实验四：查询处理算法的实现

高宏 邹兆年

1. 实验目的

掌握关系连接操作的实现算法，理解算法的 I/O 复杂性，使用高级语言语言实现重要的关系连接操作算法。

2. 实验环境

使用本实验提供函数库或者使用其他语言重构函数库实现均可。

3. 实验内容

3.1 实验任务

关系 R 具有两个属性 A 和 B，其中 A 和 B 的属性值均为 int 型（4 个字节），A 的值域为[1, 40]，B 的值域为[1, 1000]。

关系 S 具有两个属性 C 和 D，其中 C 和 D 的属性值均为 int 型（4 个字节）。C 的值域为[20, 60]，D 的值域为[1, 1000]。

1）实现关系选择算法：基于 ExtMem 程序库，使用 C 实现关系选择算法，选出 R.A=40 或 S.C=60 的元组，并将结果存放在磁盘上。要求实现至少三种选择算法，包括线性搜索算法，二元搜索算法，和任意一种索引算法（使用 B+/B-树可加分）。

2）实现关系投影算法：基于 ExtMem 程序库，使用 C 语言实现关系投影算法，对关系 R 上的 A 属性进行投影，并将结果存放在磁盘上。

3）实现连接操作算法：基于 ExtMem 程序库，使用 C 语言实现连接操作算法，对关系 R 和 S 计算 R.A 连接 S.C，并将结果存放在磁盘上。要求实现三种连接操作算法：Nest-Loop- Join 算法，Sort-Merge-Join 算法，Hash-Join 算法。

注意：

（**1**）需要在检查前准备好相应的数据，详见 **3.3** 节。

（**2**）每人需要完成上述全部算法，根据完成情况给分。

（**3**）本次实验不需要提交实验报告，但需要提交实验代码。

（**3**）如果自创函数库，使用其他高级程序语言实现选择、投影、连接算法亦可。

3.2 ExtMem 程序库介绍

ExtMem 程序库是一个专门为本课程编写的模拟外存磁盘块存储和存取的程序库，由 C 语言开发。ExtMem 程序库的功能包括内存缓冲区管理、磁盘块读/写它提供了 1 个数据结构和 7 个 API 函数。

ExtMem 程序库定义了 Buffer 数据类型，包含如下 6 个域：

- numIO：外存 I/O 次数；
- bufSize：缓冲区大小（单位：字节）；
- blkSize：块的大小（单位：字节）；
- numAllBlk：缓冲区内可存放的最多块数；
- numFreeBlk：缓冲区内可用的块数；
- data：缓冲区内存区域。

缓冲区内每个块的大小为 blkSize 个字节，其最后 4 个字节用来存放其后继磁盘块的地址（在 ExtMem 库中，我们 4 个字节来记录磁盘块地址，地址在程序中为 unsigned int 类型。若无后继磁盘块，则置为 0），其余(blkSize – 4)个字节用于存放块内的记录。

ExtMem 库提供了如下 API 函数：

- Buffer *initBuffer(size_t bufSize, size_t blkSize, Buffer *buf);
  初始化缓冲区，其输入参数 bufSize 为缓冲区大小（单位：字节），blkSize 为块的大小（单位：字节），buf 为指向待初始化的缓冲区的指针。若缓冲区初始化成功，则该函数返回指向该缓冲区的地址；否则，返回 NULL。
- void freeBuffer(Buffer *buf);
  释放缓冲区 buf 占用的内存空间。
- unsigned char *getNewBlockInBuffer(Buffer *buf);
  在缓冲区 buf 中申请一个新的块。若申请成功，则返回该块的起始地址；否则，返回 NULL。
- void freeBlockInBuffer(unsigned char *blk, Buffer *buf);
  解除块 blk 对缓冲区内存的占用，即将 blk 占据的内存区域标记为可用。
- int dropBlockOnDisk(unsigned int addr);
  从磁盘上删除地址为 addr 的磁盘块内的数据。若删除成功，则返回 0；否则，返回-1。
- unsigned char *readBlockFromDisk(unsigned int addr, Buffer *buf);
  将磁盘上地址为 addr 的磁盘块读入缓冲区 buf。若读取成功，则返回缓冲区内该块的地址；否则，返回 NULL。同时，缓冲区 buf 的 I/O 次数加 1。
- int writeBlockToDisk(unsigned char *blkPtr, unsigned int addr, Buffer *buf);
  将缓冲区 buf 内的块 blk 写入磁盘上地址为 addr 的磁盘块。若写入成功，则返回 0；否则，返回-1。同时，缓冲区 buf 的 I/O 次数加 1。

文件 test.c 中给出了 ExtMem 库使用方法的一个具体示例。

声明：ExtMem 库是为本课程专门开发的模拟外存磁盘块存储和存取的程序库，不保证其能够真正实现对磁盘块的存取操作，同时也不保证其排除一切软件错误。本课程及 ExtMem 开发者不会对使用该程序库所导致的一切错误负责。

3.3 数据准备

使用 ExtMem 程序库建立两个关系 R 和 S 的物理存储。关系的物理存储形式为磁盘块序列 $B_1, B_2, \ldots, B_n$，其中 B_i 的最后 4 个字节存放 B_{i+1} 的地址。

即 R 和 S 的每个元组的大小均为 8 个字节。

块的大小设置为 64 个字节，缓冲区大小设置为 512+8=520 个字节。这样，每块可存放 7 个元组和 1 个后继磁盘块地址，缓冲区内可最多存放 8 个块。

编写程序，随机生成关系 R 和 S，使得 R 中包含 16 * 7 = 112 个元组，S 中包含 32 * 7 = 224 个元组。

3.4 补充说明

实验要求限制使用的内存大小。即算法的操作应在 8 个块的内存缓冲区实现，不能再另开较大内存空间存放数据。必要的变量、非存放数据的较小数组等可以存放在内存的其他地址（非缓冲区）。

4. 参考资料

Abraham Silberschatz, Henry F.Korth. 《数据库系统概念（第六版）》