

Reto: Procesamiento de Imágenes con Buddy System

Contexto

El procesamiento de imágenes es una tarea que consume grandes cantidades de memoria y procesamiento debido al tamaño y la complejidad de los datos. Un ingeniero de sistemas debe desarrollar soluciones eficientes para el manejo de memoria y optimización de procesos mediante estructuras avanzadas de asignación de memoria.

En este reto, se solicita implementar un programa en **C++** que permita realizar operaciones básicas de procesamiento de imágenes (rotación y escalado) y que, además, gestione la memoria mediante el [Buddy System](#) para comparar su eficiencia con el uso convencional de `new/delete`.

Objetivo

Desarrollar un programa en C++ que permita:

1. Rotar una imagen en su centro en cualquier ángulo seleccionado por el usuario.
 2. Escalar una imagen a un tamaño definido por el usuario.
 3. Implementar una gestión de memoria mediante **Buddy System** y comparar su rendimiento frente a la asignación dinámica convencional (`new/delete`).
 4. Evaluar el consumo de memoria y el tiempo de procesamiento en ambos modos.
 5. Generar conclusiones sobre la eficiencia de la gestión de memoria y el impacto en el procesamiento de datos.
-

Descripción Técnica

Parte 1: Carga de la Imagen

- El programa debe permitir la carga de una imagen desde un archivo mediante línea de comandos.
- La imagen debe ser representada como una matriz tridimensional `pixeles[alto][ancho][canales]` donde:

- `alto` = número de filas (altura de la imagen)
 - `ancho` = número de columnas (anchura de la imagen)
 - `canales` = número de canales de color (3 para RGB o 4 para RGBA)
 - Mostrar información básica de la imagen cargada (dimensiones, canales).
-

Parte 2: Rotación de una Imagen en el Centro

- Permitir al usuario ingresar el ángulo de rotación.
 - Implementar un algoritmo de rotación utilizando interpolación bilineal para preservar la calidad de la imagen.
 - La rotación debe realizarse sobre el centro de la imagen.
 - Si la imagen rotada genera espacios vacíos, deben rellenarse mediante un valor constante (por ejemplo, color negro o blanco).
-

Parte 3: Escalado de una Imagen

- Permitir al usuario ingresar el factor de escalado (mayor o menor que 1.0).
 - Utilizar interpolación bilineal para redimensionar la imagen.
 - Mantener la proporción de la imagen durante el escalado.
 - Mostrar información sobre el nuevo tamaño de la imagen tras la operación.
-

Parte 4: Gestión de Memoria con Buddy System

- Implementar un sistema de asignación de memoria mediante **Buddy System**.
 - Si el usuario selecciona el modo Buddy System, la imagen debe ser almacenada en bloques administrados por el Buddy System.
 - Si el usuario selecciona el modo convencional, la imagen debe ser almacenada mediante `new/delete`.
 - Comparar la eficiencia de asignación de memoria en ambos casos.
-

Parte 5: Comparación de Rendimiento y Consumo de Memoria

- Medir el tiempo de procesamiento para las operaciones de rotación y escalado utilizando `std::chrono`.
- Medir el consumo de memoria utilizando `mallinfo()` o `getrusage()` para determinar el tamaño de la memoria reservada.

- Mostrar en pantalla una comparación directa entre el rendimiento y el consumo de memoria en los dos modos de asignación.
-

Parte 6: Salida de Datos

El programa debe presentar una salida clara con los siguientes datos:

- Dimensiones originales y finales de la imagen.
 - Ángulo de rotación aplicado.
 - Tiempo de procesamiento en milisegundos.
 - Consumo de memoria en ambos modos (Buddy System y convencional).
 - Diferencia de rendimiento entre ambos modos.
-

Modo de Ejecución

El programa debe ejecutarse desde la línea de comandos con los siguientes parámetros:

bash

CopyEdit

```
./programa_imagen entrada.jpg salida.jpg -angulo 45 -escalar 1.5  
-buddy
```

Parámetros:

- `entrada.jpg`: archivo de imagen de entrada
 - `salida.jpg`: archivo donde se guarda la imagen procesada
 - `-angulo`: define el ángulo de rotación
 - `-escalar`: define el factor de escalado
 - `-buddy`: activa el modo Buddy System (si se omite, se usará el modo convencional)
-

Ejemplo de Salida

bash

CopyEdit

```
=== PROCESAMIENTO DE IMAGEN ===  
Archivo de entrada: entrada.jpg
```

```
Archivo de salida: salida.jpg
Modo de asignación de memoria: Buddy System
-----
Dimensiones originales: 1920 x 1080
Dimensiones finales: 2880 x 1620
Canales: 3 (RGB)
Ángulo de rotación: 45 grados
Factor de escalado: 1.5
-----
[INFO] Imagen rotada correctamente.
[INFO] Imagen escalada correctamente.
-----
TIEMPO DE PROCESAMIENTO:
- Sin Buddy System: 120 ms
- Con Buddy System: 95 ms

MEMORIA UTILIZADA:
- Sin Buddy System: 2.1 MB
- Con Buddy System: 1.8 MB
-----
[INFO] Imagen guardada correctamente en salida.jpg
```

Preguntas de Análisis

1. ¿Qué diferencia observaste en el tiempo de procesamiento entre los dos modos de asignación de memoria?
2. ¿Cuál fue el impacto del tamaño de la imagen en el consumo de memoria y el rendimiento?
3. ¿Por qué el Buddy System es más eficiente o menos eficiente que el uso de `new/delete` en este caso?
4. ¿Cómo podrías optimizar el uso de memoria y tiempo de procesamiento en este programa?
5. ¿Qué implicaciones podría tener esta solución en sistemas con limitaciones de memoria o en dispositivos embebidos?
6. ¿Cómo afectaría el aumento de canales (por ejemplo, de RGB a RGBA) en el rendimiento y consumo de memoria?
7. ¿Qué ventajas y desventajas tiene el Buddy System frente a otras técnicas de gestión de memoria en proyectos de procesamiento de imágenes?

Entrega

- El programa debe entregarse en un archivo comprimido (.zip o .tar.gz) que incluya:
 - Código fuente (.cpp y .h).
 - Archivo Makefile para compilar el programa.
 - Archivo README.md que explique cómo compilar y ejecutar el programa.
 - Un video que muestre la ejecución del programa con los casos de prueba solicitados.
-

Rúbrica de Evaluación

Criterio	Descripción	Puntaje Máximo
Carga y representación de la imagen	El programa carga correctamente la imagen y la muestra en la estructura adecuada.	15%
Rotación de la imagen	La imagen rota correctamente y la interpolación mantiene la calidad.	20%
Escalado de la imagen	La imagen se escala correctamente y mantiene la proporción.	20%
Implementación de Buddy System	La memoria es gestionada correctamente mediante Buddy System.	20%
Comparación de rendimiento y memoria	El programa mide y compara el rendimiento y consumo de memoria correctamente.	15%
Buenas prácticas y estructura del código	El código está organizado, es modular y está bien documentado.	10%

Puntaje Total: 100 puntos

Condiciones para Aprobar

1. El programa debe compilar y ejecutarse correctamente.

2. El programa debe ser capaz de manejar imágenes de diferentes tamaños y formatos.
3. El video debe mostrar claramente la ejecución del programa y la salida de datos.
4. El estudiante debe ser capaz de explicar la implementación y las diferencias entre los dos modos de asignación de memoria.

Se Entrega por Interactiva virtual en el buzón de entrega para el parcial 2