TEC | Tecnológico de Costa Rica

Primer Proyecto

Escuela de Ingeniería en Computadores

Algoritmos y Estructuras de Datos II

Profesor:

Leonardo Araya

Estudiante:

Guillermo Sánchez Cede | 2024132555
Paula Melina Porras Mora | 2023082886

Tabla de contenidos

Introducción

En este proyecto se implementa un sistema de gestión de memoria utilizando comunicación gRPC para ofrecer una interfaz remota que permite la creación, asignación, modificación y eliminación de bloques de memoria. Este sistema gestiona bloques de memoria. Este sistema gestiona bloques de memoria de un tamaño determinado, que además se pueden manipular a través de operaciones básicas como Create, Set, Get, DecreaseRefCount y IncreaseRefCount.

El objetivo principal de este sistema es proporcionar una solución eficiente para la administración dinámica de memoria, asegurando la integridad de los datos meidante mecanismos de sincronización y un manejo de referencias que permite una gestión adecuada de la memoria, evitando así fugas mientras garantiza el uso eficiente de los recursos.

Este sistema es aplicable en entornos que requieren un control preciso de la memoria y un alto rendimiento en las operaciones de asignación y puede ser extendidopara ser utilizado en aplicaciones más complejas que necesitan gestión de recursos en tiempo real.

Descripción del problema

El problema se basa en que la gestión de memoria dinámica es un desafío crítico en el desarrollo de software especialmente en aplicaciones que requieren un control preciso sobre la asignación y liberación de memoria. Lo que implica que es sistemas convencionales, la asignación de memoria descontrolada pueda llevar a problemas como fugas de memoria, fragmentación y errores de acceso que afectan el rendimiento y estabilidad del programa.

Además, en entornos distribuidos donde múltiples clientes puedan solicitar y modificar bloques de memoria simultáneamente, la coordinación eficiente de estos accesos se vuelve aún más compleja, haciendo que este mal manejo de referencias a la memoria puede causar accesos inválidos, corrupción de datos o uso ineficiente de los recursos.

Descripción de la solución

Para abordar los problemas de gestión de memoria dinámica en este proyecto implementa un Memory Manager que administra la memoria de manera centralizada, en lugar de realizar múltiples asignaciones a lo largo de la ejecución del programa, el sistema reserva un único bloque de memoria en el heap al inicio y gestiona su uso internamente.

La comunicación entre clientes y el memory manager se realiza mediante gRPC, permitiendo que los clientes puedan solicitar y modificar bloques de memoria sin acceder directamente a la gestión de memoria del sistema operativo, para ello se implementan las operaciones:

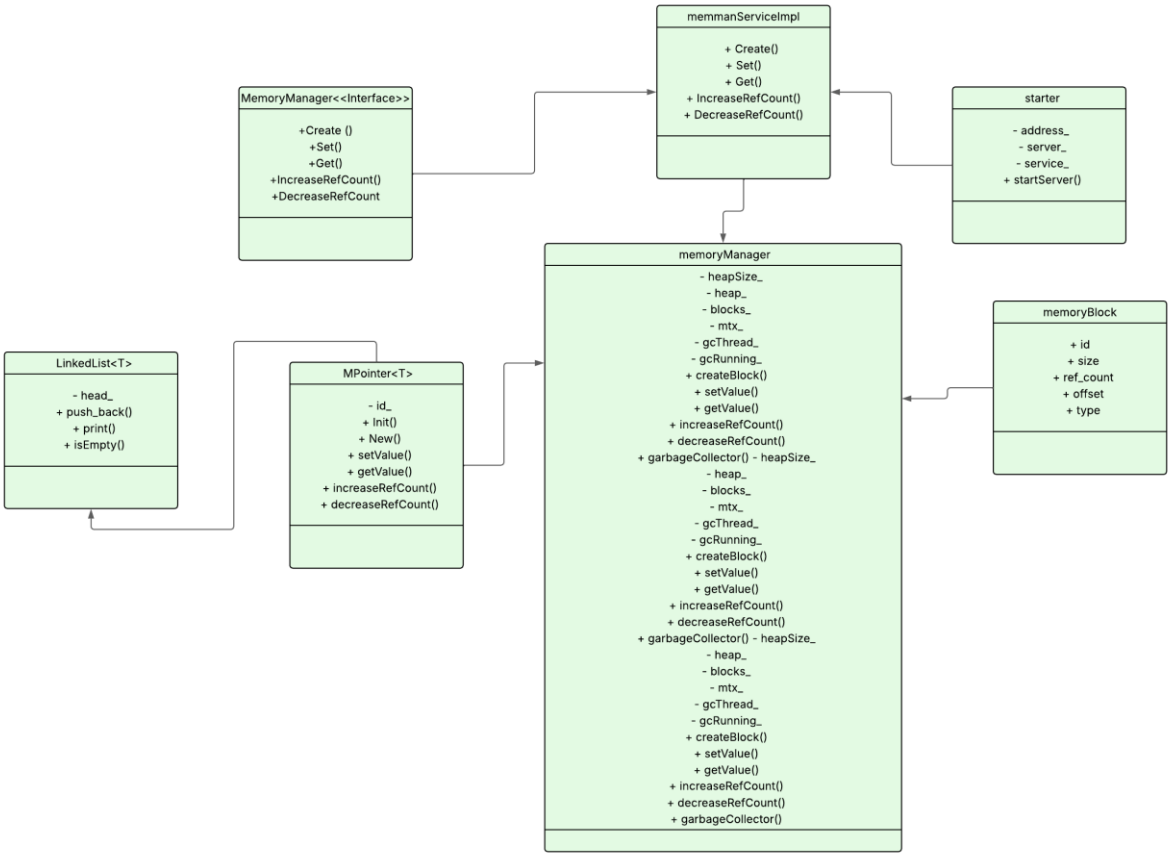
- **Create(size, type):** Reserva un bloque de memoria dentro del espacio administrado.
- **Set(id, value):** Almacena un valor en un bloque de memoria identificado por su ID.
- **Get(id):** Obtiene el valor almacenado en un bloque de memoria.
- **IncreaseRefCount(id) / DecreaseRefCount(id):** Manejan el conteo de referencias para evitar fugas de memoria.

El Memory Manager también cuenta con un **garbage collector**, que monitorea los bloques de memoria y libera aquellos que ya no tienen referencias activas. Para optimizar aún más el uso de los recursos, el sistema incluye un mecanismo de defragmentación, reorganizando la memoria cuando sea necesario para evitar espacios desperdiciados.

Además, se implementa una clase MPointer<T>, que actúa como un puntero inteligente remoto. Esta clase permite a los clientes interactuar con el Memory Manager de manera transparente, sobrecargando operadores como *, & y = para ofrecer una experiencia similar a la de los punteros convencionales, pero con gestión automática de memoria a nivel remoto.

Con este enfoque, el sistema garantiza un uso eficiente de la memoria, evita fugas y fragmentación, y facilita la interacción con la memoria a través de una API sencilla y robusta basada en gRPC.

Diagrama UML



Enlace a GitHub

https://github.com/Mandamientos/MPointers-2.0