哈尔滨工业大学

<<数据库系统>>

实验报告三

(2023 年度春季学期)

姓名:	徐柯炎
学号:	2021110683
学院:	计算学部
教师:	程思瑶

实验三

一、实验目的

1. 完成 BadgerDB Project：在提供的存储管理器之上实现一个缓冲区管理器。
2. 掌握数据库系统的物理实现方式，特别是存储方式以及缓冲区管理。

二、实验环境

该实验在虚拟机下实现。虚拟机操作系统为 ubuntu20.04，为 linux 系统。

三、实验过程及结果

由 project2 以及相关文档可知，本实验主要需要完成 buffer.cpp 文件的编写，具体实现过程如下：

1. void advanceClock()：

作用：旋转时针，使其指向缓冲池中的下一个页框。

实现：将 clockhand（当前缓冲池时钟指向的页框）加一取模（缓冲池大小）即可。

2. void allocBuf(FrameId& frame)：

作用：利用时钟算法，为需要读入缓冲池的数据分配页框。

实现：参照 project2 中的算法（如下图所示），可分为如下几步：

(1) 旋转指针；

(2) 检查 valid 位：

① 若 valid 位为 0，表示该页框未被分配，于是分配此页框即可；

② 若 valid 位为 1，表示该页框已被分配，继续检查其它条件；

(3) 检查 refbit 位：

① 若 refbit 位为 0，相当于该页框中的数据近期未被使用，继续检查其它条件；

② 若 refbit 位为 1，相当于该页框中的数据近期被使用过，进入下一个循环；

(4) 检查 pincnt：

① 若 pincnt 大于 0，表示该页框被锁定了，于是将 cnt（缓冲池中锁定的页框个数）加一，若 cnt 等于缓冲池中页框的总数，说明所有页框都无法使用，这时会出现一个异常处理；

② 若 pincnt 等于 0，继续检查其它条件；

(5) 检查 dirty 位：

① 若 dirty 位为 1，要将数据先写回磁盘然后再进行分配页框操作；

② 若 dirty 位为 0，直接分配页框即可。

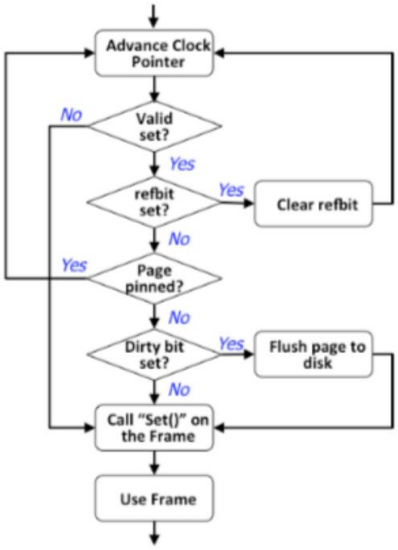


Figure 2: The Clock Replacement Algorithm

3. void readPage(File* file, const PageId PageNo, Page*& page)：

作用：给定文件以及该文件需要读取的页号，通过 page 返回该页在缓冲池中的位置。

实现：分为以下几步：

(1) 在哈希表中寻找该页；

① 若在哈希表中找到该页，通过 frame 返回该页在缓冲池的指针，并将该页的 refbit 位置 1，pincnt 加 1；

② 若未在哈希表中找到该页，说明该页还没有读入缓冲池中，于是首先通过 allocBuf 分配页框，接着将该页读入此页框，然后设置该页框中必要的位，最后将该页框插入哈希表即可。

(2) 返回该页在缓冲池中的相对位置加上缓冲池的指针。

4. void unPinPage(File* file, const PageId PageNo, const bool dirty)：

作用：程序不需要使用该文件中的某一页时，将该页的 pincnt 减 1。

实现：分为以下几步：

(1) 通过哈希表找到该页在缓冲池中的位置；

(2) 如果该页本来就没有被锁定，也就是 pincnt 等于 0，于是进行异常处理；否则将该页的 pincnt 值减 1；

(3) 如果该页修改过（dirty 为 1），则将该页的 dirty 位置 1。

5. void allocPage(File* file, PageId& PageNo, Page*& page)：

作用：为文件分配一个新的页框，返回给 page 指针。

实现：分为以下几步：

(1) 利用 allocBuf 在缓冲池中为其分配一个页框；

(2) 为文件分配一个新的页，并将该页的内容放进缓冲池中；

(3) 返回该页在文件中的页号给 pageNo；

(4) 设置缓冲区中页框的当前状态；

(5) 在哈希表中插入该页；
(6) 最后将缓冲区中该页框的指针返回给 page。

6. void flushFile(File* file)：

作用：刷新 file，也就是将 file 在缓冲池中的数据全部写回磁盘。

实现：扫描缓冲池中所有页框，筛选出所有属于 file 的页框，进行以下操作：

(1) 检查有效位：
① 若有效位为 0，说明此页无效，进行相应的异常处理；
② 若有效位为 1，继续执行；
(2) 检查 pincnt：
① 若 pincnt 不为 0，说明还有程序在使用该文件，进行相应的异常处理；
② 若 pincnt 为 0；继续执行；
(3) 检查 dirty 位，若 dirty 位为 1，则将页框写回磁盘，并将 dirty 位置 0。

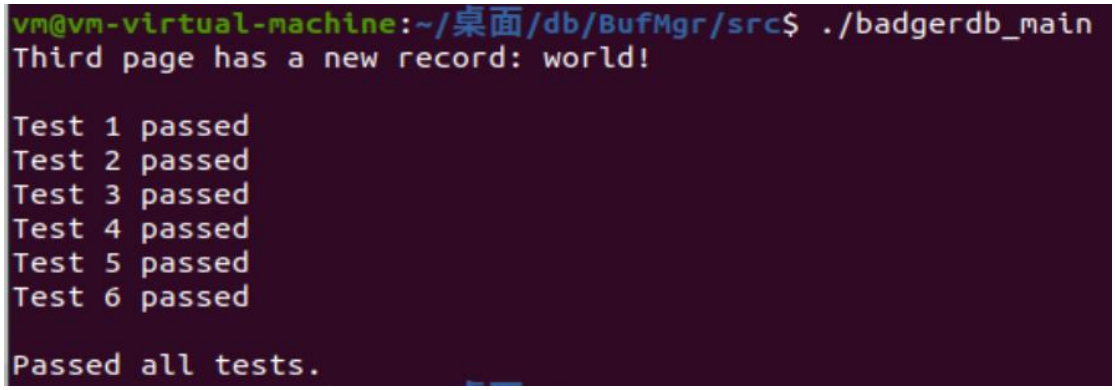
7. void disposePage(File* file, const PageId pageNo)：

作用：删除 file 中的 pageNo 页，并将其从缓冲区中删除（如果存在）。

实现：分为以下几步：

(1) 查找哈希表，如果页面在缓冲池中，则将其移出哈希表，并将缓冲池中的该页框清除。
(2) 在磁盘中删除此页。

实验结果如下：



可以看到，所有的测试都通过了。

四、实验心得

通过这次实验，我掌握了时钟算法的基本流程，对数据库缓冲池管理这一部分的内容有了更加深刻的理解，受益良多。