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:	符兴
学号:	7203610316
学院:	计算学部
教师:	李东博

实验三 Project2

一、实验目的

掌握缓冲管理器的工作原理以及工作流程，并使用 C++面向对象程序设计方法实现缓冲区管理器。

二、实验环境

Ubuntu20.04、GCC9.4、CMake3.16

三、实验过程及结果

首先介绍 Clock 页面置换算法。Clock 算法采用循环队列，并使用一个引用位来标识该页框最近的使用情况，一个页面首次装入内存时，引用位被置为 True；需要替换页面时，替换引用位为 False 的页面。clock 算法相对于 LRU 算法而言，不需要显式地区存储一个时间戳，并且按照时间戳的先后顺序对整个数据进行调整，因为整体操作代价小；同时，由于其按照某一方向顺序遍历的特定，表针刚刚扫过的可能是最近被使用或被分配的，而表针即将扫到的页框经过了一个扫描周期之后，这在某种程度上可以很快找到合适的页框进行替换页面数据。

在本次实验中，给定了存储管理器，需要我们根据 Clock 替换算法去构建一个缓冲管理器，主要的功能函数有：

(1) void BufMgr::advanceClock();

该函数模拟 Clock 算法中的表针，使其按照顺时针的方向指向下一个页框；在具体实现中只需要对 clockHand 进行加 1 操作，并且使它的值映射到[0,numBufs]即可。

(2) void BufMgr::allocBuf(FrameId &frame);

该函数是 Clock 算法的具体实现，在顺时针遍历的过程中检查每个页框的状态，具体检查过程如下所示：

a. 如果当前页框不存在有效页，则该页框被分配出去装载心得数据；

b. 如果页框的 refBit 为 True 代表该页框最近被使用过，需要将其设置为 False，并调用 advanceClock()函数是表针向前移动；

c. 如果页框的 pinCnt 不为 0，代表该页框正在被使用且未被释放；

d. 如果页框的 pinCnt 为 0，则检查该页框是否为脏页，如果为脏页需要将数据写回磁盘，

然后再将该页框分配出去；

e. 如果当前缓冲管理器的所有页框都被使用，即 pinCnt 不为 0，则扔出缓存已满的异常 BufferExceededException();

(3) void BufMgr::readPage(File *file, const PageId pageNo, Page *&page);

该函数用于读取文件中特定的页数据；程序首先会在缓冲区中查找是否缓存该页数据，如果在缓冲区中未找到则需要调用 allocBuf()函数为数据分配一个页框，然后从磁盘中装载该页数据进入该页框对应的缓冲内存中，并将该页数据信息插入哈希表中；

(4) void BufMgr::unPinPage(File *file, const PageId pageNo, const bool dirty);

该函数用于减少特定帧的 pinCnt；如果 dirty=True，则设置 dirty 位；如果 pinCnt 已经为 0，则扔出异常 PageNotPinnedException()；如果在哈希表查找中没有找到 page，则什么都不做；

(5) BufMgr::flushFile(const File *file);

该函数主要用于将某个文件的数据从缓冲区中全部删除；程序在哈希表中查找属于该文件的页，如果当前存储该页数据的页框有效位为 False，需要扔出 BadBufferException()异常；对于遇到的每个文件，如果页面是脏的，则需要将页面写回，将脏位设置为 false；然后从哈希表中删除该页面，并调用 BufDesc 的 clear()方法，清除页框中相应的信息。如果某些页面 pinCnt 不为 0，即该页被占用且未被释放，则抛出异常 PagePinnedException()；如果遇到无效页，则抛出 BadBufferException 异常。

(6) void BufMgr::allocPage(File *file, PageId &pageNo, Page *&page);

该函数用于为某个文件分配空闲页面并装载进缓冲池中；程序首先为指定页面分配一个空页，然后获取缓冲池帧，并在哈希表中插入条目，并通过 page 参数向调用者返回指向该页分配的缓冲帧指针。

(7) void BufMgr::disposePage(File *file, const PageId PageNo);

该函数用于从文件中删除特定页面；在删除之前，首先会在哈希表中查看该页面是否被缓存进缓冲池中，如果存在，则需要从缓冲池中删除该页数据，并删除哈希表条目；如果不存在，则直接在文件中删除该页面即可。

完成上述的七个功能之后，可以使用 Project2 中提供的测试程序进行验证；在实验的过程中，我手动编写了一个新的测试规则（图中 test7），以下是测试通过的截图：

图 1 实验结果

四、实验心得

1. 通过本次实验，进一步深入理解了 Clock 算法，以及缓冲管理器的工作原理和工作流程；

2. 通过本次实验，掌握 C++面向对象编程，并使用 Doxygen 进行良好地注释。