer `a`, con el resultado de combinar `a` y `b` usando la máscara `mask`.

Además se debe implementar un programa en C que reciba los parámetros mencionados por línea de comandos y que llame a las dos funciones para generar dos archivos (`salida_c.rgb` y `salida_asm.rgb`) que corresponden a las imágenes producidas por las dos llamadas.

- A. Función **`enmascarar_c (unsigned char *a, unsigned char *b, unsigned char *mask, int cant);`**
- B. Función **`enmascarar_asm (unsigned char *a, unsigned char *b, unsigned char *mask, int cant);`**