|  |
| --- |
| 哈尔滨工业大学(深圳) |
| **《数据库》实验报告** |
|  |
| **实验四**  **查询处理算法的模拟实现**  学 院: 计算机科学与技术   |  |  | | --- | --- | | 姓 名: | 胡聪 | | 学 号: | 180110505 | | 专 业: | 计算机科学与技术 | | 日 期: | 2021-05-05 | |

# 实验目的

理解索引的作用，掌握关系选择、连接、集合的交、并、差等操作的实现算法，理解算法的I/O复杂性。

# 实验环境

Windows 10操作系统、Visual Studio Code

# 实验内容

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[20, 60]，B的值域为[1000, 2000]。

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[40, 80]，D的值域为[1000, 3000]。

① 实现基于线性搜索的关系选择算法：基于ExtMem程序库，使用C语言实现线性搜索算法，选出S.C=50的元组，记录IO读写次数，并将选择结果存放在磁盘上。（模拟实现 select S.C, S.D from S where S.C = 50）

② 实现两阶段多路归并排序算法（TPMMS）：利用内存缓冲区将关系R和S分别排序，并将排序后的结果存放在磁盘上。

③ 实现基于索引的关系选择算法：利用（2）中的排序结果为关系S建立索引文件，利用索引文件选出S.C=50的元组，并将选择结果存放在磁盘上。记录IO读写次数，与（1）中的结果对比。（模拟实现 select S.C, S.D from S where S.C = 50 ）

④ 实现基于排序的连接操作算法（Sort-Merge-Join）：对关系S和R计算S.C连接R.A ，并统计连接次数，将连接结果存放在磁盘上。 （模拟实现 select S.C, S.D, R.A, R.B from S inner join R on S.C = R.A）

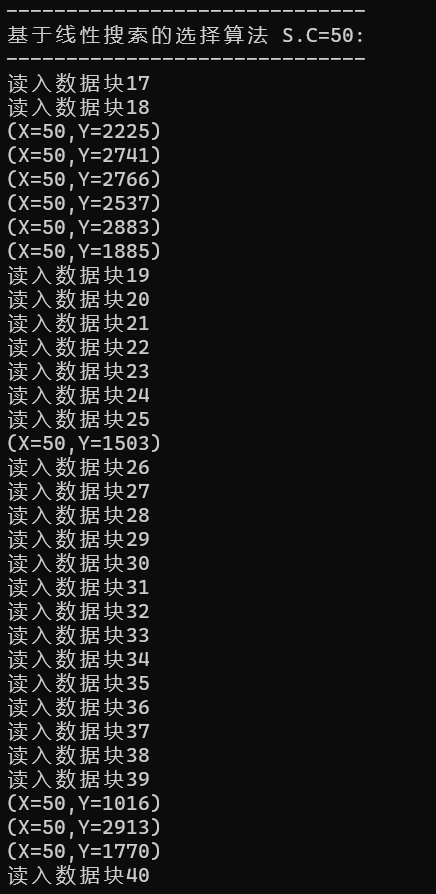
⑤ 实现基于排序或散列的两趟扫描算法，实现并（SR） 、交（SR） 、差（S - R）其中一种集合操作算法，将结果存放在磁盘上，并统计并、交、差操作后的元组个数。

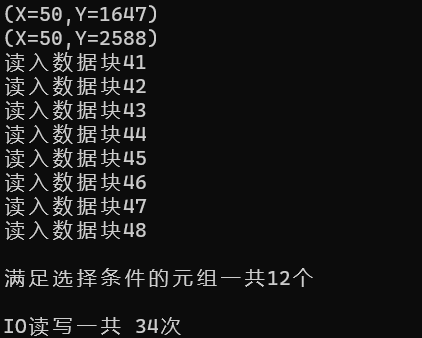
# 实验过程

1. **实现基于线性搜索的关系选择算法**

问题分析：从头到尾依次进行搜索即可找到符合题目要求的数据。

实验结果：





1. **实现两阶段多路归并排序算法（TPMMS）**

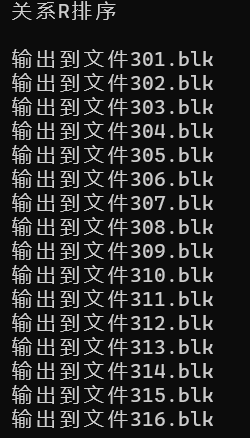
问题分析：两阶段多路归并排序算法常用来解决数据量较大的问题，特别是数据大小超过了数据库的可用主存大小，无法一次将数据载入主存的情况。该算法包括两个阶段。

阶段1：将数据片段进行排序，使得每条记录都是一个有序表的一部分，该表即为排序子表，这样的表产生了多个；

阶段2：归并排序所有的排序子表，最后形成一个排序表。

该算法只让磁盘中的数据写入主存两次，完成了数据的排序，节省了IO时间。

实验结果：

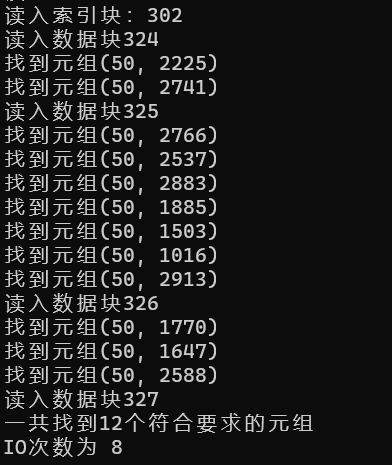
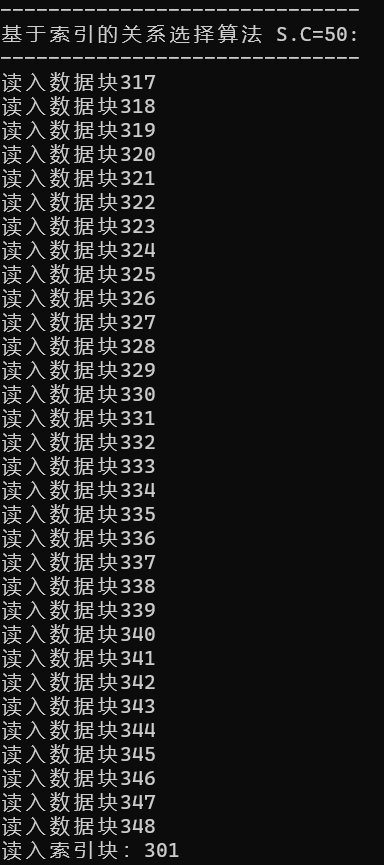




1. **实现基于索引的关系选择算法**

问题分析：在做选择操作时，如果选择条件中有属性没有被索引，那么需要将关系表中的元组加载到内存中以判断是否满足条件，如果满足则输出，如果表中查询涉及到的属性有索引，则在执行选择操作的时候，首先将索引文件加载到内存中，然后通过索引找到元组，判断是否满足条件并且输出。

实验结果：

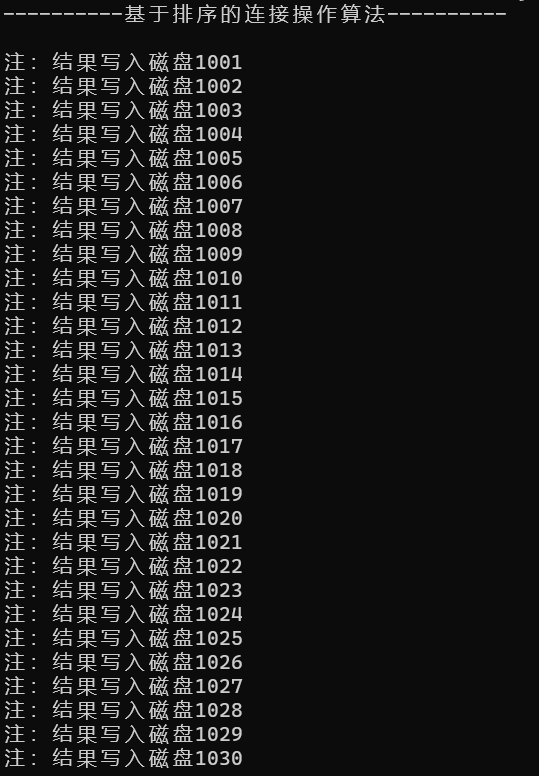


1. **实现基于排序的连接操作算法（Sort-Merge-Join）**

问题分析：第一趟：划分R和S的子表并进行子表排序，排序均基于连接属性排序。

第二趟：归并时注意是R的输入还是S的输入。R和S的两路输入之间进行连接检查并连接后输出。

实验结果：



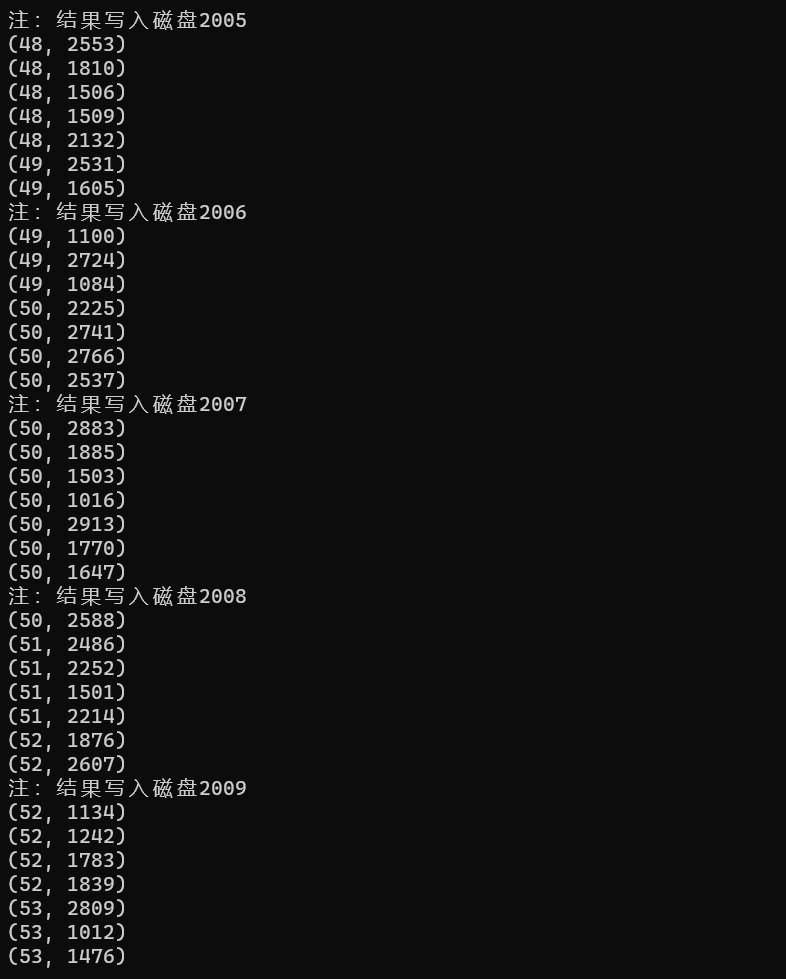
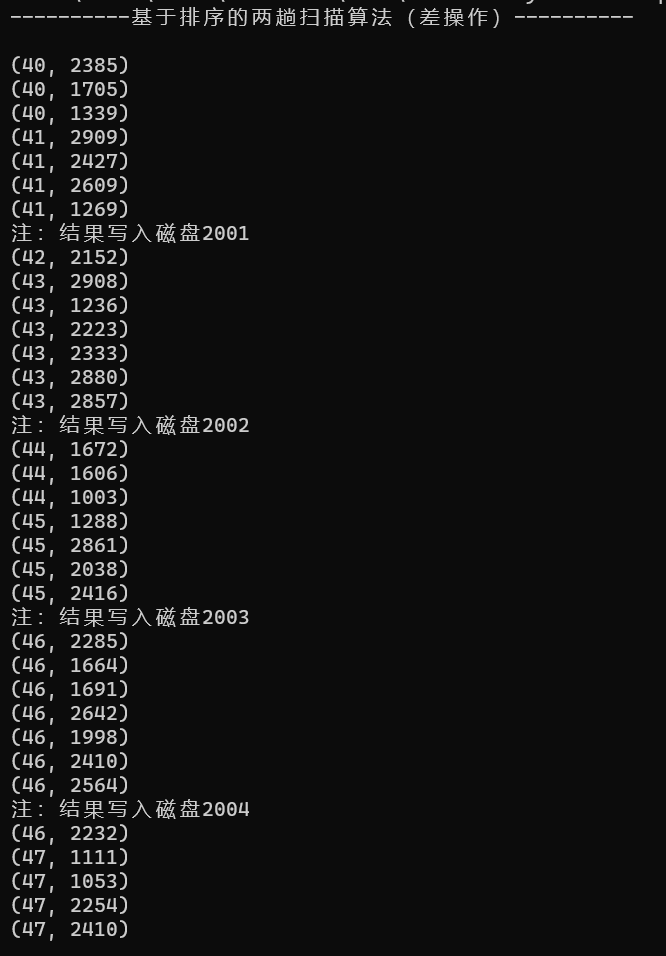


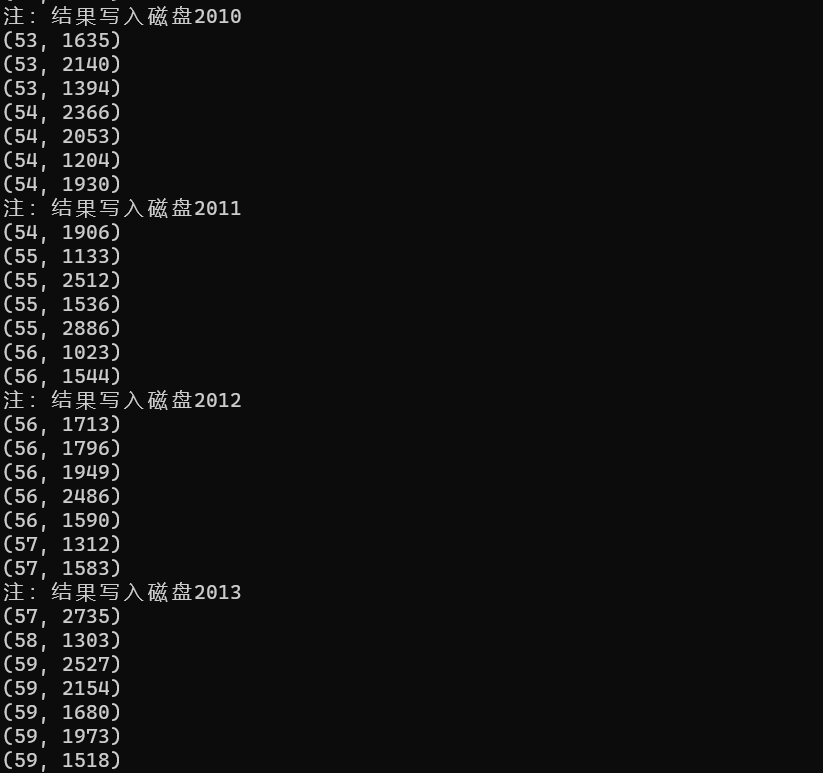


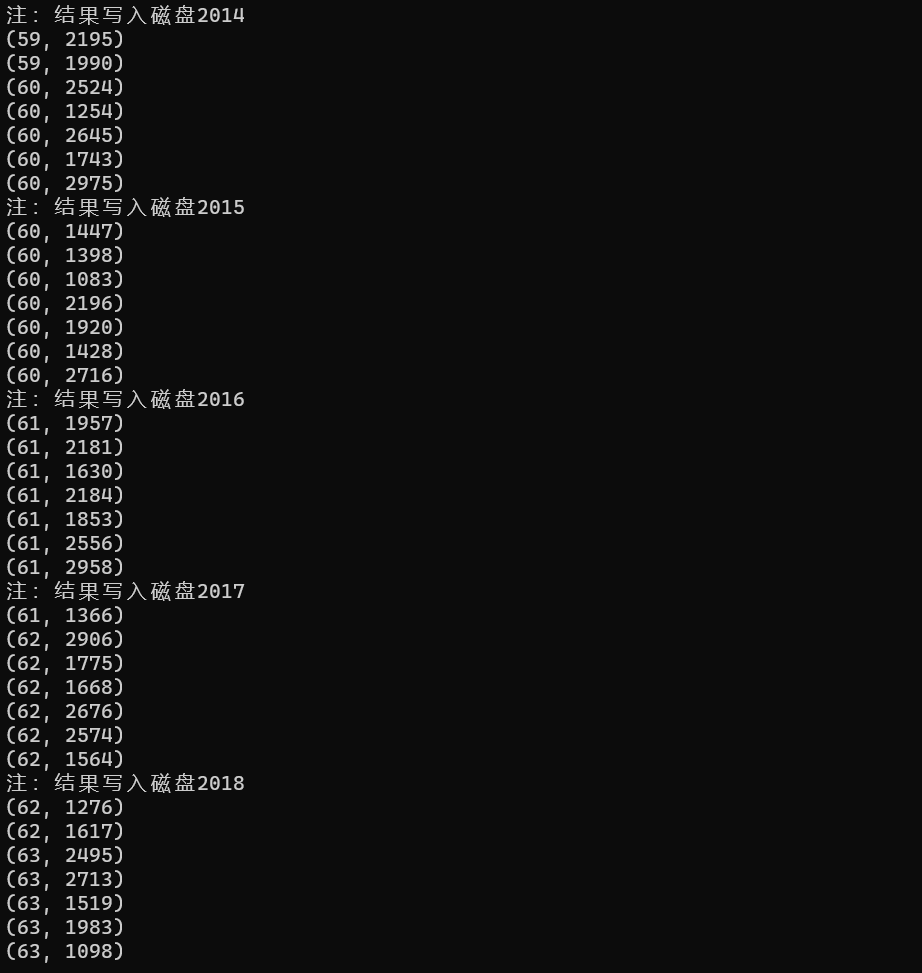
1. **实现基于散列的两趟扫描算法，实现交、并、差其中一种集合操作算法**

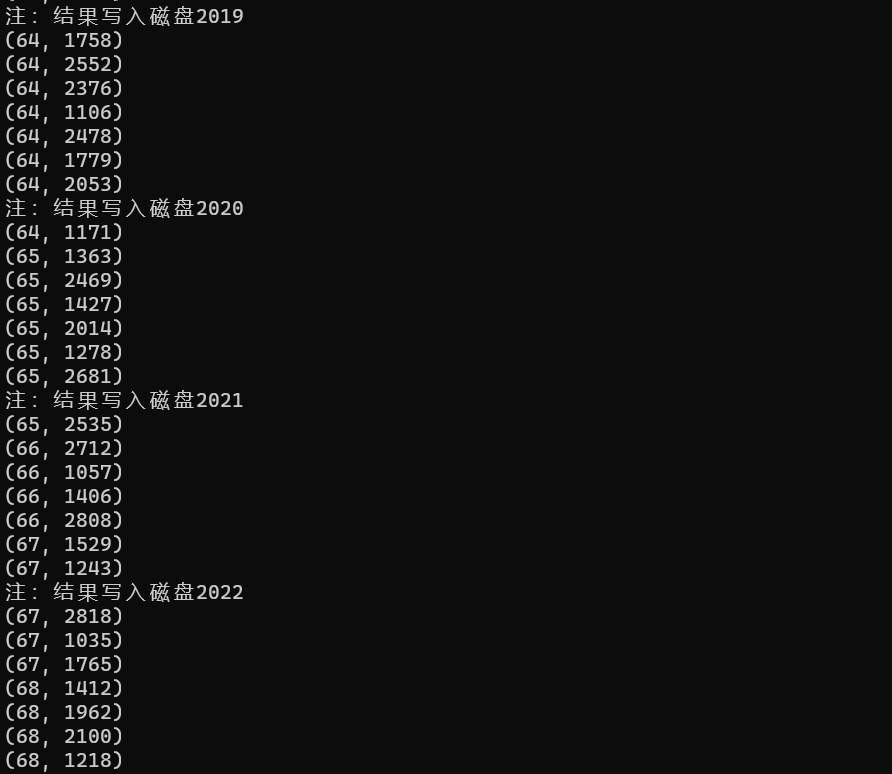
问题分析：基于散列的两趟扫描算法，第一趟划分子集，并使子集具有某种特性，如有序或相同散列值等，将原始关系通过hp散列成；第二趟处理全局性内容的操作，形成结果关系。如多子集间的归并操作，相同散列值子集的操作等。

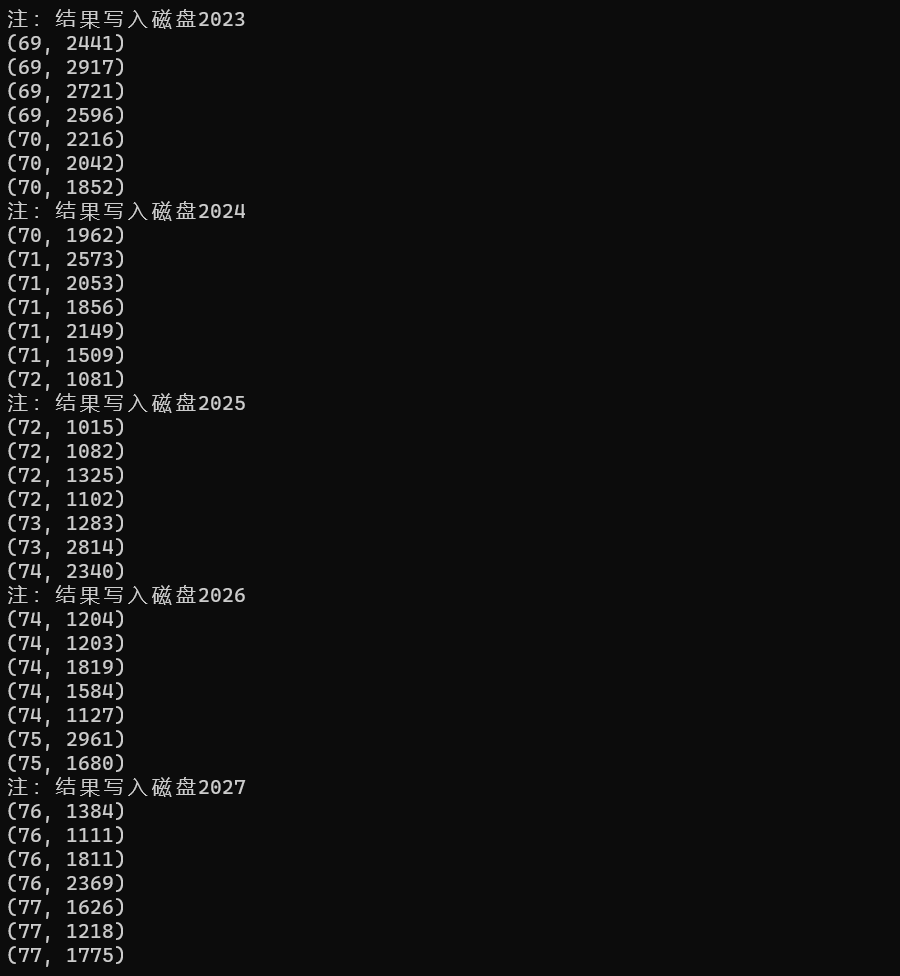
实验结果：

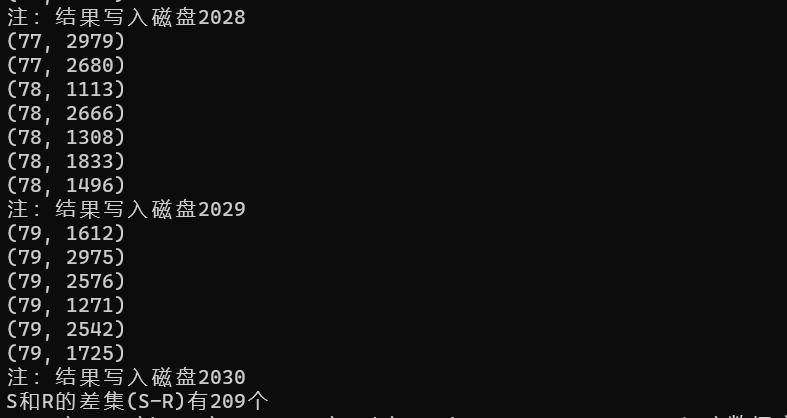












# 附加题

# 总结

本次实验，与之前的实验不同的是，涉及到了较为底层的数据库算法，且题目中要求有限的缓冲区限制，给实验过程增加了一些挑战。实验的一开始，首先需要了解ExtMem库的使用方法，仔细阅读指导书，了解库提供的一些接口，才能开始上手本实验。

实验过程中，也遇到了很多问题，例如CMake的使用再次出现了问题，调试过程中发现了段错误等问题却又很难定位到问题所在，好在后来通过查找最终解决了这些问题。虽然本次实验中很多东西以后在工作中用不上，但重要的是解决底层问题的思想，这些是很宝贵的经验。