

自己动手写一款
动态存储分配器

介绍

- 本实验需要用c语言实现一个动态的存储分配器，也就是你自己版本的**malloc**，**free**，**realloc**函数。

步骤

- `tar xvf mallocclab-handout.tar`解压文件
- 我们需要修改的唯一文件是`mm.c`，包含如下几个需要实现的函数
- `int mm_init(void);`
- `void *mm_malloc(size_t size);`
- `void mm_free(void *ptr);`
- `void *mm_realloc(void *ptr, size_t size)`

步骤

- **mm_init**:在调用**mm_malloc**, **mm_realloc** 或**mm_free**之前, 调用**mm_init**进行初始化, 正确返回0。
- **mm_malloc**: 在堆区域分配指定大小的块, 分配的空间, 返回的指针应该是8字节对齐的
- **mm_free**:释放指针指向的block

步骤

- **mm_realloc**:返回指向一个大小为**size**的区域指针，满足一下条件：
- if ptr is NULL, the call is equivalent to **mm_malloc(size)**;
- if size is equal to zero, the call is equivalent to **mm_free(ptr)**;
- if ptr is not NULL: 先按照**size**指定的大小分配空间，将原有数据从头到尾拷贝到新分配的内存区域，而后释放原来**ptr**所指内存区域

步驟

- 在memory.c中定义如下函数，大家可以调用
- void ***mem_sbrk**(int incr): Expands the heap by incr bytes, where incr is a positive non-zero integer and returns a generic pointer to the first byte of the newly allocated heap area.
- „ void ***mem_heap_lo**(void): Returns a generic pointer to the first byte in the heap.
- „ void ***mem_heap_hi**(void): Returns a generic pointer to the last byte in the heap.
- „ size_t **mem_heapsize**(void): Returns the current size of the heap in bytes.
- „ size_t **mem_pagesize**(void): Returns the system' s page size in bytes (4K onLinux systems).

验证

- **mdriver.c**: 负责测试**mm.c**的正确性,空间利用率和吞吐量
- **-f <tracefile>**: **-f**后添加一些**trace file**来测试我们实现的函数
- **-V**:打印出诊断信息
- **./mdriver -V -f short1-bal.rep**

编程规则

- 不能改变mm.c中函数接口
- 不能直接调用任何内存管理的库函数和系统函数malloc, calloc, free, realloc, sbrk, brk
- 不能定义任何全局或者静态复合数据结构如arrays, structs, trees, 允许使用integers, floats, and pointers等简单数据类型
- 返回的指针需要8字节对齐

提交

- 只要提交mm.c文件