内存池设计与实现

2011年06月29日 14:37:00 [shawngucas](https://me.csdn.net/shawngucas" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 阅读数 61343 标签： [null](https://so.csdn.net/so/search/s.do?q=null&t=blog" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[iterator](https://so.csdn.net/so/search/s.do?q=iterator&t=blog" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[byte](https://so.csdn.net/so/search/s.do?q=byte&t=blog" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[integer](https://so.csdn.net/so/search/s.do?q=integer&t=blog" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[struct](https://so.csdn.net/so/search/s.do?q=struct&t=blog" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[buffer](https://so.csdn.net/so/search/s.do?q=buffer&t=blog" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 更多

个人分类： [C/C++](https://blog.csdn.net/shawngucas/article/category/838373" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

## **1. 内存池设计**

### **1.1 目的**

在给定的内存buffer上建立内存管理机制，根据用户需求从该buffer上分配内存或者将已经分配的内存释放回buffer中。

### **1.2 要求**

尽量减少内存碎片，平均效率高于C语言的malloc和free。

### **1.3 设计思路**

将buffer分为四部分，第1部分是mem\_pool结构体；第2部分是内存映射表；第3部分是内存chunk结构体缓冲区；第4部分是实际可分配的内存区。整个buffer结构图如图1所示：

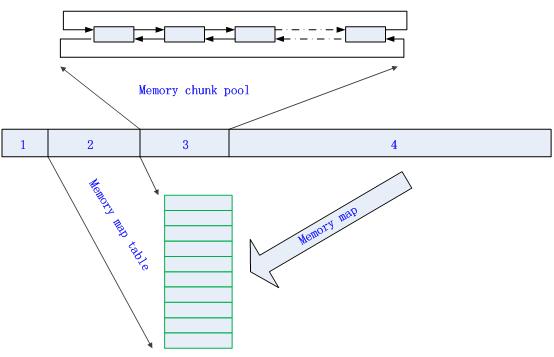


图1 内存buffer结构图

第1部分的作用是可以通过该mem\_pool结构体控制整个内存池。

第2部分的作用是记录第4部分，即实际可分配的内存区的使用情况。表中的每一个单元表示一个固定大小的内存块（block），多个连续的block组成一个chunk，每个block的详细结构如图2所示：

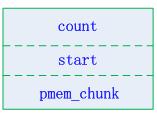


图2 memory block结构图

其中count表示该block后面的与该block同属于一个chunk的blokc的个数，start表示该block所在的chunk的起始block索引。其实start这个域只有在每个chunk的最后一个block中才会用到（用于从当前chunk寻找前一个chunk的起始位置），而pmem\_chunk则是一个指针，指向一个mem\_chunk结构体。任意一块大小的内存都会被取向上整到block大小的整数倍。

第3部分是一个mem\_chunk pool，其作用是存储整个程序可用的mem\_chunk结构体。mem\_chunk pool中的mem\_chunk被组织成双向链表结构（快速插入和删除）。每个mem\_chunk结构图如图3所示：

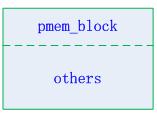


图3 memory chunk结构图

其中pmem\_block指向该chunk在内存映射表中的位置，others表示其他一些域，不同的实现对应该域的内容略有不同。

第4部分就是实际可以被分配给用户的内存。

整个内存池管理程序除了这四部分外，还有一个重要的内容就是memory chunk set。虽然其中的每个元素都来自mem\_chunk pool，但是它与mem\_chunk pool的不同之处在于其中的每个memory chunk中记录了当前可用的一块内存的相关信息。而mem\_chunk pool中的memory chunk的内容是无定以的。可以这样理解mem\_chunk pool与memory chunk set：mem\_chunk pool是为memory chunk set分配内存的“内存池”，只是该“内存池”每次分配的内存大小是固定的，为mem\_chunk结构体的大小。内存池程序主要是通过搜索这个memory chunk set来获取可被分配的内存。在memory chunk set上建立不同的数据结构就构成了不同的内存池实现方法，同时也导致了不同的搜索效率，直接影响内存池的性能，本文稍后会介绍两种内存池的实现。

### **1.4 内存池管理程序运行过程**

* 初始化：内存映射表中只有一块可用的内存信息，大小为内存池中所有可用的内存。从memory chunk pool中分配一个mem\_chunk，使其指向内存映射表中的第一个block，并根据具体的内存池实现方式填充mem\_chunk中的其他域，然后将该mem\_chunk添加到memory chunk set中。
* 申请内存：当用户申请一块内存时，首先在memory chunk set中查找合适的内存块。如果找到符合要求的内存块，就在内存映射表中找到相应的chunk，并修改chunk中相应block结构体的内容，然后根据修改后的chunk修改memory chunk set中chunk的内容，最后返回分配内存的起始地址；否则返回NULL。
* 释放内存：当用户释放一块内存时，首先根据这块内存的起始地址找到其在内存映射表中对应的chunk，然后尝试将该chunk和与其相邻的chunk合并，修改chunk中相应block的内容并修改memory chunk set中相应chunk的内容或者向memory chunk set加入新的mem\_chunk（这种情况在不能合并内存是发生）。

### **1.5 减少内存碎片**

本文设计的方法只能在一定程度上减少内存碎片，并不能彻底消除内存碎片。具体方法如下：

在用户释放内存时，尝试将该内存与其相邻的内存合并。如果其相邻内存为未分配内存则合并成功，合并后作为一整块内存使用；如火其相邻内存为已分配内存则不能合并，该释放的内存块作为一个独立的内存块被使用。

## **2 内存池实现-链表结构**

### **2.1 性能分析**

链表结构的内存池实现是指将memory chunk set实现为双链表结构。这种方法的优缺点如下：

优点：释放内存很快，O(1)复杂度。

缺点：分配内存较慢，O(n)复杂度。

### **2.2 内存池运行状态转移图**

绿色表示未使用的内存，红色表示已经使用的内存。其中每个block表示64B，这个值可以根据具体需要设定。

* 初始化

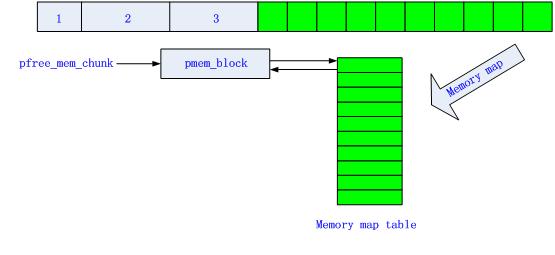


图4 内存池初始化状态

* 申请内存

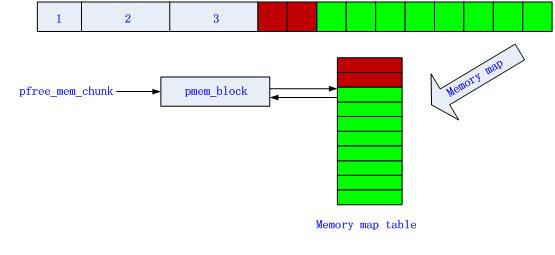


图5 第1次申请128B内存后

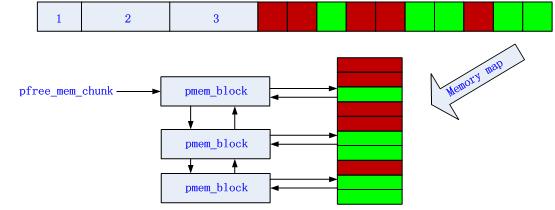


图6 第n次申请、释放内存后

* 释放内存

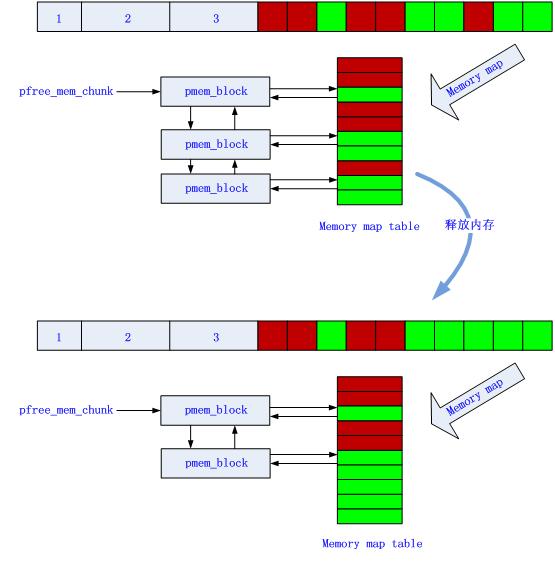


图7 释放64B内存前后

## **3 内存池实现-大顶堆结构**

### **3.1 性能分析**

大顶堆结构的内存池实现是指将memory chunk set实现为大顶堆结构。这种方法的优缺点如下：

优点：降低了分配内存的时间复杂度，O(log(n))。

缺点：增加了释放内存的时间复杂度，O(log(n))。

### **3.2 内存池运行状态转移图**

绿色表示未使用的内存，红色表示已经使用的内存。其中每个block表示64B，这个值可以根据具体需要设定。

* 初始化

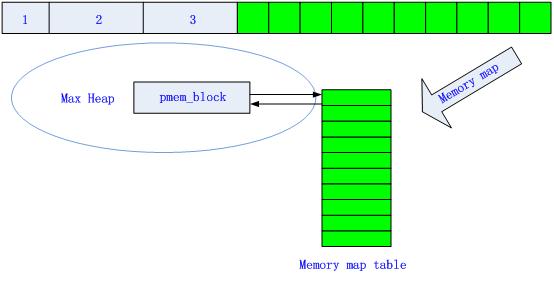


图8 内存池初始化状态

* 申请内存

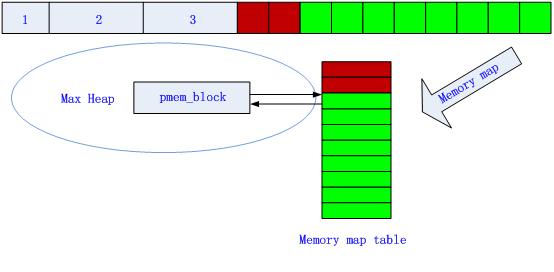


图9 第1次申请128B内存后

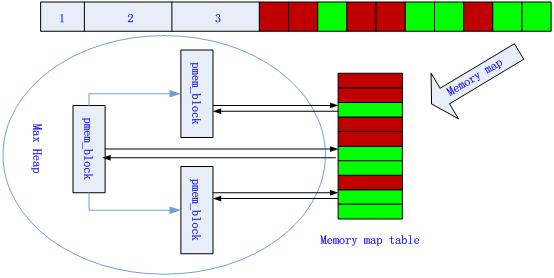


图10 第n次申请、释放内存后

* 释放内存

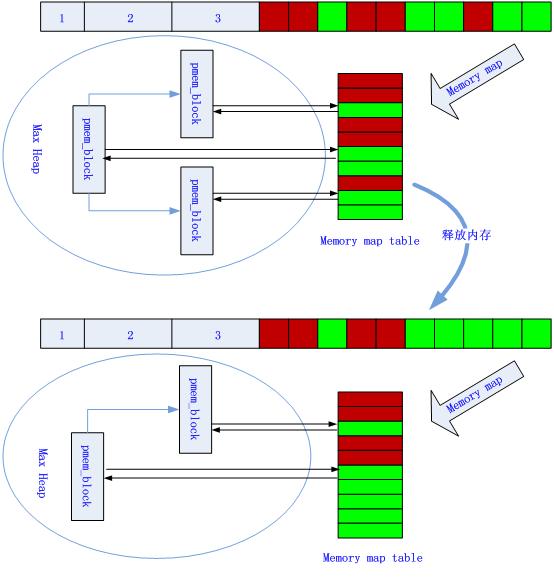


图11 释放64B内存前后

## **4 性能测试**

* 测试对象：C语言的malloc、free和本文的两种内存池（大小为500M，实际可分配内存为310M）。
* 测试指标：执行n=2000次随机分配、释放随机大小内存（范围为64B~1024B）的时间比。
* 测试方法1：

（1） 生成n个随机数，大小在64~1024之间，用于表示n个要分配的内存大小；

（2） 生成n个随机数，取值 为0或者1，表示每次分配内存后紧接着是否释放内存；

（3） 测量C语言的malloc、free和本文两种内存池执行n次随机分配、释放随机大小内存的时间比ratio；

（4） 重复（3）m=200次，记录每次活动的ratio，并绘制相应的曲线。

* 测试方法2：

（1） 生成n个随机数，大小在a~b之间（初始值a=64，b=1024），用于表示n个要分配的内存大小；

（2） 测量C语言的malloc、free和本文两种内存池执行n次分配、释放随机大小内存的时间比ratio；

（3） 重复（2）m=512次，每次分配的内存容量的范围比前一次大1024B，记录每次获得的ratio，并绘制相应曲线。

### **4.1 性能测试结果-链表结果内存池**

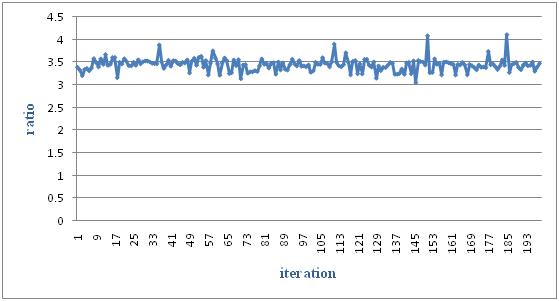


图12 链表结构内存池性能测试结果1

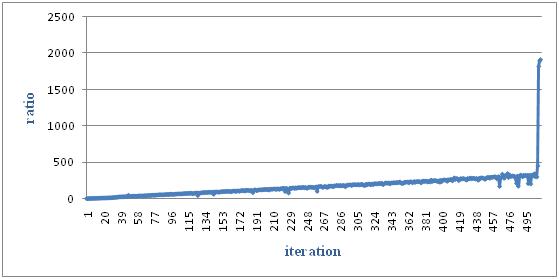


图13 链表结构内存池性能测试结果2

### **4.2 性能测试结果-大顶堆结构内存池**

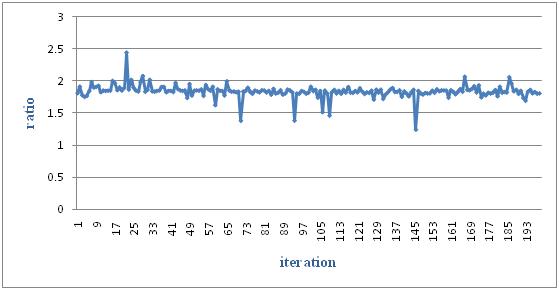


图14 大顶堆内存池性能测试结果1

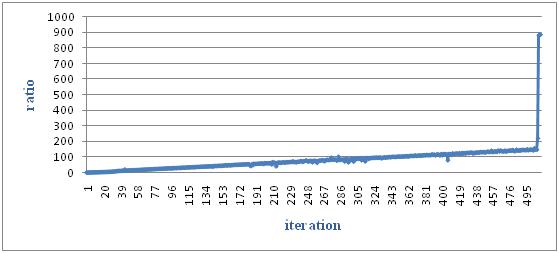


图15 大顶堆内存池性能测试结果2

### **4.3 性能比较**

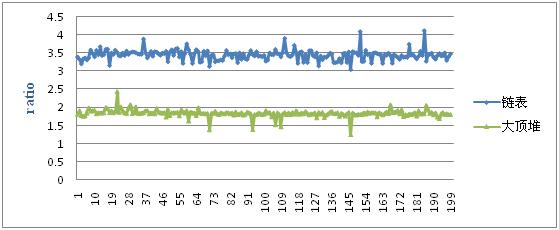


图16 两种内存池性能测试结果比较1

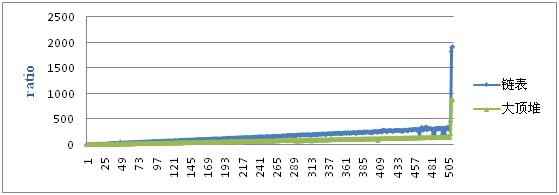


图17 两种内存池性能测试结果比较2

## **5 结论**

从上面的内存池性能测试结果中可以看出，相比C语言的malloc和free，内存池使得用户分配内存和释放内存的效率有了较大的提高，这一优势尤其分配较大快的内存时体现的尤为突出。

同时也可以看出大顶堆结够的内存池的性能并不比链表结构的内存池性能高，反而低于链表结构内存池的性能。这再一次表明O(log(n))优于O(n)是有条件的。当然，本文的测试具有一定的局限性，也许在其他的测试案例中大顶堆结构的内存池性能会超越链表结构的内存池。

## **附：源代码**

链表结构内存池：

MemoryPool.h



MemoryPool.cpp



MemoryPoolTest.cpp



大顶堆结构内存池：

MemoryPool.h



MemoryPool.cpp



MemoryPoolTest.cpp



大家都在看！深圳入户条件放宽，社保满一年即可申请，详情测分明德教育 · 顶新

[IMG_275](https://blog.csdn.net/shawngucas/article/details/javascript:void(0);)

窗体顶端



窗体底端

* [](https://me.csdn.net/God_luck)

[big\_good\_luck：](https://me.csdn.net/God_luck" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 楼主你的性能图和内存池运行图是用什么神器画的？(1年前#25楼)

* [](https://me.csdn.net/luckywll)

[luckywll：](https://me.csdn.net/luckywll" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 有两个错误 1.判断block块是否为最后一段 2.有一处地方start写成了count(2年前#24楼)

* [](https://me.csdn.net/saodilaoshu)

[六饼嫖鼠：](https://me.csdn.net/saodilaoshu" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 博主文中所写的内存管理链表节点里的size,指代的是free+used的总和吗?这里感觉有漏洞(2年前#23楼)

* [](https://me.csdn.net/erxiaodao)

[erxiaodao：](https://me.csdn.net/erxiaodao" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 好文章！ 请教下，性能测试结果图是用什么工具画的？(4年前#22楼)查看回复(3)

* [IMG_280](https://me.csdn.net/CAICHAO1234)

[头头：](https://me.csdn.net/CAICHAO1234" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 楼主，你这个有重大bug,当连续分配2 块以上的内存的时候，然后再来释放，就内存泄露，bug了(6年前#21楼)查看回复(1)

* [](https://me.csdn.net/liziminhuhu)

[逍遥龙子：](https://me.csdn.net/liziminhuhu" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 可以解释一下创建内存池的时候，那个mem\_pool-&gt;mem\_map\_unit\_count = (sBufSize - mem\_pool\_struct\_size + MINUNITSIZE - 1) / MINUNITSIZE; 这行代码吗楼主(3年前#20楼)

* [](https://me.csdn.net/saloyun)

[saloyun：](https://me.csdn.net/saloyun" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 2楼是真懒，你将上面的代码直接copy编译不就完了？还要楼主给你打包，懒成这样，也是服了。(3年前#19楼)

* [](https://me.csdn.net/qq_17354605)

[Tom\_Zou\_007：](https://me.csdn.net/qq_17354605" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 建议博主看一下别人写的，真是晦涩难懂，讲述抽象。 http://www.cnblogs.com/bangerlee/archive/2011/09/01/2161437.html 清晰易懂，不失深度。(3年前#18楼)

* [](https://me.csdn.net/qq_17354605)

[Tom\_Zou\_007：](https://me.csdn.net/qq_17354605" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 写的什么东西，生涩难懂(3年前#17楼)

* [](https://me.csdn.net/huweipan)

[huweipan：](https://me.csdn.net/huweipan" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 楼主有没有可以用的版本代码啊，可以给一份吗？谢谢(3年前#16楼)

* [](https://me.csdn.net/xukaitju)

[xukaitju：](https://me.csdn.net/xukaitju" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 太牛了，大神，很感谢(3年前#15楼)

* [](https://me.csdn.net/tifentan)

[露蛇：](https://me.csdn.net/tifentan" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 假设输入500M的内存被管理的结构吃了100多M？ 不太科学啊。(4年前#14楼)

* [](https://me.csdn.net/u011220857)

[it\_child：](https://me.csdn.net/u011220857" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 或者说315行的 size\_t index = pre\_block-&gt;count; 应该改成 size\_t index = pre\_block-&gt;start; 楼主求解释啊(5年前#13楼)

* [](https://me.csdn.net/u011220857)

[it\_child：](https://me.csdn.net/u011220857" \t "https://blog.csdn.net/shawngucas/article/details/_blank) void FreeMemory(void \*ptrMemoryBlock, PMEMORYPOOL pMem) 函数中的 309. // 最后一个 310. else if (current\_index == pMem-&gt;mem\_block\_count-1) 311. { 312. if (current\_block-&gt;count &lt; pMem-&gt;mem\_block\_count) 313. { 314. pre\_block = &amp;(pMem-&gt;pmem\_map[current\_index-1]); 315. size\_t index = pre\_block-&gt;count; 316. pre\_block = &amp;(pMem-&gt;pmem\_map[index]); 317. 318. // 如果前一个内存块是空闲的，合并 请问第316行是不是有问题啊(5年前#12楼)

* [](https://me.csdn.net/u011373710)

[樱花树樱花开：](https://me.csdn.net/u011373710" \t "https://blog.csdn.net/shawngucas/article/details/_blank) 今天刚刚学了 操作系统中的 内存管理...~我怎么就没想到 把操作系统中的这种内存管理来这样使用呢~~(5年前#11楼)

登录 查看 29 条热评

#### [内存池的设计和实现 -- 《C++应用程序性能优化》](https://blog.csdn.net/RoyalApex/article/details/8059699" \o "内存池的设计和实现 -- 《C++应用程序性能优化》" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 9237](https://blog.csdn.net/RoyalApex/article/details/8059699" \o "内存池的设计和实现  --  《C++应用程序性能优化》" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[引言本书主要针对的是C++程序的性能优化，深入介绍C++程序性能优化的方法和实例。全书由4个篇组成，第1篇介绍C++语言的对象模型，该篇是优化C++程序的基础；第2篇主要针对如何优化C++程序的内存使...](https://blog.csdn.net/RoyalApex/article/details/8059699" \o "内存池的设计和实现  --  《C++应用程序性能优化》" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 做事先做人！](https://blog.csdn.net/RoyalApex" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C语言——内存池的设计和实现](https://blog.csdn.net/weixin_40203143/article/details/85217972" \o "C语言——内存池的设计和实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 144](https://blog.csdn.net/weixin_40203143/article/details/85217972" \o "C语言——内存池的设计和实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[看完了之前两套代码后，我打算自己动手实现一个简单的内存池，我的思路与代码二类似，这样做的好处是可以给不同的对象使用不同的内存池，可以相互隔绝：memorypool下面挂一个memory的链表，可以选择...](https://blog.csdn.net/weixin_40203143/article/details/85217972" \o "C语言——内存池的设计和实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： weixin\_40203143的博客](https://blog.csdn.net/weixin_40203143" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [基于C语言的内存池的设计与实现](https://blog.csdn.net/xikangsoon/article/details/52759866" \o "基于C语言的内存池的设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1609](https://blog.csdn.net/xikangsoon/article/details/52759866" \o "基于C语言的内存池的设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[介绍：    设计内存池的目标是为了保证服务器长时间高效的运行，通过对申请空间小而申请频繁的对象进行有效管理，减少内存碎片的产生，合理分配管理用户内存，从而减少系统中出现有效空间足够，而无法分配大块连...](https://blog.csdn.net/xikangsoon/article/details/52759866" \o "基于C语言的内存池的设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： xikangsoon的博客](https://blog.csdn.net/xikangsoon" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C语言如何实现内存池](https://bbs.csdn.net/topics/330209591" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[04-07](https://bbs.csdn.net/topics/330209591" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[C语言实现内存池，C++不行，哪位高手有这源码，多谢！论坛](https://bbs.csdn.net/topics/330209591" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池例子-有代码和解析](https://download.csdn.net/download/lzp1822/2219068" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[04-09](https://download.csdn.net/download/lzp1822/2219068" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[内存池的详细设计说明，有代码和解析 简单的内存池的实现](https://download.csdn.net/download/lzp1822/2219068" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[下载](https://download.csdn.net/download/lzp1822/2219068" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [如何设计一个简单内存池](https://blog.csdn.net/LF_2016/article/details/53456402" \o "如何设计一个简单内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 4497](https://blog.csdn.net/LF_2016/article/details/53456402" \o "如何设计一个简单内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[什么是内存池？？？ 通常我们用new或malloc来分配内存的话，由于申请的大小不确定，所以当频繁的使用时会造成内存碎片和效率的降低。为了克服这种问题我们提出了内存池的概念。内存池是一种内存分配方式。...](https://blog.csdn.net/LF_2016/article/details/53456402" \o "如何设计一个简单内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： Sunnylunch-blog](https://blog.csdn.net/LF_2016" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [简单内存池设计与实现](https://blog.csdn.net/yiran_fancier/article/details/77094715" \o "简单内存池设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 429](https://blog.csdn.net/yiran_fancier/article/details/77094715" \o "简单内存池设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[使用Java进行开发的时候，不用过多地考虑内存的问题。而C/C++就不得不考虑了，比如每一次的内存申请，都不能忘了释放，不然会出现内存泄漏等等诸多问题。在网络服务程序里，每来一个请求或者链接，都会开辟...](https://blog.csdn.net/yiran_fancier/article/details/77094715" \o "简单内存池设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： YiranFancier的博客](https://blog.csdn.net/yiran_fancier" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池的设计与实现](https://blog.csdn.net/LKMIRACLE/article/details/30034663" \o "内存池的设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1323](https://blog.csdn.net/LKMIRACLE/article/details/30034663" \o "内存池的设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[内存池(MemoryPool)是一种内存分配方式。通常我们习惯直接使用new、malloc等API申请分配内存，这样做的缺点在于：由于所申请内存块的大小不定，当频繁使用时会造成大量的内存碎片并进而降低...](https://blog.csdn.net/LKMIRACLE/article/details/30034663" \o "内存池的设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： LKMIRACLE的专栏](https://blog.csdn.net/LKMIRACLE" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [固定大小内存池设计与实现](https://blog.csdn.net/wzb56/article/details/31829359" \o "固定大小内存池设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1549](https://blog.csdn.net/wzb56/article/details/31829359" \o "固定大小内存池设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[developerWorks图书频道:C++应用程序性能优化，第6章：内存池本书主要针对的是C++程序的性能优化，深入介绍C++程序性能优化的方法和实例。全书由4个篇组成，第1篇介绍C++语言的对象模...](https://blog.csdn.net/wzb56/article/details/31829359" \o "固定大小内存池设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： wzb56的资料库](https://blog.csdn.net/wzb56" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池设计与实现 - 华的专栏 - CSDN博客](https://blog.csdn.net/wh8_2011/article/details/88101346" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[3-4](https://blog.csdn.net/wh8_2011/article/details/88101346" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[https://blog.csdn.net/shawngucas/article/details/6574863 阅读更多 收藏 分享...博文 来自: ugg的专栏 简单内存池设计与实现 08-11 阅读数 304 使用Java...](https://blog.csdn.net/wh8_2011/article/details/88101346" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池设计与实现 - zhaojiecll的专栏 - CSDN博客](https://blog.csdn.net/zhaojiecll/article/details/45362359" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[10-31](https://blog.csdn.net/zhaojiecll/article/details/45362359" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[转自http://blog.csdn.net/shawngucas/article/details/6574863 感谢原作者的分享。 1. 内存池设计 1.1 目的 在给定的内存buffer上建立内存管理机制,根据用户...](https://blog.csdn.net/zhaojiecll/article/details/45362359" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

【实拍】7月起，建设有种学历叫自考本科，一年课程，正规本科，学信网可查爱华国际 · 顶新

#### [内存池的实现](https://blog.csdn.net/innobase/article/details/52628451" \o "内存池的实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 581](https://blog.csdn.net/innobase/article/details/52628451" \o "内存池的实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[基本的架构设计：基本运行结果：$./mem\_pool...UseBlocks[0]=(nil),FreeBlocks[0]=(nil)...UseBlocks[1]=(nil),FreeBlocks[...](https://blog.csdn.net/innobase/article/details/52628451" \o "内存池的实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： innobase的博客](https://blog.csdn.net/innobase" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C语言——内存池的设计和实现 - weixin\_40203143的博客 - CSDN博客](https://blog.csdn.net/weixin_40203143/article/details/85217972" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[6-27](https://blog.csdn.net/weixin_40203143/article/details/85217972" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [简单内存池设计与实现 - YiranFancier的博客 - CSDN博客](https://blog.csdn.net/yiran_fancier/article/details/77094715" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[11-4](https://blog.csdn.net/yiran_fancier/article/details/77094715" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[,所以参照了Apache的内存池进行简单的设计(修剪<(~︶~)>)和在Linux下的实现...来自: shawn的专栏 内存池的设计与实现 06-11 1177 内存池(Memory Pool)...](https://blog.csdn.net/yiran_fancier/article/details/77094715" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C++ 实现高性能内存池](https://blog.csdn.net/xjtuse2014/article/details/52302083" \o "C++ 实现高性能内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1万+](https://blog.csdn.net/xjtuse2014/article/details/52302083" \o "C++ 实现高性能内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[一、概述在C/C++中，内存管理是一个非常棘手的问题，我们在编写一个程序的时候几乎不可避免的要遇到内存的分配逻辑，这时候随之而来的有这样一些问题：是否有足够的内存可供分配?分配失败了怎么办?如何管理自...](https://blog.csdn.net/xjtuse2014/article/details/52302083" \o "C++ 实现高性能内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： xjtuse2014的博客](https://blog.csdn.net/xjtuse2014" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C++内存池介绍与经典内存池的实现 - Dablelv的博客专栏 - CSDN博客](https://blog.csdn.net/K346K346/article/details/49538975" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[6-27](https://blog.csdn.net/K346K346/article/details/49538975" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [设计一个简单的内存池 - everthing willl be ok - CSDN博客](https://blog.csdn.net/qq_37964547/article/details/81335119" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[11-9](https://blog.csdn.net/qq_37964547/article/details/81335119" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[以上就是一个简单的内存池的设计过程。 阅读更多 想对作者说点什么? 我来说一句 内存池设计与实现 - shawn的专栏 06-29 5.9万 1. 内存池设计1.1 目的...](https://blog.csdn.net/qq_37964547/article/details/81335119" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [高效内存池的设计方案[C语言]](https://blog.csdn.net/RoyalApex/article/details/8195357" \o "高效内存池的设计方案[C语言]" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 8187](https://blog.csdn.net/RoyalApex/article/details/8195357" \o "高效内存池的设计方案[C语言]" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[一、前言概述本人在转发的博文《内存池的设计和实现》中，详细阐述了系统默认内存分配函数malloc/free的缺点，以及进行内存池设计的原因，在此不再赘述。通过对Nginx内存池以及《内存池的设计和实现...](https://blog.csdn.net/RoyalApex/article/details/8195357" \o "高效内存池的设计方案[C语言]" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 做事先做人！](https://blog.csdn.net/RoyalApex" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池设计和原理](https://blog.csdn.net/yuanchunsi/article/details/78497624" \o "内存池设计和原理" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 327](https://blog.csdn.net/yuanchunsi/article/details/78497624" \o "内存池设计和原理" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[C/C++基础巩固](https://blog.csdn.net/yuanchunsi/article/details/78497624" \o "内存池设计和原理" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： yuanchunsi的博客](https://blog.csdn.net/yuanchunsi" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [基于C语言的内存池的设计与实现 - xikangsoon的博客 - CSDN博客](https://blog.csdn.net/xikangsoon/article/details/52759866" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[4-21](https://blog.csdn.net/xikangsoon/article/details/52759866" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[设计内存池的目标是为了保证服务器长时间高效的运行,通过对申请空间小而申请频繁的对象进行有效管理,减少内存碎片的产生,合理分配管理用户内存,从而减少系统中出现有效...](https://blog.csdn.net/xikangsoon/article/details/52759866" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [需要自己建立内存池吗? - yazhouren的专栏 - CSDN博客](https://blog.csdn.net/yazhouren/article/details/77816142" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[11-14](https://blog.csdn.net/yazhouren/article/details/77816142" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存管理的源码实现（内存池实现源代码）](https://download.csdn.net/download/yiyinianhua/2821715" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[11-10](https://download.csdn.net/download/yiyinianhua/2821715" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[内存管理的源码实现，通过设计一个内存池，实现对内存的管理，对于理解内存管理有很大的帮助](https://download.csdn.net/download/yiyinianhua/2821715" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[下载](https://download.csdn.net/download/yiyinianhua/2821715" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

七月起，深圳有种成人学历叫免考本科，一年课程，在职可读！爱华学历中心 · 猎媒

#### [设计一个内存池，怎么做呢？](https://bbs.csdn.net/topics/70516139" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[08-12](https://bbs.csdn.net/topics/70516139" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[设计一个内存池，怎么做呢？论坛](https://bbs.csdn.net/topics/70516139" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [Golang多级内存池设计与实现](https://blog.csdn.net/uxiAD7442KMy1X86DtM3/article/details/79447210" \o "Golang多级内存池设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 547](https://blog.csdn.net/uxiAD7442KMy1X86DtM3/article/details/79447210" \o "Golang多级内存池设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[上个月，牙膏厂intel因为Meltdown和Spectre两个bug需要给CPU固件和系统打了补丁。我们生产环境使用的是阿里云，打完补丁后，几台IO密集型的机器性能下降明显，从流量和cpuload估...](https://blog.csdn.net/uxiAD7442KMy1X86DtM3/article/details/79447210" \o "Golang多级内存池设计与实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 架构文摘](https://blog.csdn.net/uxiAD7442KMy1X86DtM3" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [Netty内存池整体设计解读](https://blog.csdn.net/kuangzhanshatian/article/details/80881904" \o "Netty内存池整体设计解读" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1917](https://blog.csdn.net/kuangzhanshatian/article/details/80881904" \o "Netty内存池整体设计解读" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[Netty内存池整体设计解读Netty内存池整体设计解读背景名词定义总体设计思路单个Chunk的内存分配算法分配内存释放内存Chunk移动小容量内存分配算法分配算法释放内存管理方式优化改进线程缓存流程...](https://blog.csdn.net/kuangzhanshatian/article/details/80881904" \o "Netty内存池整体设计解读" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： kuangzhanshatian的专栏](https://blog.csdn.net/kuangzhanshatian" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池设计及实现参考资料](https://blog.csdn.net/jsh13417/article/details/25792905" \o "内存池设计及实现参考资料" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1339](https://blog.csdn.net/jsh13417/article/details/25792905" \o "内存池设计及实现参考资料" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[利用默认的内存管理函数new/delete或malloc/free在堆上分配和释放内存会有一些额外的开销。系统在接收到分配一定大小内存的请求时，首先查找内部维护的内存空闲块表，并且需要根据一定的算法（...](https://blog.csdn.net/jsh13417/article/details/25792905" \o "内存池设计及实现参考资料" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 轻飘飞扬](https://blog.csdn.net/jsh13417" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C++内存池介绍与经典内存池的实现](https://blog.csdn.net/K346K346/article/details/49538975" \o "C++内存池介绍与经典内存池的实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1万+](https://blog.csdn.net/K346K346/article/details/49538975" \o "C++内存池介绍与经典内存池的实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[1.默认内存管理函数的不足利用默认的内存管理函数new/delete或malloc/free在堆上分配和释放内存会有一些额外的开销。系统在接收到分配一定大小内存的请求时，首先查找内部维护的内存空闲块表...](https://blog.csdn.net/K346K346/article/details/49538975" \o "C++内存池介绍与经典内存池的实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： Dablelv的博客专栏](https://blog.csdn.net/K346K346" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

医生提醒!千寒易除，一湿难去，去湿气的简单方法在这里！豫用 · 猎媒

#### [固定大小块的内存池设计](https://blog.csdn.net/wbwwf8685/article/details/51897461" \o "固定大小块的内存池设计" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 783](https://blog.csdn.net/wbwwf8685/article/details/51897461" \o "固定大小块的内存池设计" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[摘要在一些场景下，应用需要频繁的申请和释放一个或多个大小相同的内存（例如传输固定大小的数据块），这时候如果调用的是普通的malloc和free函数，效率就相对较低，且分配和释放的次数多，会容易导致内存...](https://blog.csdn.net/wbwwf8685/article/details/51897461" \o "固定大小块的内存池设计" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 亦大乐谍的嵌入式分享小站——思考/验证/总结/分享](https://blog.csdn.net/wbwwf8685" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [设计模式之最简单的单例模式](https://blog.csdn.net/weixin_40839342/article/details/80640951" \o "设计模式之最简单的单例模式" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 391](https://blog.csdn.net/weixin_40839342/article/details/80640951" \o "设计模式之最简单的单例模式" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[publicclassSingleTon{//1.将构造方法私有化，不允许外部直接创建对象privateSingleTon(){}//类加载时static就被初始化恶汉privatestaticSin...](https://blog.csdn.net/weixin_40839342/article/details/80640951" \o "设计模式之最简单的单例模式" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： weixin\_40839342的博客](https://blog.csdn.net/weixin_40839342" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [简单的内存池实现](https://blog.csdn.net/qq_32105589/article/details/50203741" \o "简单的内存池实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 274](https://blog.csdn.net/qq_32105589/article/details/50203741" \o "简单的内存池实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[实现简单的内存池，高效newdelete结点](https://blog.csdn.net/qq_32105589/article/details/50203741" \o "简单的内存池实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： qq\_32105589的博客](https://blog.csdn.net/qq_32105589" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [用C++实现自己的内存池](https://blog.csdn.net/chexlong/article/details/7071922" \o "用C++实现自己的内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1万+](https://blog.csdn.net/chexlong/article/details/7071922" \o "用C++实现自己的内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[打开浏览器，搜索了下内存管理的概念，百度百科中是这样定义的：内存管理，是指软件运行时对计算机内存资源的分配和使用的技术。其最主要的目的是如何高效，快速的分配，并且在适当的时候释放和回收内存资源。说到内...](https://blog.csdn.net/chexlong/article/details/7071922" \o "用C++实现自己的内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： Bruce Che 的专栏](https://blog.csdn.net/chexlong" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池设计与实现 .](https://blog.csdn.net/fendouxiaoxiaoniao/article/details/45395681" \o "内存池设计与实现 ." \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 321](https://blog.csdn.net/fendouxiaoxiaoniao/article/details/45395681" \o "内存池设计与实现 ." \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[原地址：http://blog.csdn.net/shawngucas/article/details/6574863](https://blog.csdn.net/fendouxiaoxiaoniao/article/details/45395681" \o "内存池设计与实现 ." \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： fendouxiaoxiaoniao的专栏](https://blog.csdn.net/fendouxiaoxiaoniao" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [c++内存池实现](https://blog.csdn.net/neustar1/article/details/7478311" \o "c++内存池实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1万+](https://blog.csdn.net/neustar1/article/details/7478311" \o "c++内存池实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[利用C/C++开发大型应用程序中，内存的管理与分配是一个需要认真考虑的部分。本文描述了内存池设计原理并给出内存池的实现代码，代码支持Windows和Linux，多线程安全。内存池设计过程中需要考虑好内...](https://blog.csdn.net/neustar1/article/details/7478311" \o "c++内存池实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： neustar1的专栏](https://blog.csdn.net/neustar1" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [【小项目】内存池的实现](https://blog.csdn.net/pointer_y/article/details/53456149" \o "【小项目】内存池的实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 950](https://blog.csdn.net/pointer_y/article/details/53456149" \o "【小项目】内存池的实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[1.为什么会有内存池的出现。我们在频繁的开辟和释放小块空间时，很有可能造成内存外碎片的问题。为什么呢？看下图：上图中还回来的20k和剩下的20k就可以称之为内存外碎片问题。除了外碎片的问题，频繁的在内...](https://blog.csdn.net/pointer_y/article/details/53456149" \o "【小项目】内存池的实现" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： pointer\_y的博客](https://blog.csdn.net/pointer_y" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C语言实现内存池](https://blog.csdn.net/gongmin856/article/details/79584663" \o "C语言实现内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 361](https://blog.csdn.net/gongmin856/article/details/79584663" \o "C语言实现内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[转自：http://blog.csdn.net/szkbsgy/article/details/50491216，我根据自己情况做了部分修改。在编程中，为了避免由于频繁的malloc/free产生内存...](https://blog.csdn.net/gongmin856/article/details/79584663" \o "C语言实现内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 海阔天空的专栏](https://blog.csdn.net/gongmin856" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [各种池---内存池的高效实现（C语言）](https://blog.csdn.net/NBA_1/article/details/84975544" \o "各种池---内存池的高效实现（C语言）" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 199](https://blog.csdn.net/NBA_1/article/details/84975544" \o "各种池---内存池的高效实现（C语言）" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[转自https://blog.csdn.net/forever5312/article/details/78029672序言在编程过程中，尤其是对于C语言开发者，其实编程就是在使用内存，不停地变化内存...](https://blog.csdn.net/NBA_1/article/details/84975544" \o "各种池---内存池的高效实现（C语言）" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： NBA\_1的博客](https://blog.csdn.net/NBA_1" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池&amp;经典的内存池技术](https://blog.csdn.net/hairetz/article/details/4490542" \o "内存池&amp;经典的内存池技术" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 2651](https://blog.csdn.net/hairetz/article/details/4490542" \o "内存池&amp;经典的内存池技术" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[网上有一关于内存池的文章，拿来分享一下：内存池（MemPool）技术备受推崇。我用google搜索了下，没有找到比较详细的原理性的文章，故此补充一个。另外，补充了boost::pool组件与经典Mem...](https://blog.csdn.net/hairetz/article/details/4490542" \o "内存池&amp;经典的内存池技术" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： hairetz的专栏](https://blog.csdn.net/hairetz" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

别再玩假传奇了！这款传奇爆率9.8，你找到充值入口算我输！贪玩游戏 · 顶新

#### [内存池(C语言实现)](https://download.csdn.net/download/yangyong1980919/4195024" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[04-02](https://download.csdn.net/download/yangyong1980919/4195024" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[内存池(C语言实现)内存池(C语言实现)内存池(C语言实现)内存池(C语言实现)](https://download.csdn.net/download/yangyong1980919/4195024" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[下载](https://download.csdn.net/download/yangyong1980919/4195024" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池实现原理 (中文)](https://download.csdn.net/download/lzbcll8254011/2408654" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[05-30](https://download.csdn.net/download/lzbcll8254011/2408654" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[自定义内存池的思想通过这个"池"字表露无疑，应用程序可以通过系统的内存分配调用预先一次性申请适当大小的内存作为一个内存池，之后应用程序自己对内存的分配和释放则可以通过这个内存池来完成。只有当内存池大小](https://download.csdn.net/download/lzbcll8254011/2408654" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[下载](https://download.csdn.net/download/lzbcll8254011/2408654" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C语言实现完整功能的内存池](https://download.csdn.net/download/a29562268/10687052" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[09-25](https://download.csdn.net/download/a29562268/10687052" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[使用C语言编写的一个可以用来使用的内存池，编写方式达到内存0碎片的标准。](https://download.csdn.net/download/a29562268/10687052" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[下载](https://download.csdn.net/download/a29562268/10687052" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [一块高效的C语言内存池](https://download.csdn.net/download/tr0217/7120137" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[03-30](https://download.csdn.net/download/tr0217/7120137" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[这是一款高效、灵活、跨平台的内存池实现。使用MIT Licence发布，完全不排斥商业使用。它已经在许多生产环境中使用了。在该实现中内存被划分为节点（node）和切片（slice）。node一大块内存](https://download.csdn.net/download/tr0217/7120137" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[下载](https://download.csdn.net/download/tr0217/7120137" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C++实现内存池](https://blog.csdn.net/u010183728/article/details/81531392" \o "C++实现内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 3334](https://blog.csdn.net/u010183728/article/details/81531392" \o "C++实现内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[1.内存池设计1.1目的在给定的内存buffer上建立内存管理机制，根据用户需求从该buffer上分配内存或者将已经分配的内存释放回buffer中。1.2要求尽量减少内存碎片，平均效率高于C语言的ma...](https://blog.csdn.net/u010183728/article/details/81531392" \o "C++实现内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 一只大笨猫的博客](https://blog.csdn.net/u010183728" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [谁有高效的内存池，提供原理或者实现都可以，非常感谢](https://bbs.csdn.net/topics/120011858" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[07-22](https://bbs.csdn.net/topics/120011858" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[我自己开发一个分布式FTP服务软件,开发完成后通过对比，发现标准库或者STL或者MFC库的数据或者容器，在开发客户端时问题不大，但是我把它用在服务器端的数据结构中后，存在效率问题，而这些效率问题，无论论坛](https://bbs.csdn.net/topics/120011858" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池简单实现（一）](https://blog.csdn.net/u013043408/article/details/81508871" \o "内存池简单实现（一）" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 748](https://blog.csdn.net/u013043408/article/details/81508871" \o "内存池简单实现（一）" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[概念内存池就是在程序启动时，预先向堆中申请一部分内存，交给一个管理对象。在程序运行中，需要时向管理对象“借”，不需要时“还”给管理对象。原理很简单，关键是怎样才能高效地“借”、“还”。实例场景在C/S...](https://blog.csdn.net/u013043408/article/details/81508871" \o "内存池简单实现（一）" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： Hi, Boy](https://blog.csdn.net/u013043408" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池](https://blog.csdn.net/mercy_ps/article/details/82947394" \o "内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 511](https://blog.csdn.net/mercy_ps/article/details/82947394" \o "内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[为什么需要内存池？C/C++下我们经常需要分配足够的内存、追踪内存的分配、在不需要的时候释放内存——这个任务相当复杂。而直接使用系统调用malloc/free、new/delete进行内存分配和释放，...](https://blog.csdn.net/mercy_ps/article/details/82947394" \o "内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： mercy\_ps的博客](https://blog.csdn.net/mercy_ps" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池------------------到底有没有存在的必要?????????????????????](https://bbs.csdn.net/topics/360052385" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[02-21](https://bbs.csdn.net/topics/360052385" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[频繁的申请和释放内存会降低程序的效率,因此诞生了内存池. 我自己写了个IOCP,想进一步提高效率,于是想用内存池. 想了一会,写了大概几十行代码,越来越发现没有写的必要. 内存池的作用:一次向系统申请论坛](https://bbs.csdn.net/topics/360052385" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [分享一个内存池方案，求优化~~](https://bbs.csdn.net/topics/390677585" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[12-24](https://bbs.csdn.net/topics/390677585" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \* memory pool \* \* created by Sandr论坛](https://bbs.csdn.net/topics/390677585" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [Nginx源码学习——内存池](https://blog.csdn.net/u012109704/article/details/79773039" \o "Nginx源码学习——内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 597](https://blog.csdn.net/u012109704/article/details/79773039" \o "Nginx源码学习——内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[Nginx特点是占有内存少，并发能力强，这得益于Nginx所采用的内存管理手段——内存池。内存池的目的是预先分配较大的内存块留作备用，当有新的内存需求时，可直接在内存池中分配（指针移动），避免了频...](https://blog.csdn.net/u012109704/article/details/79773039" \o "Nginx源码学习——内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： handaifeng的专栏](https://blog.csdn.net/u012109704" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [从内存池到连接池 老码农眼中的资源池](https://blog.csdn.net/wireless_com/article/details/79072305" \o "从内存池到连接池 老码农眼中的资源池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 3254](https://blog.csdn.net/wireless_com/article/details/79072305" \o "从内存池到连接池 老码农眼中的资源池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[如果将互联网应用比喻成冲浪的话，可能需要先学会在池中游泳吧。引子AI赋能万物，老码农的伙伴们也曾经开发了一个基于图数据库的知识问答系统，在压力测试的时候发现随着并发数的增加，响应的时延明显变长，看时延...](https://blog.csdn.net/wireless_com/article/details/79072305" \o "从内存池到连接池 老码农眼中的资源池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 我相信......](https://blog.csdn.net/wireless_com" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [高并发内存池](https://blog.csdn.net/hahahaxiaozhao/article/details/88602851" \o "高并发内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 78](https://blog.csdn.net/hahahaxiaozhao/article/details/88602851" \o "高并发内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[高并发内存池内存池是一种动态内存分配与管理技术。通常情况下，程序员习惯直接使用new、delete、malloc、free等API申请分配和释放内存，这样导致的后果是：当程序长时间运行时，由于所申请内...](https://blog.csdn.net/hahahaxiaozhao/article/details/88602851" \o "高并发内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： hahahaxiaozhao的博客](https://blog.csdn.net/hahahaxiaozhao" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [菜鸟学习Nginx之内存池](https://blog.csdn.net/xxb249/article/details/81783612" \o "菜鸟学习Nginx之内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 86](https://blog.csdn.net/xxb249/article/details/81783612" \o "菜鸟学习Nginx之内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[从今天开始深入介绍Nginx框架。首先来谈谈我对《深入理解Nginx模块开发与架构解析》看法，这本书应该是到目前为止，市面写的最详细，最充实的书籍（没有之一），值得拥有。然而此书对于一个小白来说，并不...](https://blog.csdn.net/xxb249/article/details/81783612" \o "菜鸟学习Nginx之内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 程序员的世界](https://blog.csdn.net/xxb249" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [【C++学习】C++实现高效内存池](https://blog.csdn.net/TwT520Ly/article/details/80924224" \o "【C++学习】C++实现高效内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 424](https://blog.csdn.net/TwT520Ly/article/details/80924224" \o "【C++学习】C++实现高效内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[1.内存池简介2.函数实现2.1主函数2.2StackAlloc.h2.3StackAlloc.cpp2.4禁用拷贝赋值2.5静态断言与断言2.6使用reinterpret\_cast1.内存池简介内存...](https://blog.csdn.net/TwT520Ly/article/details/80924224" \o "【C++学习】C++实现高效内存池" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 小小小草儿的博客哟](https://blog.csdn.net/TwT520Ly" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [Nginx源码剖析之内存池，与内存管理](https://blog.csdn.net/v_JULY_v/article/details/7040425" \o "Nginx源码剖析之内存池，与内存管理" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 6万+](https://blog.csdn.net/v_JULY_v/article/details/7040425" \o "Nginx源码剖析之内存池，与内存管理" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[Nginx源码剖析之内存池，与内存管理作者：July、dreamice、阿波、yixiao。出处：http://blog.csdn.net/v\_JULY\_v/。引言    Nginx（发音同engin...](https://blog.csdn.net/v_JULY_v/article/details/7040425" \o "Nginx源码剖析之内存池，与内存管理" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 结构之法 算法之道](https://blog.csdn.net/v_JULY_v" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池技术介绍](https://blog.csdn.net/zgaoq/article/details/79145460" \o "内存池技术介绍" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 418](https://blog.csdn.net/zgaoq/article/details/79145460" \o "内存池技术介绍" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[看到一篇关于内存池技术的介绍文章，受益匪浅，转贴至此。 原贴地址：http://www.ibm.com/developerworks/cn/linux/l-cn-ppp/index6.html 6.1...](https://blog.csdn.net/zgaoq/article/details/79145460" \o "内存池技术介绍" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 一缕阳光的博客](https://blog.csdn.net/zgaoq" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [C++代码片段（三）过滤可变模板参数中某些类型的变量](https://blog.csdn.net/sinat_35261315/article/details/80500617" \o "C++代码片段（三）过滤可变模板参数中某些类型的变量" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 137](https://blog.csdn.net/sinat_35261315/article/details/80500617" \o "C++代码片段（三）过滤可变模板参数中某些类型的变量" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[将可变模板参数列表中的某些类型过滤掉，然后返回剩下数据的元组。用到了上一篇中的判断可变模板参数中是否包含某一特定类型的方法#include&amp;lt;iostream&amp;gt;#includ...](https://blog.csdn.net/sinat_35261315/article/details/80500617" \o "C++代码片段（三）过滤可变模板参数中某些类型的变量" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 一个程序渣渣的小后院](https://blog.csdn.net/sinat_35261315" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [c++\_cast\_最完整最详细的解释](https://blog.csdn.net/gyymen/article/details/53534200" \o "c++_cast_最完整最详细的解释" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 4436](https://blog.csdn.net/gyymen/article/details/53534200" \o "c++_cast_最完整最详细的解释" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[关于强制类型转换的问题，很多书都讨论过，写的最详细的是C++之父的《C++的设计和演化》。最好的解决方法就是不要使用C风格的强制类型转换，而是使用标准C++的类型转换符：static\_cast,dyn...](https://blog.csdn.net/gyymen/article/details/53534200" \o "c++_cast_最完整最详细的解释" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： DotLive](https://blog.csdn.net/gyymen" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [求助：C++内存池Mempool问题](http://ask.csdn.net/questions/700530" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[09-19](http://ask.csdn.net/questions/700530" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[-问答](http://ask.csdn.net/questions/700530" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [内存池和对象池的区别](https://bbs.csdn.net/topics/390732759" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[03-15](https://bbs.csdn.net/topics/390732759" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[经常听到内存池，也听到说对象池，最近仔细学习了一下，他们好像是一个东西啊。有什么区别吗，请大侠们指点一下。谢谢！论坛](https://bbs.csdn.net/topics/390732759" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [python图片处理类之~PIL.Image模块(ios android icon图标自动生成处理)](https://blog.csdn.net/themagickeyjianan/article/details/52386981" \o "python图片处理类之~PIL.Image模块(ios android icon图标自动生成处理)" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 14万+](https://blog.csdn.net/themagickeyjianan/article/details/52386981" \o "python图片处理类之~PIL.Image模块(ios android icon图标自动生成处理)" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[1.从pyCharm提示下载PIL包nn http://www.pythonware.com/products/pil/nn nn2.解压后，进入到目录下nnncd /Users/jianan/Dow...](https://blog.csdn.net/themagickeyjianan/article/details/52386981" \o "python图片处理类之~PIL.Image模块(ios android icon图标自动生成处理)" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 专注于cocos+unity+node.js全栈网游开发](https://blog.csdn.net/themagickeyjianan" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [搭建单机版的FastDFS服务器](https://blog.csdn.net/u012453843/article/details/69951920" \o "搭建单机版的FastDFS服务器" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1万+](https://blog.csdn.net/u012453843/article/details/69951920" \o "搭建单机版的FastDFS服务器" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[由于FastDFS集群搭建非常复杂，对于初期学习FastDFS来说，搭建个单机版的作为入门更为实际一些。n第一步：搭建虚拟环境n        FastDFS需要在Linux系统上安装，我们一般使用的...](https://blog.csdn.net/u012453843/article/details/69951920" \o "搭建单机版的FastDFS服务器" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： u012453843的专栏](https://blog.csdn.net/u012453843" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [jquery/js实现一个网页同时调用多个倒计时(最新的)](https://blog.csdn.net/wuchengzeng/article/details/50037611" \o "jquery/js实现一个网页同时调用多个倒计时(最新的)" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 59万+](https://blog.csdn.net/wuchengzeng/article/details/50037611" \o "jquery/js实现一个网页同时调用多个倒计时(最新的)" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[jquery/js实现一个网页同时调用多个倒计时(最新的)nn最近需要网页添加多个倒计时. 查阅网络,基本上都是千遍一律的不好用. 自己按需写了个.希望对大家有用. 有用请赞一个哦!nnnn//jsn...](https://blog.csdn.net/wuchengzeng/article/details/50037611" \o "jquery/js实现一个网页同时调用多个倒计时(最新的)" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： Websites](https://blog.csdn.net/wuchengzeng" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [异常点/离群点检测算法——LOF](https://blog.csdn.net/wangyibo0201/article/details/51705966" \o "异常点/离群点检测算法——LOF" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 8万+](https://blog.csdn.net/wangyibo0201/article/details/51705966" \o "异常点/离群点检测算法——LOF" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[局部异常因子算法-Local Outlier Factor(LOF)　　在数据挖掘方面，经常需要在做特征工程和模型训练之前对数据进行清洗，剔除无效数据和异常数据。异常检测也是数据挖掘的一个方向，用于反...](https://blog.csdn.net/wangyibo0201/article/details/51705966" \o "异常点/离群点检测算法——LOF" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： wangyibo0201的博客](https://blog.csdn.net/wangyibo0201" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [【深度剖析HMM（附Python代码）】1.前言及隐马尔科夫链HMM的背景](https://blog.csdn.net/tostq/article/details/70846702" \o "【深度剖析HMM（附Python代码）】1.前言及隐马尔科夫链HMM的背景" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 3万+](https://blog.csdn.net/tostq/article/details/70846702" \o "【深度剖析HMM（附Python代码）】1.前言及隐马尔科夫链HMM的背景" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[1. 前言rn隐马尔科夫HMM模型是一类重要的机器学习方法，其主要用于序列数据的分析，广泛应用于语音识别、文本翻译、序列预测、中文分词等多个领域。虽然近年来，由于RNN等深度学习方法的发展，HMM模型...](https://blog.csdn.net/tostq/article/details/70846702" \o "【深度剖析HMM（附Python代码）】1.前言及隐马尔科夫链HMM的背景" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： tostq的专栏](https://blog.csdn.net/tostq" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [微信支付V3微信公众号支付PHP教程(thinkPHP5公众号支付)/JSSDK的使用](https://blog.csdn.net/hayixia606/article/details/79237220" \o "微信支付V3微信公众号支付PHP教程(thinkPHP5公众号支付)/JSSDK的使用" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 20万+](https://blog.csdn.net/hayixia606/article/details/79237220" \o "微信支付V3微信公众号支付PHP教程(thinkPHP5公众号支付)/JSSDK的使用" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[扫二维码关注，获取更多技术分享nnn 本文承接之前发布的博客《 微信支付V3微信公众号支付PHP教程/thinkPHP5公众号支付》必须阅读上篇文章后才可以阅读这篇文章。由于最近一段时间工作比较忙，...](https://blog.csdn.net/hayixia606/article/details/79237220" \o "微信支付V3微信公众号支付PHP教程(thinkPHP5公众号支付)/JSSDK的使用" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： Marswill](https://blog.csdn.net/hayixia606" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [idea创建springcloud项目图文教程(EurekaServer注册中心)（六）](https://blog.csdn.net/hcmony/article/details/77855158" \o "idea创建springcloud项目图文教程(EurekaServer注册中心)（六）" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 5万+](https://blog.csdn.net/hcmony/article/details/77855158" \o "idea创建springcloud项目图文教程(EurekaServer注册中心)（六）" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[上一篇： nnidea创建springboot项目图文教程(配置文件)（五）nn nn      http://blog.csdn.net/hcmony/article/details/7785499...](https://blog.csdn.net/hcmony/article/details/77855158" \o "idea创建springcloud项目图文教程(EurekaServer注册中心)（六）" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： hcmony的博客](https://blog.csdn.net/hcmony" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [linux上安装Docker(非常简单的安装方法)](https://blog.csdn.net/qq_36892341/article/details/73918672" \o "linux上安装Docker(非常简单的安装方法)" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 32万+](https://blog.csdn.net/qq_36892341/article/details/73918672" \o "linux上安装Docker(非常简单的安装方法)" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[最近比较有空，大四出来实习几个月了，作为实习狗的我，被叫去研究Docker了，汗汗！nnDocker的三大核心概念：镜像、容器、仓库n镜像：类似虚拟机的镜像、用俗话说就是安装文件。n容器：类似一个轻量...](https://blog.csdn.net/qq_36892341/article/details/73918672" \o "linux上安装Docker(非常简单的安装方法)" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 我走小路的博客](https://blog.csdn.net/qq_36892341" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [webService学习（二）—— 调用自定义对象参数](https://blog.csdn.net/wild46cat/article/details/54572361" \o "webService学习（二）—— 调用自定义对象参数" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 4万+](https://blog.csdn.net/wild46cat/article/details/54572361" \o "webService学习（二）—— 调用自定义对象参数" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[webService学习（二）—— 调用自定义对象参数rn本文主要内容：rn1、如何通过idea进行webService Client的简单实现（不再使用wsimport的方式，其实是ide帮我们做了...](https://blog.csdn.net/wild46cat/article/details/54572361" \o "webService学习（二）—— 调用自定义对象参数" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 止水的专栏](https://blog.csdn.net/wild46cat" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [单链表-Python操作](https://blog.csdn.net/qq_14959801/article/details/52988861" \o "单链表-Python操作" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 1万+](https://blog.csdn.net/qq_14959801/article/details/52988861" \o "单链表-Python操作" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[链表是数据结构中最基本常用的，C++语言中单链表是利用指针操作实现的，python作为面向对象编程的，可以使用创建一个Node类来实现链表，利用类的属性引用来代替指针操作。n 下面我们创建了一个...](https://blog.csdn.net/qq_14959801/article/details/52988861" \o "单链表-Python操作" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 令狐公子的博客](https://blog.csdn.net/qq_14959801" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

#### [Spring Boot集成持久化Quartz定时任务管理和界面展示](https://blog.csdn.net/u012907049/article/details/73801122" \o "Spring Boot集成持久化Quartz定时任务管理和界面展示" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[阅读数 8万+](https://blog.csdn.net/u012907049/article/details/73801122" \o "Spring Boot集成持久化Quartz定时任务管理和界面展示" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[前言本文是对之前的一篇文章Spring+SpringMVC+mybatis+Quartz整合代码部分做的一个修改和补充， 其中最大的变化就是后台框架变成了Spring Boot。本工程所用到的技术或工...](https://blog.csdn.net/u012907049/article/details/73801122" \o "Spring Boot集成持久化Quartz定时任务管理和界面展示" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博文[来自： 天降风云的博客](https://blog.csdn.net/u012907049" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[设计制作学习](https://edu.csdn.net/combos/o363_l0_t" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[机器学习教程](https://edu.csdn.net/courses/o5329_s5330_k" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[Objective-C培训](https://edu.csdn.net/courses/o280_s351_k" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[交互设计视频教程](https://edu.csdn.net/combos/o7115_s388_l0_t" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[颜色模型](https://edu.csdn.net/course/play/5599/104252" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[mysql关联查询两次本表](https://www.csdn.net/gather_24/MtTaEg3sMDM5MS1ibG9n.html" \t "https://blog.csdn.net/shawngucas/article/details/_blank) [native底部 react](https://www.csdn.net/gather_10/MtjaIg3sMTUzMy1kb3dubG9hZAO0O0OO0O0O.html" \t "https://blog.csdn.net/shawngucas/article/details/_blank) [extjs glyph 图标](https://www.csdn.net/gather_1b/Ntzagg1sOTU3LWRvd25sb2Fk.html" \t "https://blog.csdn.net/shawngucas/article/details/_blank) [设计培训中心](https://www.csdn.net/gather_4a/MtjaEg5sMDUtZWR1.html" \t "https://blog.csdn.net/shawngucas/article/details/_blank) [设计培训网](https://www.csdn.net/gather_4a/MtjaMg0sNDYtZWR1.html" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[IMG_292](https://blog.csdn.net/shawngucas)

**[shawngucas](https://blog.csdn.net/shawngucas" \t "https://blog.csdn.net/shawngucas/article/details/_blank)**

关注

[原创](https://blog.csdn.net/shawngucas?t=1)

**[4](https://blog.csdn.net/shawngucas?t=1)**

粉丝

**33**

喜欢

**27**

评论

**31**

等级：

访问：

6万+

积分：

430

排名：

16万+

### 最新文章

* [Visual Studio 调试小技巧](https://blog.csdn.net/shawngucas/article/details/6619920" \t "https://blog.csdn.net/shawngucas/article/details/_blank)
* [调试术语](https://blog.csdn.net/shawngucas/article/details/6619840" \t "https://blog.csdn.net/shawngucas/article/details/_blank)
* [Vs 2008 使用技巧系列](https://blog.csdn.net/shawngucas/article/details/6617590" \t "https://blog.csdn.net/shawngucas/article/details/_blank)
* [vs2008 工程配置--路径设置](https://blog.csdn.net/shawngucas/article/details/6605665" \t "https://blog.csdn.net/shawngucas/article/details/_blank)
* [socket关闭过程](https://blog.csdn.net/shawngucas/article/details/6587633" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

### 个人分类

* [C/C++1篇](https://blog.csdn.net/shawngucas/article/category/838373)
* [Tips4篇](https://blog.csdn.net/shawngucas/article/category/839659)
* [调试2篇](https://blog.csdn.net/shawngucas/article/category/849983)

### 归档

* [2011年7月6篇](https://blog.csdn.net/shawngucas/article/month/2011/07)
* [2011年6月1篇](https://blog.csdn.net/shawngucas/article/month/2011/06)

### 热门文章

* [内存池设计与实现](https://blog.csdn.net/shawngucas/article/details/6574863" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

阅读数 61330

* [vs2008 工程配置--路径设置](https://blog.csdn.net/shawngucas/article/details/6605665" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

阅读数 3724

* [socket关闭过程](https://blog.csdn.net/shawngucas/article/details/6587633" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

阅读数 1329

* [关于lib和dll](https://blog.csdn.net/shawngucas/article/details/6579677" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

阅读数 1221

* [Visual Studio 调试小技巧](https://blog.csdn.net/shawngucas/article/details/6619920" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

阅读数 821

### 最新评论

* [内存池设计与实现](https://blog.csdn.net/shawngucas/article/details/6574863" \l "comments" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[God\_luck：](https://my.csdn.net/God_luck" \t "https://blog.csdn.net/shawngucas/article/details/_blank)楼主你的性能图和内存池运行图是用什么神器画的？

* [内存池设计与实现](https://blog.csdn.net/shawngucas/article/details/6574863" \l "comments" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[luckywll：](https://my.csdn.net/luckywll" \t "https://blog.csdn.net/shawngucas/article/details/_blank)有两个错误 1.判断block块是否为最后一段 2.有一处地方start写成了count

* [内存池设计与实现](https://blog.csdn.net/shawngucas/article/details/6574863" \l "comments" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[saodilaoshu：](https://my.csdn.net/saodilaoshu" \t "https://blog.csdn.net/shawngucas/article/details/_blank)博主文中所写的内存管理链表节点里的size,指代的是free+used的总和吗?这里感觉有漏洞

* [内存池设计与实现](https://blog.csdn.net/shawngucas/article/details/6574863" \l "comments" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[kelvin\_yin：](https://my.csdn.net/kelvin_yin" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[reply]erxiaodao[/reply] 数据输入excel，生成的图表

* [内存池设计与实现](https://blog.csdn.net/shawngucas/article/details/6574863" \l "comments" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[u014338577：](https://my.csdn.net/u014338577" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[reply]CAICHAO1234[/reply]



程序人生

[](https://blog.csdn.net/csdnnews?utm_source=csdn_footer)

CSDN资讯

[kefu@csdn.net](mailto:webmaster@csdn.net" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[QQ客服](http://wpa.b.qq.com/cgi/wpa.php?ln=1&key=XzgwMDE4MDEwNl80ODc3MzVfODAwMTgwMTA2XzJf" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[客服论坛](http://bbs.csdn.net/forums/Service" \t "https://blog.csdn.net/shawngucas/article/details/_blank)400-660-0108

工作时间 8:30-22:00

**[关于我们](https://www.csdn.net/company/index.html" \l "about" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[招聘](https://www.csdn.net/company/index.html" \l "recruit" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[广告服务](https://www.csdn.net/company/index.html" \l "contact" \t "https://blog.csdn.net/shawngucas/article/details/_blank)** **[网站地图](https://www.csdn.net/gather/A" \t "https://blog.csdn.net/shawngucas/article/details/_blank)**

[百度提供站内搜索](https://zn.baidu.com/cse/home/index" \t "https://blog.csdn.net/shawngucas/article/details/_blank) [京ICP备19004658号](http://beian.miit.gov.cn/publish/query/indexFirst.action" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[京公网安备11010502030143](http://www.beian.gov.cn/portal/registerSystemInfo?recordcode=11010502030143" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

©1999-2019 北京创新乐知网络技术有限公司

[经营性网站备案信息](https://csdnimg.cn/cdn/content-toolbar/csdn-ICP.png" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[网络110报警服务](http://www.cyberpolice.cn/" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[北京互联网违法和不良信息举报中心](http://www.bjjubao.org/" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

[中国互联网举报中心](http://www.12377.cn/" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[家长监护](https://download.csdn.net/index.php/tutelage/" \t "https://blog.csdn.net/shawngucas/article/details/_blank)[版权申诉](https://blog.csdn.net/blogdevteam/article/details/90369522" \t "https://blog.csdn.net/shawngucas/article/details/_blank)

27

[29](https://blog.csdn.net/shawngucas/article/details/6574863" \l "commentBox" \o "写评论)

[IMG_295VIP  
免广告](https://mall.csdn.net/vip" \t "https://blog.csdn.net/shawngucas/article/details/_blank)IMG_296手机看IMG_297客服IMG_298举报IMG_299返回  
顶部

