程序设计 II，2014 年秋
大作业：简易数据存储系统

发布日期：2014 年 12 月 6 日

期末检查：2015 年 1 月 3 日（第 16 周周六）

最终检查：待定（全部考试结束后的晚上和第二天）

1. 概述

程序设计 2（数据结构）大作业，强调数据结构的选择和对性能的分析。涉及查找、树、散列等操作。目的在于让同学们利用所学的知识，设计并构建一个简易的数据存储系统，进一步理解和掌握基本的数据结构和算法。

2. 功能要求

你需要完成对数据的增删改查。详细描述，就是每一组数据都包含一个名字和其内容，你实现的存储系统需要将每组数据的内容保存在磁盘中，并且能够根据名字查找对应的内容，进行查看，删除，修改操作。数据存储、索引的形式有较大的灵活性，你可以选择合适的数据结构进行实现。所以你需要在最后提交的文档中描述你使用了怎样的数据结构，效果如何；以及磁盘中数据和索引的组织格式。

2.1. 数据文件

数据文件存储所保存数据的内容，保证下一次运行时能够从磁盘文件中读取。数据文件的格式可以自行设计，包括文件头，数据描述信息等。你的系统应提供对数据进行增删改查的接口，接口可参考附录资料，也可以自行设计。

2.2. 索引文件

除了数据以外，还需要考虑对数据进行索引，以便根据名字能快速找到对应的数据内容。通过索引可以快速定位数据在文件中的位置并获取相关数据。

你可以在磁盘上采用树、散列等数据结构保存索引信息，具体实现自行设计，自由度较大。

你可以将全部索引保存在一个文件中，或者保存在多个文件中，甚至采用二级索引以便减小单个索引文件的大小。数据文件你也可以将其全部保存在一个文件中，或者保存在多个文件中。但不允许以每组数据（名字和其内容）为单位单独保存在一个文件中，索引和数据文件需要分开。

2.3. 一致性刷新方法设计

因为数据量较大，系统运行时不能够将所有的数据存放在内存中，因此一份数据可能同

时出现在磁盘和内存中。你需要考虑何时对文件进行读写以便保持文件的一致性，且保证运行效率。

3. 系统测试

关于测试的程序，你的测试应该独立于实现的数据存储系统，并且应该直接针对该系统的 API 进行编写，不需要通过读取测试脚本等外部 IO 方式进行测试。

具体的测试内容可以自行设计，至少包括增删改查操作。可参考附录[1]中的测试。

3.1. 正确性测试

正确性测试既包括结果正确没有引发程序的异常退出或内存泄露，也包括符合预期的性能要求。你需要自行书写测试内容，测试数据可以用使用一些随机数据生成工具。

测试应涵盖程序的各个操作，并且应当保证数据量不少于 100 万条。

3.2. 性能测试

性能测试与正确性测试要求类似，但注重考查程序的性能。同样需要进行不少于 100 万数据量的测试，可以记录不同数据量的操作所耗费的时间，不同数据规模时进行同样的操作所耗费的时间。并使用相关绘图软件绘制时间-数据量的散点图或者拟合图，并对结果图进行分析解释。

注意，由于每个人的机器配置不一样，和其他人比较性能是没有意义的，我们也不会根据你的性能进行评分，我们关注的是你通过测试可以证明程序符合设计的预期，并且通过测试可以发现在使用和不使用索引的性能差异，以及不同数据结构的性能差异。

提交作业时，需要包含测试文件和测试报告，测试报告应描述你的测试过程，测试结果图和对测试结果的分析。

4. 其他限制要求

4.1. 自行实现关键数据结构

大作业程序中使用的关键数据结构，例如散列、B 树需要自行实现，文件的一致性刷新也需要自行实现，不可使用任何已有的库。

4.2. 函数库的使用

不涉及关键数据结构的部分，可以随意使用你喜欢的库进行开发。注意不要实现过多无用的功能而忽视了作业要求的部分，这对你的成绩不会有任何有益的影响。

4.3. 开发和编译环境

程序需要用 C/C++实现，答辩时需要大家演示自己的系统，程序为单机程序，不需要实现多线程、网络通讯等功能。

5. 检查

一共有两次检查的时间。

如果你在期末检查前完成了全部要求，可以直接给我们进行检查，通过后不需要第二次检查。

如果你在期末检查没有完成全部要求，务必在最终提交时给我们进行检查，最终提交是大作业的最后期限。

具体的检查安排，包括检查地点，会在检查前公布在课程网站上，请留意。

6. 评分

大作业占这门课程总成绩的 20%。满分 20 分中，存储系统的实现占 10 分，测试和文档占 10 分。如果你的索引实现了 B 树等较为复杂的数据结构，或者在某一部分使用了较为高级的算法和数据结构，存储系统的实现部分的 10 分都可拿到满分。

7. 提示和建议

一般来说，测试和修复错误的时间会占到开发总时间的一半以上，尤其涉及到一些较为复杂的算法时测试所占的比重往往会更多。所以请尽早着手完成作业，注意合理安排好自己的时间。

考虑使用二进制文件读写而非文本文件读写，这样既可以提高程序效率，又可以降低开发难度。

虽然我们不允许你直接使用已有的库来实现系统中关键的数据结构，但建议你查阅相关资料，参考已有的数据结构而非从头开始自己的设计。例如你可以去了解数据库系统都是用了怎样的方式管理数据，虽然数据库系统可能有很多较为复杂而我们不需要关心的高级话题（如并发控制），但对于数据的存储和索引，这些设计对我们很具有参考价值。

建议你在开始书写程序之前就对你要使用的数据结构有所设想，以避免在开发过程中，因为要使用某种数据结构而出现对已经完成的代码大量的返工。

8. 参考及附录

1. 数据库函数库
2. SQLite: http://www.sqlite.org/
3. Oracle Berkeley DB: http://www.oracle.com/technetwork/database/berkeleydb/index.html
4. Google C++ Style Guide: http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml
5. JPL Coding Standard
6. 华为编程规范