实验2: ﻿缓冲区管理器

主讲教师: 邹兆年(znzou@hit.edu.cn)

姓名: 李扬名 学号: 1163200511 日期: 2019年05月19日

1. 实验要求

在已提供的存储管理器 API 上实现一个缓冲管理器（基于Clock算法）的功能。

1. ﻿实现方法

~BufMgr：

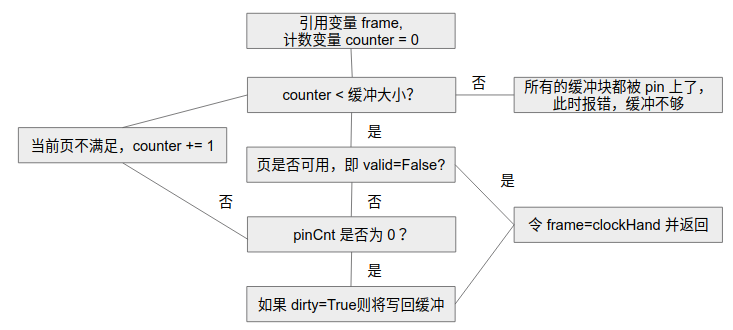
析构函数。要做两件事情。首先将所有的脏页（同时要求是有效的，即 valid=True）写回到磁盘中；其次，要释放所有动态申请（例如hashTable, bufPool等）的内存。

advanceClock：

很简单，将环形缓冲的指针向前更新一步。但注意超过 numBufs 时需要取模。

allocBuf：

很简实验的关键部分，实现了 Clock Algorithm。基本的思路是对每个缓冲块维护三个标记变量，即valid，refbit和pinCnt。基本的逻辑如下图：



readPage：

读取文件中的指定页到缓冲并返回页在缓冲的指针。这里有两种情况，要么页早就在缓冲内存中（可以同步够查hash表获知）了，此时只需直接返回并设置 refbit = truem binCnt += 1（即并引用一次）；要么页还未被缓冲，此时需要调用分配函数allocBuf分配一段内存。

unPinPage：

指定文件对象中某一页 pinCnt -= 1并可以设置脏位dirty=True，如果 pinCnt=0 则报错。

flushFile：

将缓冲块中属于给定文件的全部页（尤其是脏页）全部写会，并清空哈希表和记录表。这里注意有几种报错：其一，页无效但属于文件；其二，在写回页时，pinCnt!=0。

allocPage：

将给定文件的页读入缓冲（分配缓冲，alloBuf），并设置好hash表和描述表。

disposePage：

将指定页从文件中删除。注意同时也需要同步到缓存，此时无论页脏直接设置 valid=True 即可。

3. ﻿总结与心得

本次实验我联系了Clock Algorithm 的实现，而且熟悉了一些C++的工程代码。