数据库系统概论荣誉课程

Lab1- 存储管理实验

2025 年 10 月 23 日

创建私有仓库

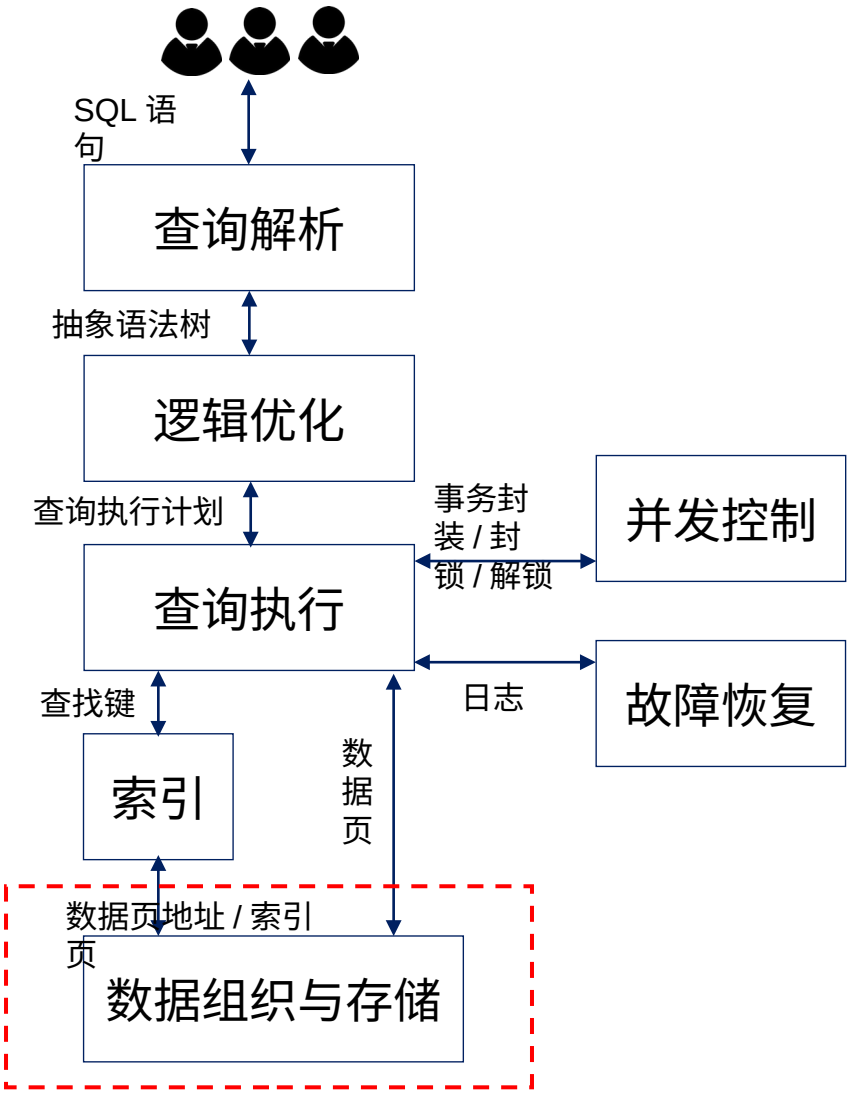
克隆 public 仓库：

git clone --bare https://github.com/ruc-deke/rucbase-lab.git rucbase-public

在 github 上 new 一个 private 仓库

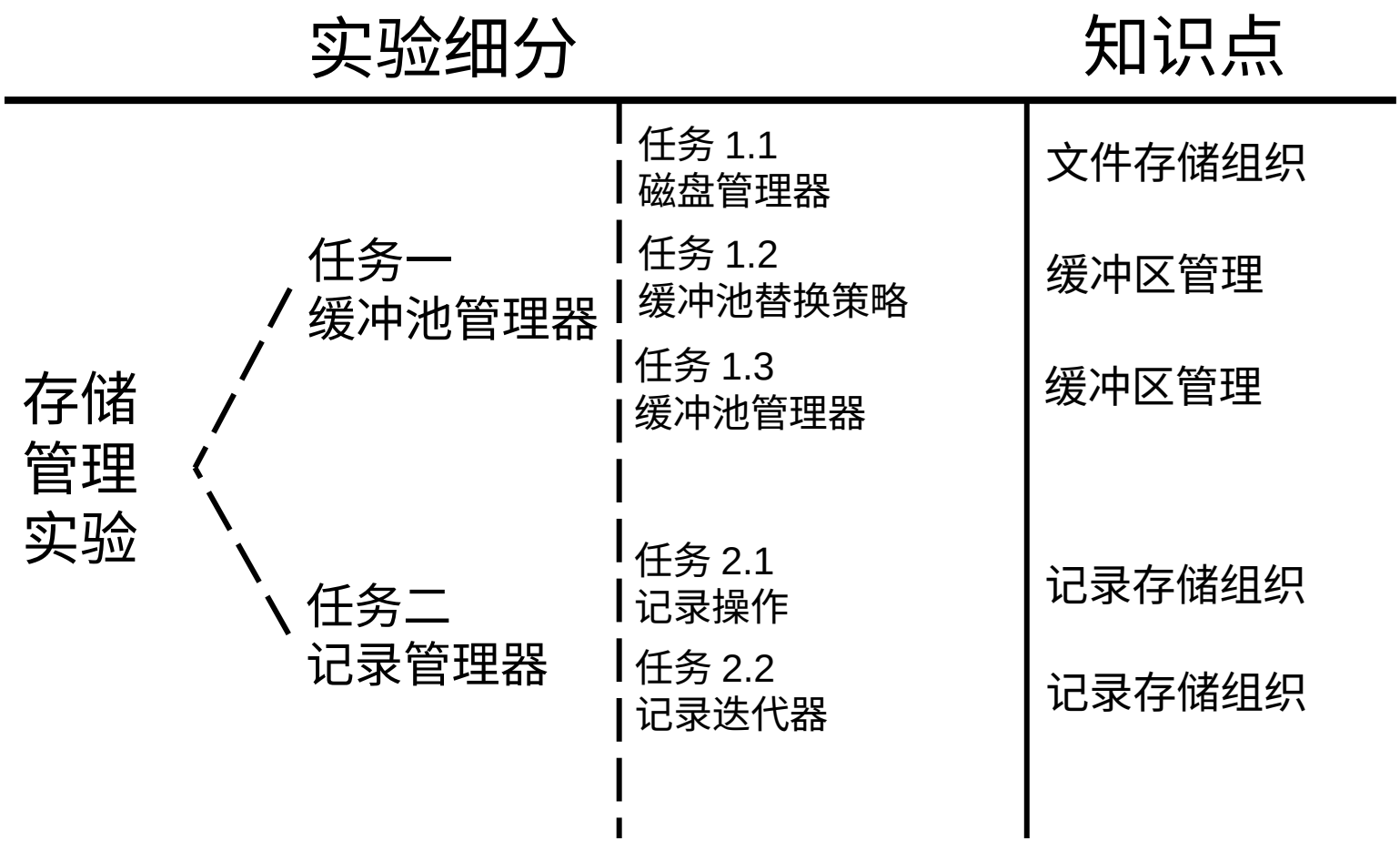
git push https://github.com/xxx/rucbase-private.git master

存储管理知识点对应情况



- 存储管理包含知识点：
  - 基本原理
  - 文件存储组织
  - 元数据存储组织
  - 记录存储组织
  - 缓冲区管理
- 实验任务划分：
  - 任务一包含的内容有文件存储组织、缓冲区管理。
  - 任务二的内容为记录存储组织。

存储管理实验设计


实验细分
知识点
存储管理实验
任务一 缓冲池管理器
任务二 记录管理器
任务 1.1 磁盘管理器
任务 1.2 缓冲池替换策略
任务 1.3 缓冲池管理器
任务 2.1 记录操作
任务 2.2 记录迭代器
文件存储组织
缓冲区管理
缓冲区管理
记录存储组织
记录存储组织

存储管理模块划分

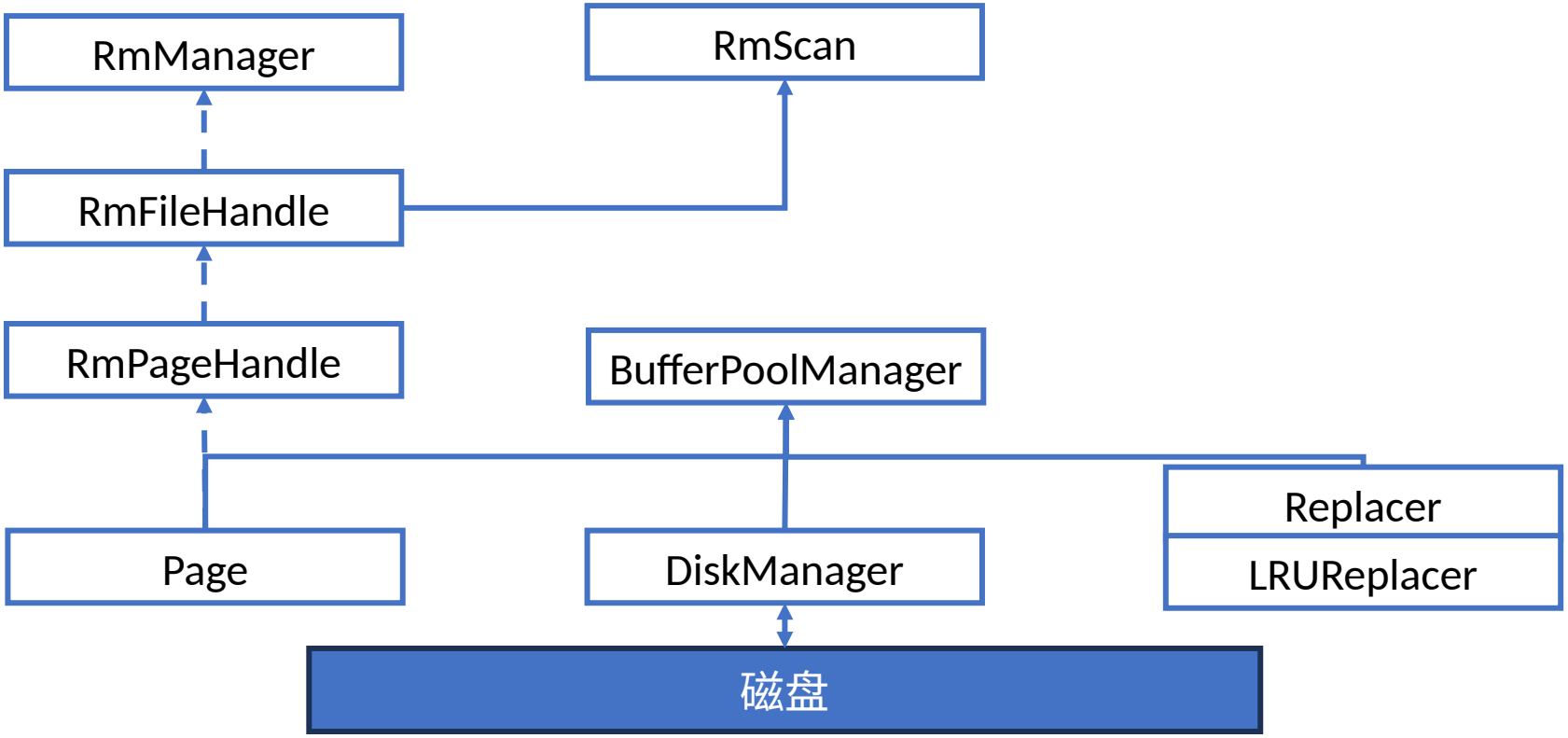
- 在 Rucbase 中，我们将存储管理划分为三个部分：
  - 磁盘管理：其任务是根据上层需要对磁盘进行文件读写操作。一个磁盘文件包含了多个页。
  - 缓冲区管理：缓冲区是内存中用于存储磁盘文件的拷贝的页，缓冲区管理的任务是根据上层需要提供内存页、分配内存页。
  - 记录管理：一个内存页中存储了多条记录，记录管理的任务是根据需要对内存页中的记录进行操作。

存储管理模块与主要类的对应关系

- 在 Rucbase 中，各模块对应的主要类为:
  - 磁盘管理对应的类主要为:
    - DiskManager
  - 缓冲区管理对应的类主要为:
    - BufferPoolManager、 Page、 Replacer(LRUReplacer)
  - 记录管理对应的类主要为:
    - RmPageHandle、 RmFileHandle、 RmManager、 RmScan

存储管理主要类之间的关系介绍

- Rucbase 中存储管理涉及的类的关系：



注： A 指向 B 的实线代表 A 作为 B 的成员变量， A 指向 B 的虚线表示 A 被 B 使用

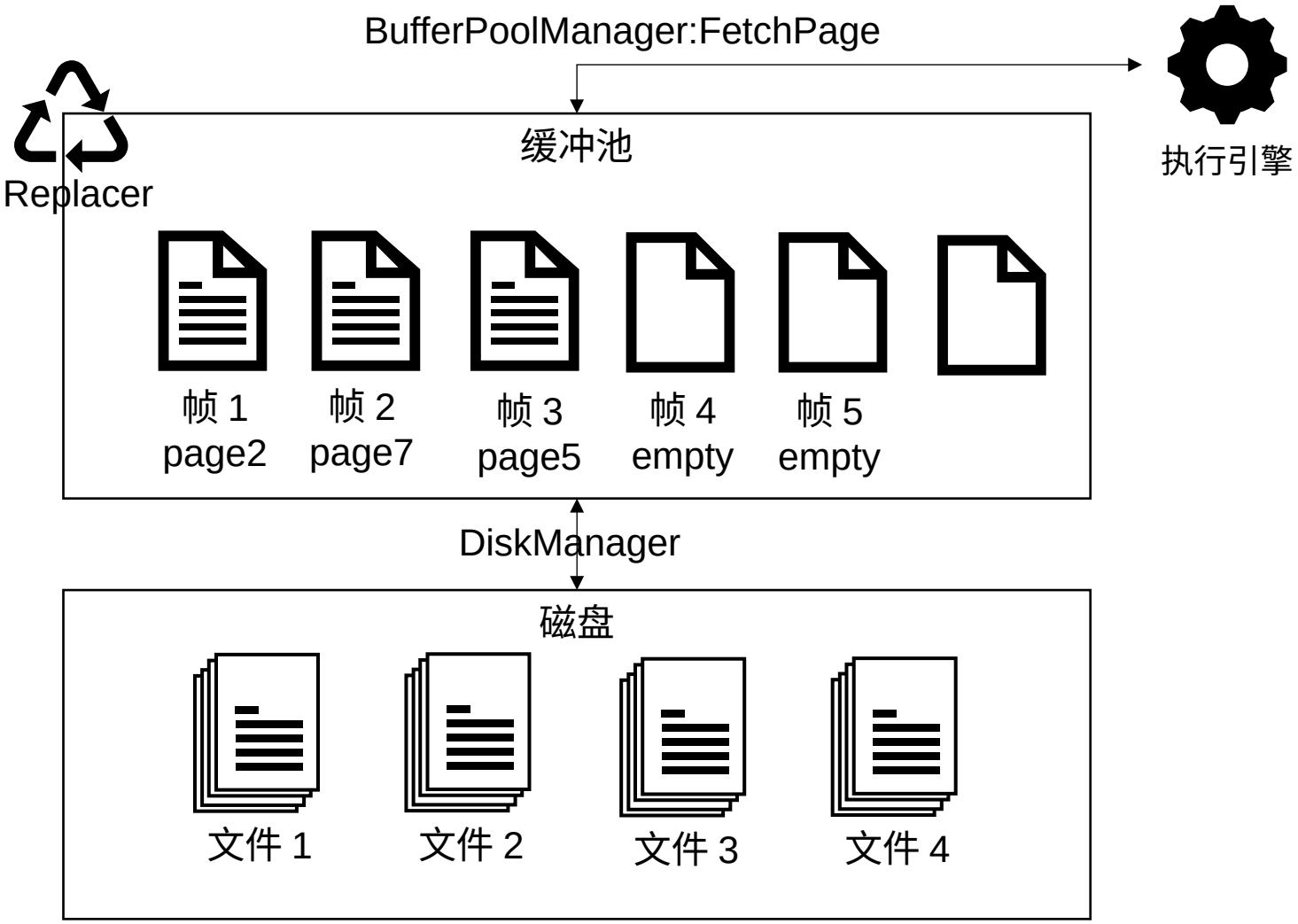
缓冲器管理和磁盘管理的主要类介绍

- BufferPoolManager 用于对内存进行管理，管理的粒度为固定数量的 frame ，一个 frame 可以存储一个 page 。
- DiskManager 用于进行磁盘操作。具体功能包含创建删除文件夹、创建删除打开关闭一个文件、读写一个 page 、为指定文件进行页号管理。
- Page 用于指代一个数据块，该数据块存储在内存的一个 frame 中，其在磁盘的某个文件中有着等量的占用空间。
- Replacer 主要用于选择一个页写回磁盘。 LRUReplacer 是 Replacer 的子类。

记录管理的主要类介绍

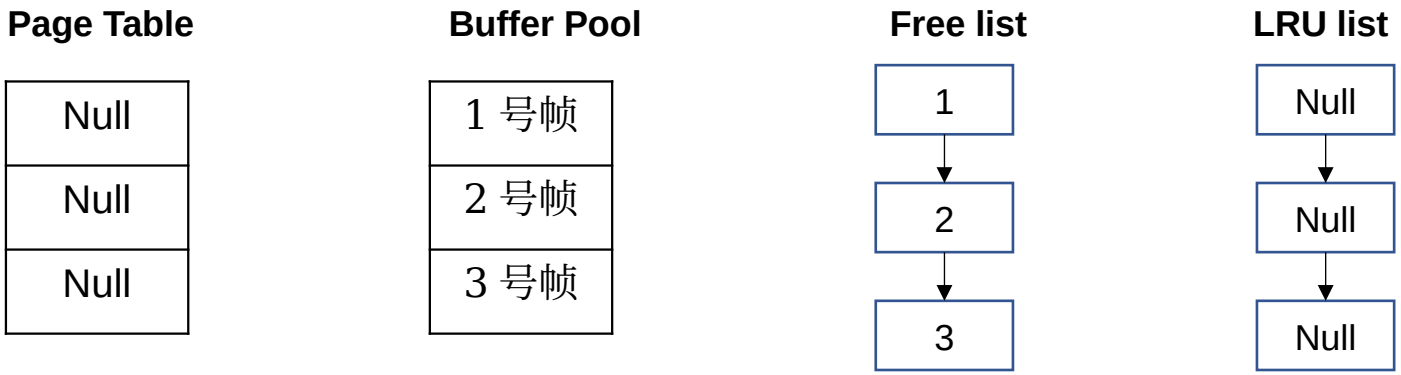
- RmScan 提供给上层查询执行，查询 RecordFile 中下一个 record 。
- RmManager 用于创建、打开、关闭、删除 RecordFile 。
- RmFileHandle 用于对一个已经打开的 RecordFile 进行操作，操作粒度可以为一条记录、也可以为一个 page 。
- RmPageHandle 用于对一个 Page 进行操作，一个文件可以包含多个 page ，一个 page 有多个 slot 可以存储 record 。

缓冲器管理与磁盘管理结构



缓冲器管理的页面置换示例

初始状态



说明:

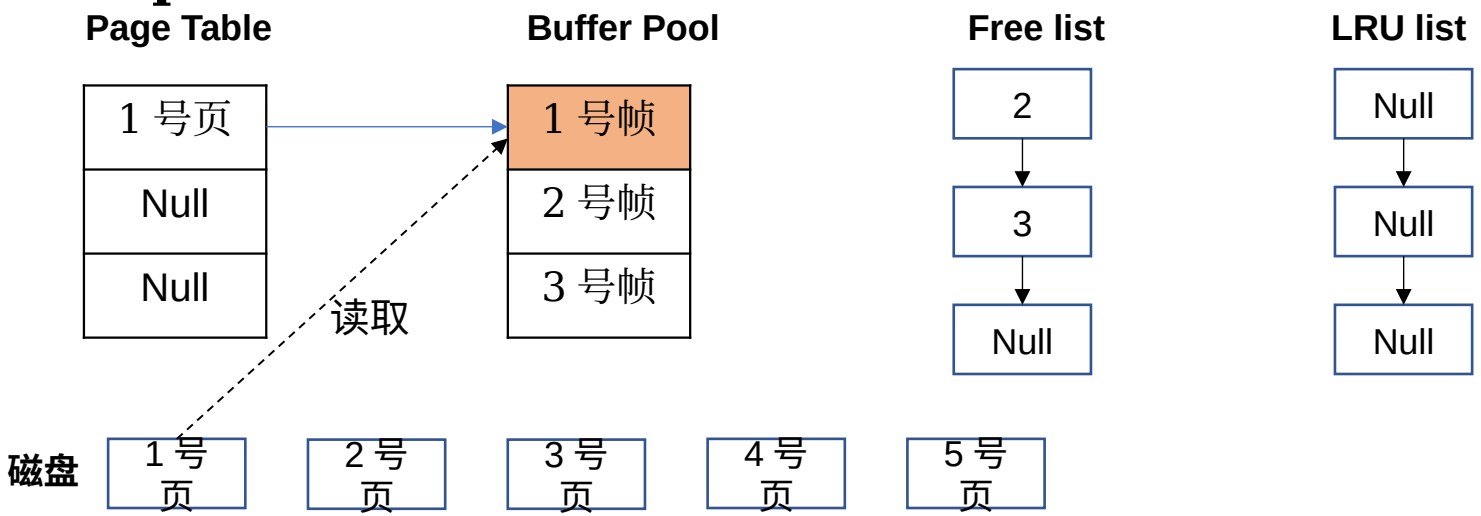
Page Table 为空

Buffer Pool 共三个 frame （全部为空闲）

LRU list 为空

缓冲器管理的页面置换示例

操作 1-1：上层申请使用 1 号页（pin）



说明:

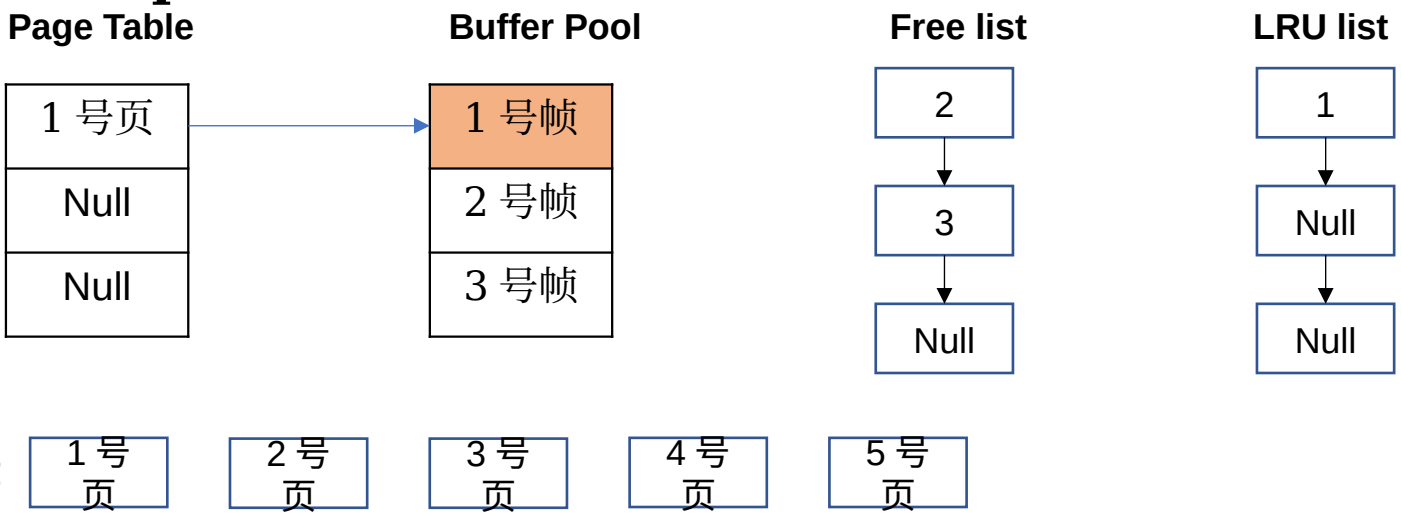
1 号页不在 Page Table 中，即该页不在内存中。

查找 Free list，得到 1 号帧。

将 1 号页的内容读取到 1 号帧。

缓冲器管理的页面置换示例

操作 1-2：上层使用完 1 号页（unpin）

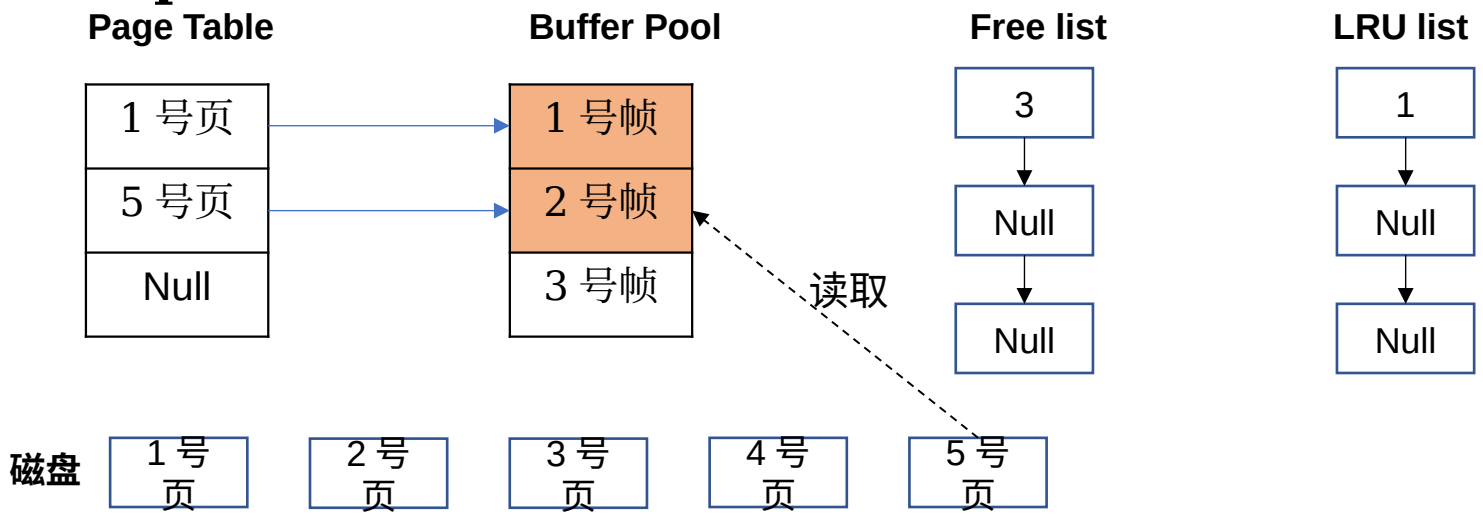


说明:

使用完 1 号页之后，将 1 号页对应的 1 号帧加入到 LRU list 的首部。

缓冲器管理的页面置换示例

操作 2-1：上层申请使用 5 号页（pin）



说明:

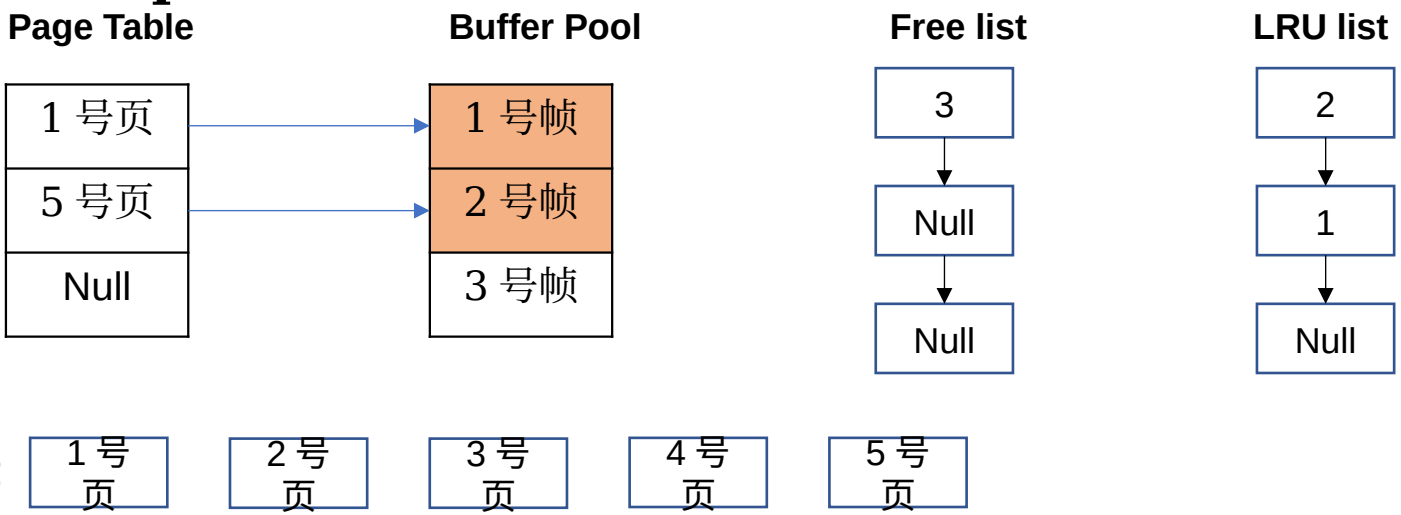
5 号页不在 Page Table 中，即该页不在内存中。

查找 Free list 得到 2 号帧。

将 5 号页内容读取到 2 号帧。

缓冲器管理的页面置换示例

操作 2-2：上层使用完 5 号页（unpin）

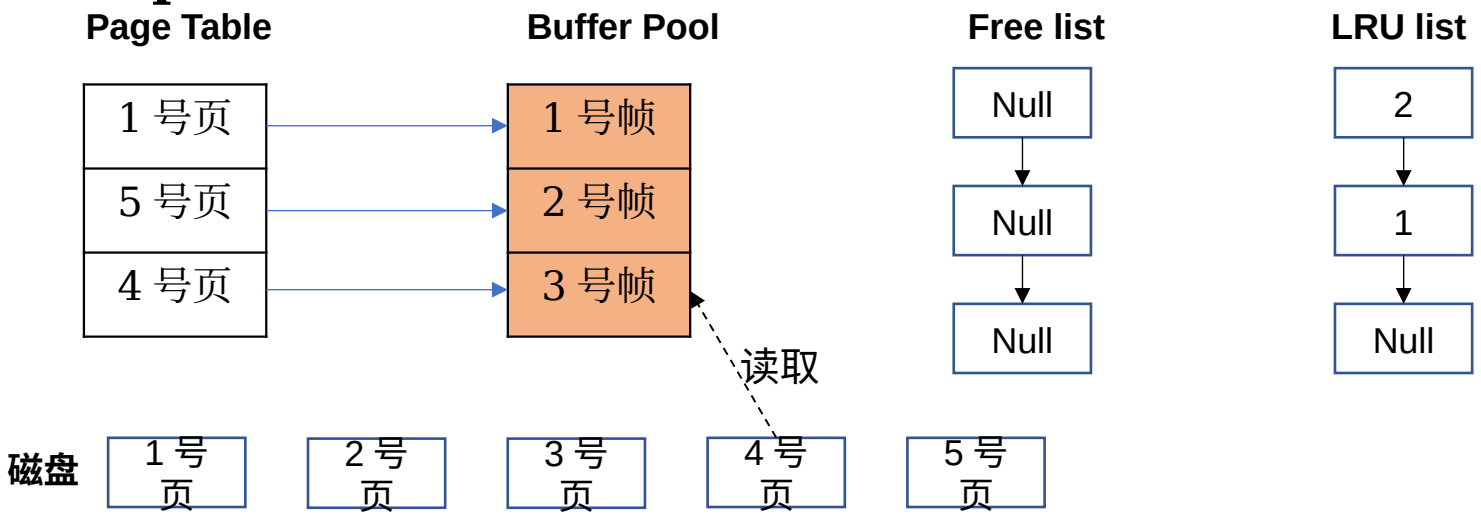


说明:

使用完 5 号页之后，将 5 号页对应的 2 号帧加入到 LRU list 的首部。

缓冲器管理的页面置换示例

操作 3-1：上层申请使用 4 号页（pin）



说明：

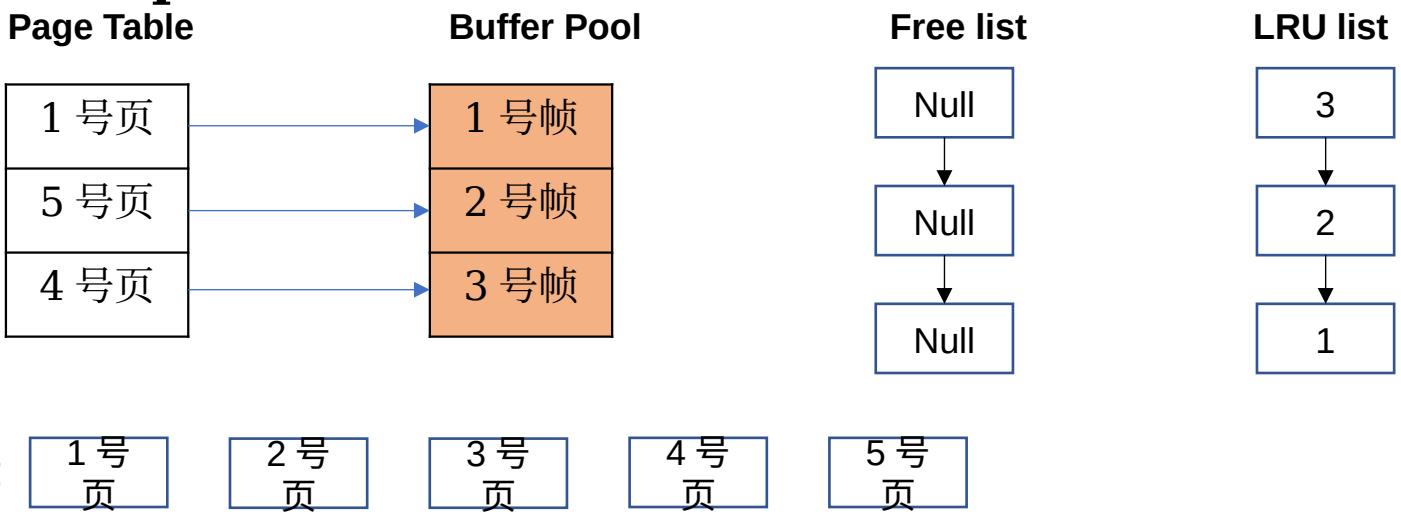
4 号页不在 Page Table 中，即该页不在内存中。

查找 Free list 得到 3 号帧。

将 4 号页的内容读取到 3 号帧。

缓冲器管理的页面置换示例

操作 3-2：上层使用完 4 号页（unpin）

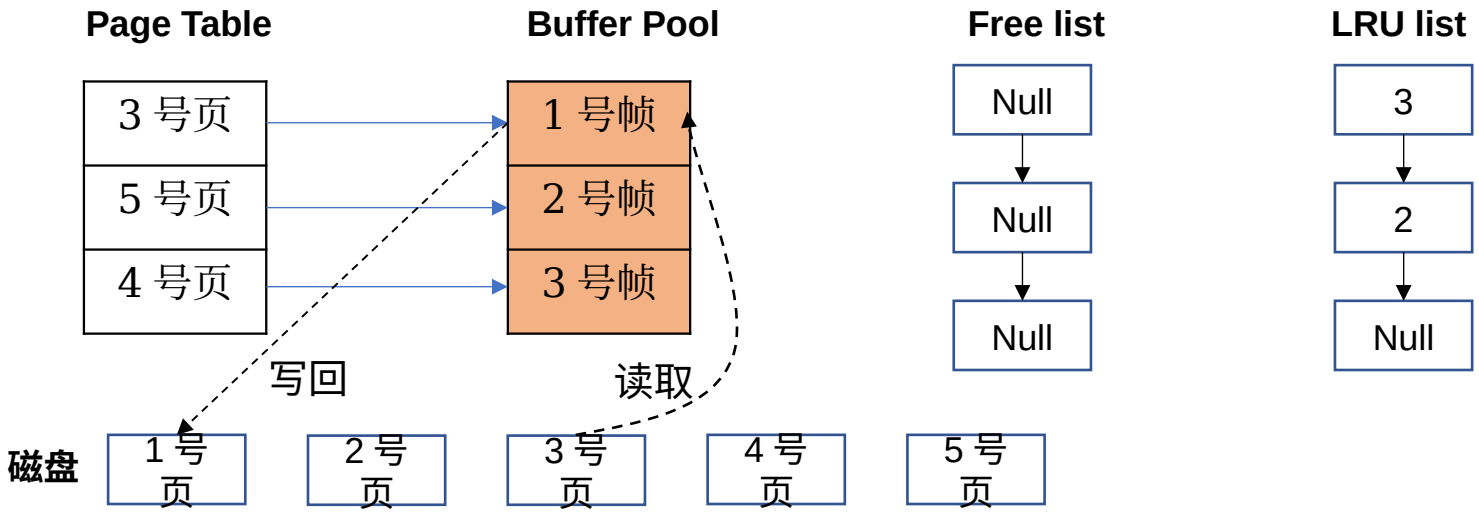


说明:

使用完 4 号页之后，将 4 号页对应的 3 号帧加入到 LRU list 的首部。

缓冲器管理的页面置换示例

操作 4：上层申请使用 3 号页（Victim & Pin）



说明:

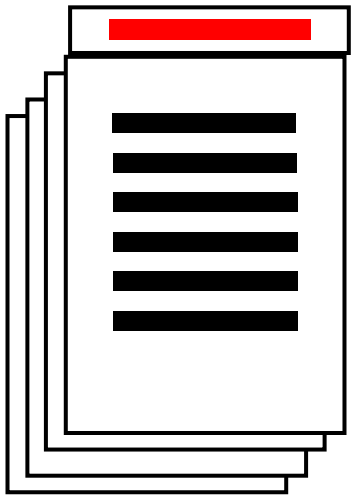
3 号页不在 Page Table 中，即该页不在内存中。

查找 Free list，链表为空。

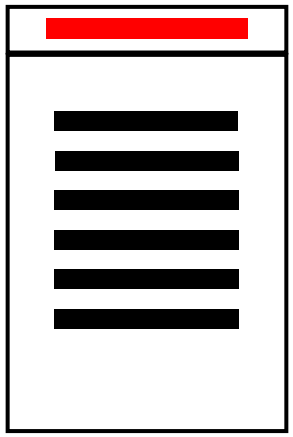
访问 LRU list，由替换策略选择淘汰 1 号帧。

将 1 号帧内容写回磁盘，将 3 号页内容读取到 1

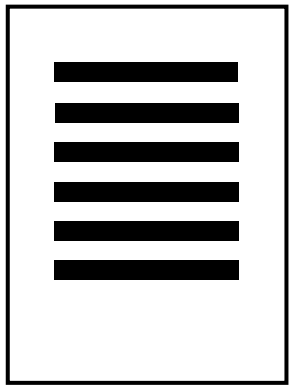
记录管理的数据结构



记录文件
RmFileHandle



页 0
RmFileHdr



页 1、 2…
RmPageHandle

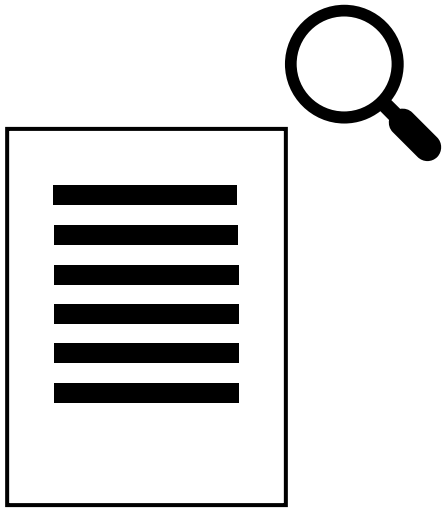
RmFileHdr(record file header)：记录 tuple 大小，文件分配了多少个 page，每个 page 的 tuple 最大容量，当前文件第一个可用的 page 编号，bitmap 大小。

: 记录文件的数据部分

一个记录文件实际只存储一个 RmFileHdr： RmManager 生成一个 RmFileHandle 对象时，直接赋值 RmFileHdr 部分，然后写回磁盘。

所以一个 RmFileHandle 对象有一个 RmFileHdr 结构体成员。

记录管理的数据结构



页
RmPageHandle

offset：占 4 字节，由 OFFSET_PAGE_HDR 定义，存储的是用于日志的 page_lsn 。

RmPageHdr： 记录当前页满了之后下一个可用的 page 的编号；记录当前 page 中存储的 record 个数。

Bitmap： 位示图，表示各个位置是否实际存储了 record 。

Slot: 在 RMDB 中，一个 slot 存储一条记录。

存储管理实验完成流程

- 阅读“ docs/Rucbase-Lab1[存储管理实验文档].md”
- 按照实验文档和 Rucbase 的代码框架中的注释补全各个任务所要求的函数代码。
- 完成一个任务后去跑对应的 Googletest 测试。

- 举例：实验一的任务 1.1 磁盘存储管理器。
  - 通过阅读实验文档和代码框架中的注释补全 DiskManager 的代码。
  - 进入 rucbase_lab 根目录的 build 文件夹，依次键入指令：
    - make disk_manager_test （或者 make –j4 ）
    - ./bin/disk_manager_test
  - 查看 googletest 测试结果。若测试未通过，则根据结果和 src/storage/ 目录下的 disk_manager_test.cpp 来查看未通过的测试的具体逻辑，进行代码修改，修改后再次键入指令” rucbase_lab/build/bin/disk_manager_test” 来跑该题目的 Googletest 测试

- 详细流程参考“ docs/Rucbase学生实验操作说明示例.md”