

Cómo crear su propia biblioteca malloc

Gestión de memoria en C

Realloc

es conceptualmente

malloc + memcpy + free en el puntero antiguo.

Prueba

Cree el siguiente run.sh

comando deseado
sh run.sh \$(CMD)

Actuación

A la memoria de manera eficiente asignación previa

Para blocks

iferenciamos entre 3 tamaños de bloque

MALL
TINY
y
LARGE

usamos 128 bytes para un tamaño máximo de malloc

Dado que cada malloc tiene que almacenar sus metadatos

, no podremos almacenar todos los 128 bloques.

ada zona de memoria asignada por mmap

la memoria de manera eficiente asignación previa

Estructura de datos

Llamamos heap a

lleno de bloques

C gestiona de forma nativa las variables en la memoria

Las Variables estaticas y globales

A la memoria principal

Asignación dinámica y mmap

mmap

maneja el mapeo de zonas de memoria fisica a direcciones virtuales y devuelve un punterio a su inicio

malloc

El uso de una llamada al sistema cada vez daría lugar a una degradación grave del rendimiento. Malloc mitiga este problema

Se puede pensar como

Un rendimiento al rededor mmap

La biblioteca

malloc

asigna un bloque de memoria y devuelve un punterio a su inicio.

Cuando el programa ya no lo necesita

free()
que libera la memoria.
realloc
aumenta o disminuye el tamaño de un malloc.
puntero y conserva sus datos

El algoritmo malloc

Busca una referencia de heap list si existe

llenar un espacio liberado .

Si el último montón no tiene suficiente espacio disponible, el algoritmo crea uno nuevo heap llamando a mmap

Si no había ningún bloque disponible, agregue el bloque al final del último heap

buscó un espacio disponible y devuelve su dirección

No podemos mover bloques viejo

sus direcciones virtuales no pueden cambiar.

debemos manejar el espacio libre fragmentado entre nuestros bloques.