# 高级数据库技术实验二实验报告

姓名：杨扬 学号：SA13226226

1. 实验目的

本次实验是设计并且实现一个数据库缓冲区以及存储管理器。在整个数据库系统实现的五层架构中，缓冲区与存储管理器是处于第一层以及第二层，是上层记录管理器、执行引擎和查询编译器的低层支持，位于整个数据库的底层。

学习并且掌握数据库缓冲区以及存储管理器的设计对今后继续学习数据库底层结构、数据库性能分析，进而参与设计下一代数据库的设计有很大帮助。

1. 实验环境

|  |  |  |  |
| --- | --- | --- | --- |
| 硬件环境 | | 软件环境 | |
| CPU | Intel Pentium E5300 | 操作系统 | Windows 7 Ultimate |
| 内存 | DDR2 800MHz 4GB | 开发工具 | Visual Studio Ultimate 2012 |
| 硬盘 | Seagate 7200r/m 320G \* 2 | 开发语言 | C++ |
| 磁盘阵列 | RAID 0 |  |  |

1. 实验设计
2. 数据库缓冲区设计

* 相关常量设定

缓冲区帧的容量：4096

缓冲区帧的数量：1024（默认值）

* 主要数据结构

缓冲区LRU链表节点：LRUElement。该数据结构包含两个bool型的变量isHead和isTrail，分别表示该节点是否为该节点的头结点或者尾节点。int型变量frameID表示该LRU节点所指代的缓冲区的帧的帧号。int型变量pageID表示该LRU节点所指代的缓冲区的帧所对应的块的块号。另外设定两个节点类型指针用来遍历LRU链表。

isHead

isTrail

frameID

pageID

More Recent

Less Recent

LRU链表：LRU链表是执行LRU替换算法所必需的数据结构。在本次实现中，为了提高代码的可重用性，该链表被定义为具有头节点和尾节点的双线链表。除了头结点和尾节点有特定的bool型标志位标识以外，其他各个数据节点均为LRUElement的普通数据模式。

**……**

Head

LRUElemen

Trail

LRUElemen

缓冲区控制块：BCB。其主要数据成员与LRUElement基本相同，只是额外增添了一个bool型变量isWrite作为数据是否被修改的标志位，如若数据在程序执行过程中被修改，那么在执行缺页调度的时候需要将该BCB所对应的数据块写回磁盘。

isHead

isTrail

frameID

pageID

isWrite

next

previous

P2F哈希映射表：P2F哈希映射表是一个与缓冲区的帧的数量一致的静态哈希数组，每一个数组指代一个特定的哈希桶，并且存放指向该桶头结点地址的指针。在桶内部实现的是以BCB为节点的，带有头结点和尾节点的双向链表。在哈希映射表内部采用静态数组以方便随机存取，在哈希桶内部采用双向链表结构以方便增删改查的操作。假设当前测试缓冲区的帧的数量为N：

……

 BCB

Hash\_P2F

[0]

Hash\_P2F

[1]

Hash\_P2F

[2]

Hash\_P2F

[N-2]

Hash\_P2F

[N-1]

Head

Head

Head

Head

Head

Trail

BCB

BCB

Trail

Trail

BCB

BCB

Trail

BCB

BCB

Trail

BCB

BCB

BCB

1. 存储器管理器设计

* 相关常量设定

磁盘块的容量：4096

磁盘块的数量：600000

1. 设计实现
2. 运行结果及分析