**高级数据库系统课程实验**

Storage and Buffer Manager

实验报告

姓 名： 李 爽

学 号： SA19225033

班 级： 软设四班

实验日期： 2020年7月7日

# 一、实验背景

为了加深对数据库课程学习的理解，进行了此实验，实现了一个简单的存储和缓存管理器。在这个实验中涉及了缓存技术、应用Hash散列技术组织缓冲区、文件的存储结构等各个方面的知识。

# 二、实验环境

开发环境：操作系统：Windows 10（64）

开发语言：C++

集成开发环境：Visual Studio 2017，Windows SDK 10.0.17134.0

# 三、实验内容

1、实现对此磁盘中的数据进行读写。

2、实现应用Hash散列技术及双向链表组织缓冲区。

3、使用文件老师所给数据文件验证系统。

4、更改缓存器大小并对产生现象进行分析。

# 三、设计思路与实现

## 1、实现对此磁盘中的数据进行读写。

文件存储格式如下：

每条记录长度为316字节

页面大小设定为4KB

则每个磁盘页可放入12条记录

页面数为50000

文件记录总大小

每页大小

总页数

4 byte

4 byte

4 byte

4 byte

记录页号

4 byte

4 byte

4 byte

存储页 50000

4 096

byte

... …

4096

byte

4096

byte

4096

byte

图1.1 分页存储设计



图1.2 数据库文件建立函数

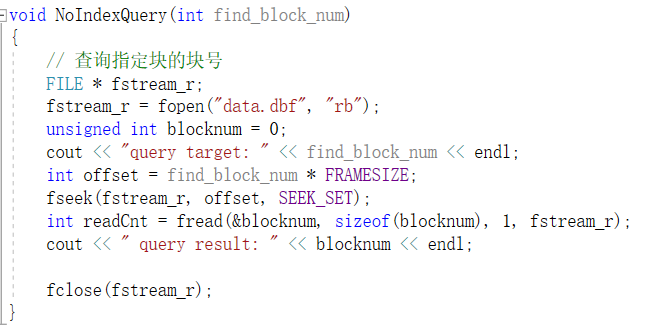


图1.3 数据库文件访问测试

首先建立数据库，名称为data.dbf。数据库只需建立一次，得到数据库文件后可以将CreateBlockWRTest(int num\_blocks)函数的调用注释掉，并备份测试数据库。

## 2、缓存管理器设计

**2.1确定缓存区的结构**

缓存页大小设为4KB。

缓存区中frame的数量初始定为1024。

**2.2缓存控制块设计（BCB）**

一般地，1个BCB维护着1个磁盘页在缓存（内存）中的信息，包括磁盘页号、此块号对应的缓存页号、用户占用计数、时间戳、脏位。BCB结构体如图2.2。在主流的实现方式中，BCB作为数组或者链表的结点，组成便于管理的数据结构，提供快速的查找、更新、置换等功能。

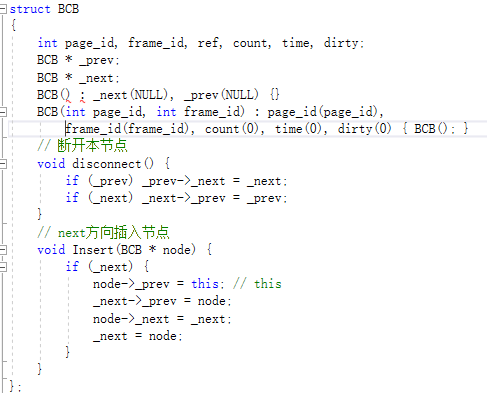


图2.2 BCB结构设计

**2.3基于Hasu与双向链表的缓存区**

LRUCache的private成员：

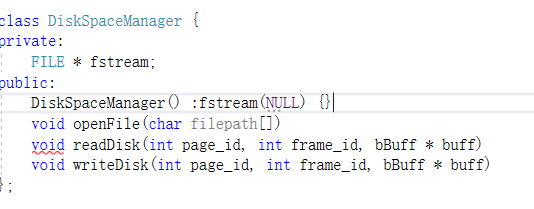


LRUCache的public成员**：**



**2.4 实现磁盘文件操作**

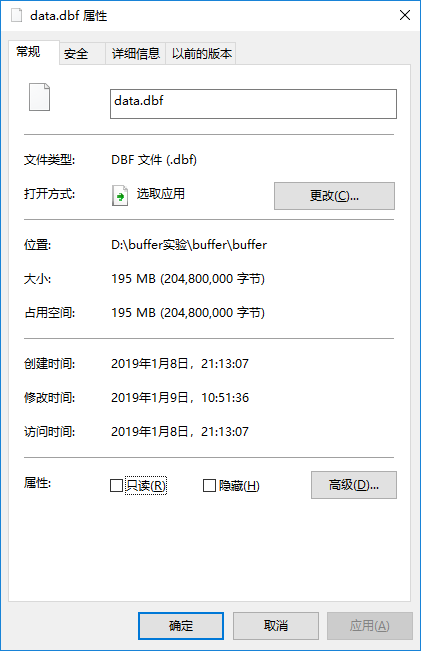
文件的流操作：



# 五、实验结果及分析

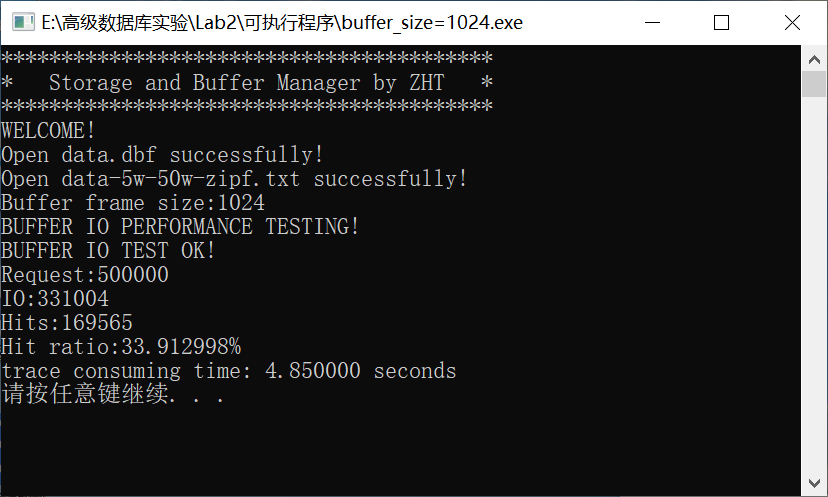
## 1、生成数据库文件

使用CreateBlockWRTest函数生成含50000个磁盘页的数据库文件data.dbf

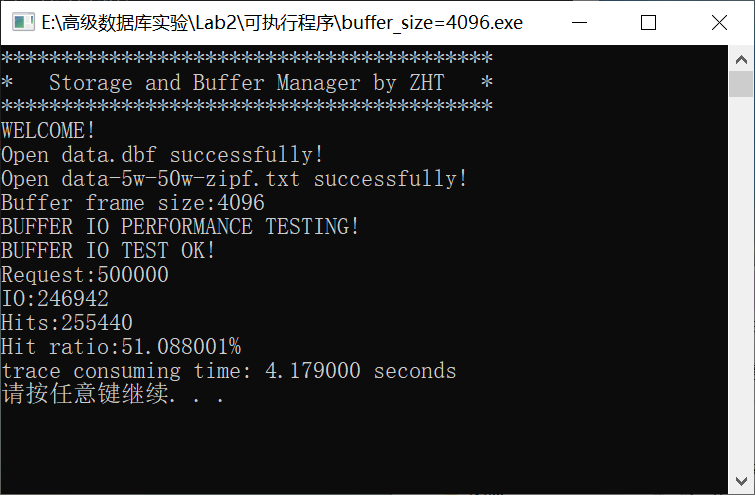


## 2、对不同缓存区大小进行实验

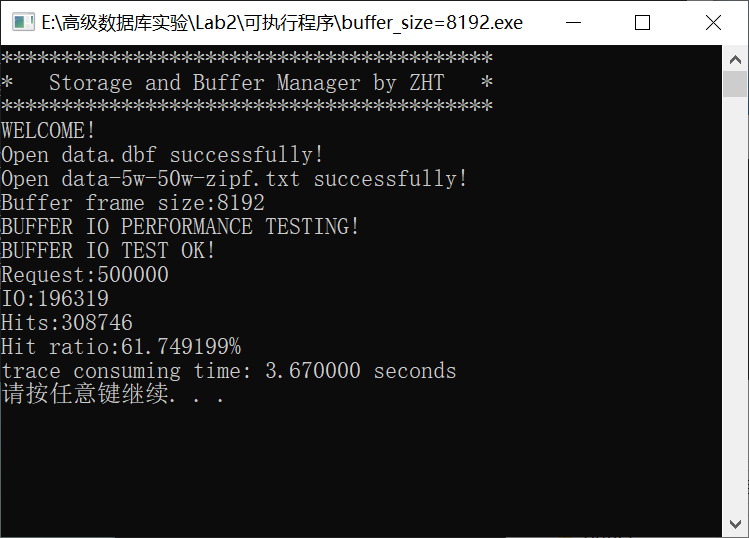
设置BUFFSIZE=1024进行测试，结果如下图：



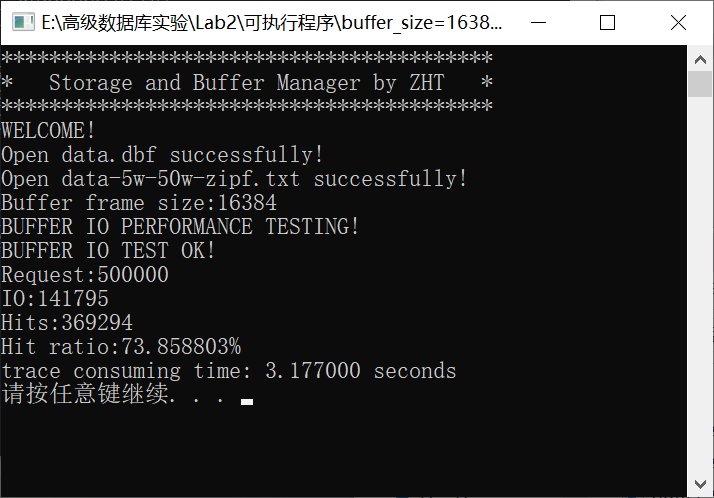
设置 BUFFSIZE=4096进行测试，结果如下图：



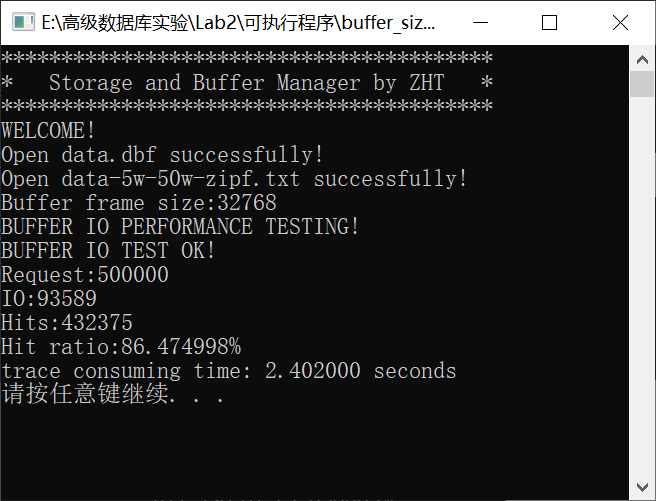
设置 BUFFSIZE=8192进行测试，结果如下图：



设置 BUFFSIZE=16384进行测试，结果如下图：



设置 BUFFSIZE=32768进行测试，结果如下图：



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BUFFSIZE | IO | Hits | Hit ratio | time/s | memory/MB |
| 1024 | 331004 | 169565 | 33.91% | 4.85 | 14 |
| 4096 | 246942 | 255440 | 51.09% | 4.179 | 28 |
| 8192 | 196319 | 308746 | 61.75% | 3.67 | 46 |
| 16384 | 141795 | 369294 | 73.86% | 3.177 | 82 |
| 32768 | 93589 | 432375 | 86.48% | 2.402 | 154 |

1. 观察表可得，当缓冲区越大，磁盘IO次数越少、命中数越高，但是文件访问时间会上升，等到缓冲区上升到一定大小时，访问时间又开始减小。说明，为了提升文件访问效率不，能盲目增加缓冲区的大小。
2. 随着BUFFSIZE的增大，IO数量将下降到一个稳定值，命中数、命中率页上升到了一个稳定值，即BUFFSIZE继续增大将不会继改善IO性能。
3. 所以要合理的设计缓冲区大小。过度增大缓冲区，一方面占用了更多主存储器的宝贵资源；另一方面，在缓冲区置换页面过程中，遍历帧占用时间也会随之增大，甚至会带来负作用。

# 六、源码

实验源码在压缩包中，也包括了可执行文件。