

内存池设计与实现

内存池设计与实现

一、前言

作为 C++ 程序员，想必对于内存操作这一块是比较熟悉和操作比较频繁的；

比如申请一个对象，使用 `new`，申请一块内存使用 `malloc` 等等；

但是，往往会有一些困扰烦恼着大家，主要体现在两部分：

- 申请内存后忘记释放，造成内存泄漏
- 内存不能循环使用，造成大量内存碎片

这两个原因会影响我们程序长期平稳的运行，也有可能会导致程序的崩溃；

二、内存池

内存池是池化技术中的一种形式。通常我们在编写程序的时候回使用 `new delete` 这些关键字来向操作系统申请内存，而这样造成的后果就是每次申请内存和释放内存的时

候，都需要和操作系统沟通内存分配，从而为分配所需的

嗨，简悦 高级账户 内置了 原生的 PDF 转换方式，有需要的话请前往开启此功能。

开启

不再提示



存碎片进而降低内存的分配性能，甚至出现内存分配失败的情况。

而内存池就是为了解决这个问题而产生的一种技术。从内存分配的概念上看，内存申请无非就是向内存分配方案要一个指针，当向操作系统申请内存时，操作系统需要进行复杂的内存管理调度之后，才能正确的分配出一个相应的指针。而这个分配的过程中，我们还面临着分配失败的风险。

所以，每一次进行内存分配，就会消耗一次分配内存的时间，设这个时间为 T ，那么进行 n 次分配总共消耗的时间就是 nT ；如果我们一开始就确定好我们可能需要多少内存，那么在最初的时候就分配好这样的一块内存区域，当我们需要内存的时候，直接从这块已经分配好的内存中使用即可，那么总共需要的分配时间仅仅只有 T 。当 n 越大时，节约的时间就越多。

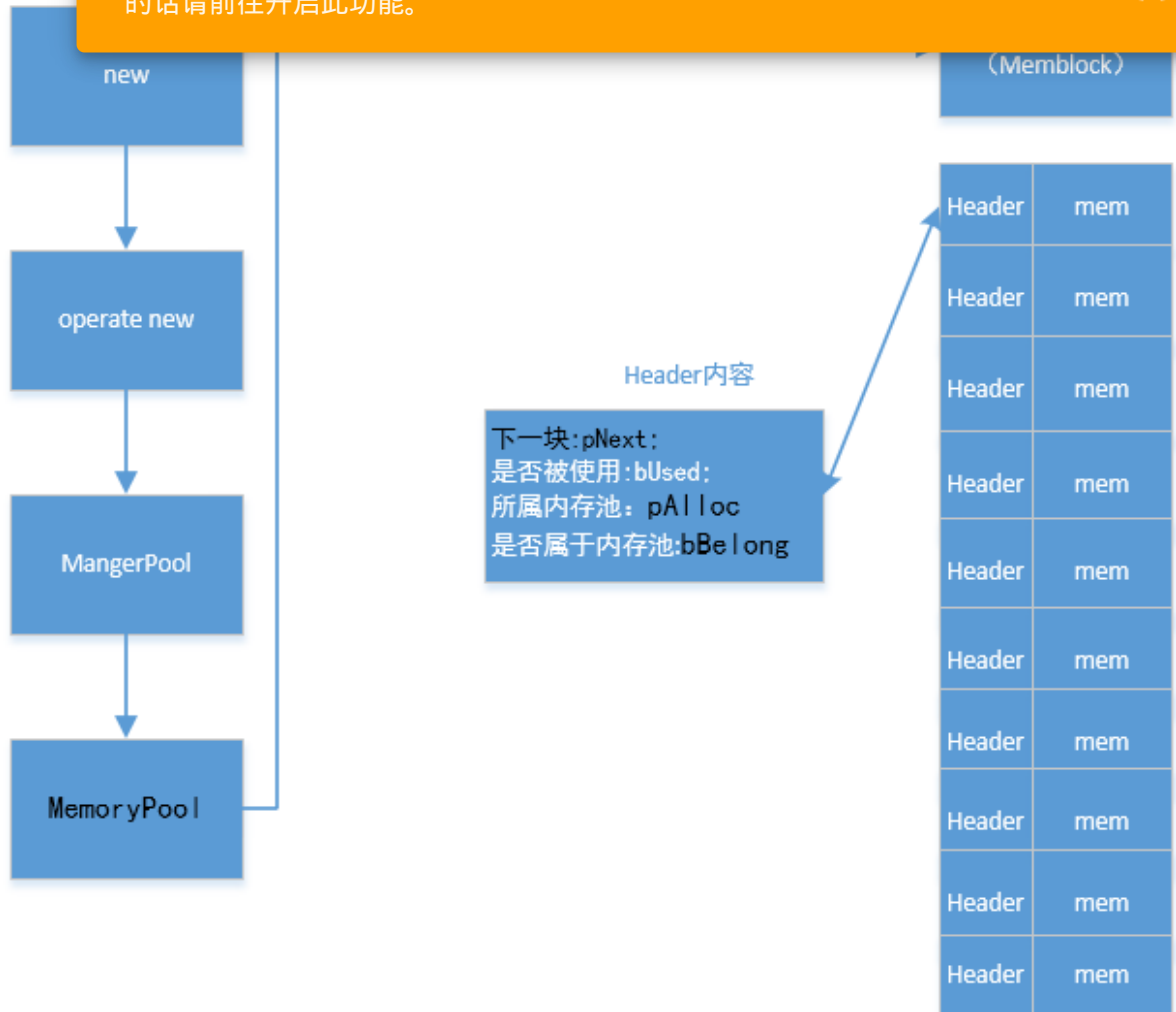
--- 引用来源互联网

三、内存池设计

嗨，简悦 高级账户 内置了 原生的 PDF 转换方式，有需要的话请前往开启此功能。

开启

不再提示



内存池设计实现中主要分为以下几部分：

- 重载 new
- 创建内存节点
- 创建内存池

• 管理内存池

嗨，简悦 高级账户 内置了 原生的 PDF 转换方式，有需要的话请前往开启此功能。

开启

不再提示



下面，比较详细的来说说设计细节：

重载 new 就不说了，直接从内存节点开始；

内存池节点

内存池节点需要包含以下几点元素：

1. 所属池子（ `pMem` ），因为后续在内存池管理中可以直接调用申请内存和释放内存
2. 下一个节点（ `pNext` ），这里主要是使用链表的思路，将所有的内存块关联起来；
3. 节点是否被使用（ `bUsed` ），这里保证每次使用前，该节点是没有被使用的；
4. 是否属于内存池（ `bBelong` ），主要是一般内存池维护的空间都不是特别大，但是用户申请了特别大的内存时，就走正常的申请流程，释放时也就正常释放；

内存池设计

1. 内存首地址 (`_pBuffer`)，也就是第一块内存，这样以后方面寻找后面的内存块；
2. 内存块头 (`_pHeader`)，也就是上面说的内存池节点；
3. 内存块大小 (`_nSize`)，也就是每个节点多大；
4. 节点数 (`_nBlock`)，及时有多少个节点；

这里面需要注意的是，申请内存块的时候，需要加上节点头，但是申请完后返回给客户使用的需要去掉头；但是释放的时候，需要前移到头，不然就会出现异常；

释放内存：

释放内存的时候，将使用过的内存置为 `false`，然后指向头部，将头部作为下一个节点，这样的话，节点每次回收就可以相应的被找到；

内存池管理

内存池创建后，会根据节点大小和个数创建相应的内存池；

内存池管理主要就是根据不同的需求创建不同的内存池，以达到

管理的目的

嗨，简悦 高级账户 内置了 原生了 PDF 转换方式，有需要的话请前往开启此功能。

开启

不再提示



这里主要有一个概念：数组映射

数组映射就是不同的范围内，选择不同的内存池；

添一段代码：

```
void InitArray(int nBegin,int nEnd, MemoryPool*pMemPool) { for (int
```

根据范围进行绑定；

四、内存池实现

ManagerPool.hpp

```
#ifndef _MEMORYPOOL_HPP_#define _MEMORYPOOL_HPP_#include <iostream>#
```

OperatorMem.hpp

```
#ifndef _OPERATEMEM_HPP_#define _OPERATEMEM_HPP_#include <iostream>#
```

mian.cpp

```
#include "OperateMem.hpp"using namespace std;int main(){ char* p = n
```

想了解学习后台服务器开发知识，请关注微信公众号：简悦

号：=====

嗨，简悦 高级账户 内置了 原生的 PDF 转换方式，有需要的话请前往开启此功能。

开启

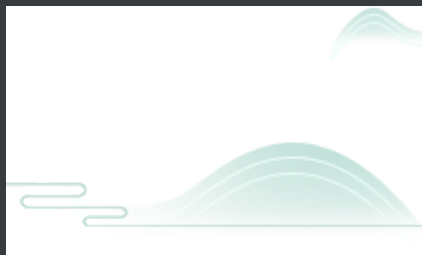
不再提示



往期精彩文章

- [linux 内存分配管理](#)
- [你必须知道的 linux 网络](#)
- [linux 服务器程序性能调试](#)

扫码 CPP 后台服务器开发



嗨，简悦 高级账户 内置了 原生了 PDF 转换方式，有需要的话请前往开启此功能。

开启

不再提示



点击蓝字



转载是一种动力 分享是一种美德

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎^{beta}，[点击查看详细说明](#)

