

从上周六开始，一直玩到周二，浪费了许多时间。

这周学习的内容：

《Linux系统编程》（李慧芹） p1-p9，主要包括标准IO部分，学习到的技巧：

- man 查看对象的系统IO函数，获得帮助
- int argc, char ** argv从命令行获得参数，注意包括第一个可执行文件名称是第一个参数，从0开始计数
- IO分为系统IO和标准IO，为什么要有标准IO呢？对应于不同的操作系统
- 后续跟着这个课程走

LeetCode刷题，从周四晚上开始刷二分查找的题目

33,81,29,162,50,74，值得关注的是

- 50题，快速幂运算
- 29两数相除

中兴数据库比赛代码优化，主要采用unordered_multimap和unordered_map进行数据的存储，还是不太理解，为什么他说占用的内存很小，只需要把当前需要计算的数据载入内存。

- 数据读入，装入到vector<dataOriginal*>中
- 将t2表格的数据存储到unordered_multimap<int, dataOriginal*>中，key是id3,value就是对应的数据，循环嵌套，在t2中查找t1Id3，生成一个pair<dataOriginal*, dataOriginal*>，最终join的结果就是vector<pair<dataOriginal*, dataOriginal*>>。
- 然后进行group。group的依据是t1Id2, t2Id2，还是采用map来做这件事情，默认按key排序了嘛。此时的map类型是unordered_map<int, unordered_map<int, groupData*>>，第一个key就是t1Id2，第二个key就是t2Id2，此时统计分组的maxT1Id1和minT2Id1即可。最终在将分好组的结果转换成一个

vector (已经按t1Id2,t2Id2排好序) , 最终得到group结果是vector<groupData*>.

- 排序直接采用std::stable_sort, 自定义一个比较函数。
- 速度110ms, 占用内存395k
- 该优化过程同时上传是github, 方便日后查看

linux工具使用

- 内存分析工具valgrind
 - 生成对应的文件, valgrind --tool=massif ./result
 - 可视化对应的结果 massif-visualizer + 上述程序生成结果
- vim使用
 - 光标跳转至行尾\$
 - 删除光标至结尾部分D
 - 行选择shift+v, 列选择ctrl+v
 - 剪切x, 粘贴p
 - 将当前vim转至后台运行Ctrl+z,输入fg转回至vim
 - 撤销u,恢复撤销Ctrl+r
- C文件编译, 直接采用make+对应的文件名 (去掉.c后缀) , 就可以执行编译

下周工作计划

- linux系统编程---p46进程之前
- 网络编程提高篇+期中复习
- 数据结构课程结束 (最小生成树+最短路径)
- 深度探索C++对象模型复习
- STL源码剖析