

# malloc lab 实验报告

赵浩宇,2016012390, 计科 60

2018 年 6 月 15 日

## 1 实验要求

本次实验中，要求实现一个动态存储分配的一个 C 程序，实现 malloc, free 和 realloc 三个部分。

## 2 算法概述

首先最简单的是根据课本上的代码，进行复现。之后再将课本上的程序进行修改，修改成为显式空闲链表的实现。之后再对 realloc 进行优化，即如果当前块后面是一个空闲块，而且当前块与后面空闲块的总大小可以进行 realloc，则不进行新的 alloc，直接返回原来的指针。最后根据测试样例的形式，根据数据进行相应的优化。

## 3 详细实现描述

### 3.1 对课本代码进行复现

这一个部分比较简单，就是复用书本上的代码，最后实现一点点的辅助函数即可。但是可以发现，书本上面宏的定义让后面针对‘地址’的操作变得清晰了许多。对书本上的方法进行了实现之后，用测评程序可以得到大概 50 分。

### 3.2 加入显式空闲链表

我们使用双向链表作为显示空闲链表，算法就是，讲所有空闲块连成链表的数据结构，每次新加入一个空闲块的时候，都将其放在最链表的最前面，当有一个空闲块被使用后，就将这个节点在空闲链表中删除。如果一个空闲块的空间减小，那么我的实现方法是首先将原来的空闲块从链表中删除，之后在链表头部加入一个空间比原来小的空闲链表。

至于对显式空闲链表的实现，我新建了一个全局变量 (unsigned int) 作为显式空闲链表头部的地址，之后封装了插入节点和删除节点的辅助函数。其他的地方就是调用辅助函数即可，但是需要注意提前设置‘指针’。实现完成后测试分数大概为 83 分。

### 3.3 针对 realloc 进行优化

原本 realloc 的实现方法是先进行一次 malloc，再将原来的指针 free 掉，这种实现方式看起来就没有进行优化，所以我针对一种特定的情况进行优化，即如果需要 realloc 的块后面的跟着一个空闲块，并且两块的总大小比需要真正重新分配的空间要大，那么就不再重新 malloc 了，直接将原来的非空闲块即可。

实现上与 malloc 基本类似，也是根据剩余的大小进行分类，如果剩余空间比较小，那么就不再将空闲空间分开，如果还有较大的剩余空间，那么将空闲的区域分开。加入 realloc 优化后，测试分数从 83 分变为 85 分。

### 3.4 针对数据进行优化

经过跟同学的讨论，发现如果对数据进行优化可以有小幅度的提高，即如果 malloc 的大小在 64-128 之间，那么将分配的大小变为 128，这样可以对测试分数有小幅度的提高，最终测试分数为 87 分。

## 4 思考与讨论

通过 malloc lab 的实验，我更加深入的了解动态内存分配的内容，并且知道了使用一些宏定义可以让程序变得简洁清晰，也改善了我的代码习惯。