

Lab1 test report

1.1 程序输入

程序将在控制台接收用户输入，该输入应为某一目录下的一个数独谜题文件，该文件包含多个数独谜题，每个数独谜题按固定格式存储在该文件中。

1.2 程序输出

  实验中把数独的解按与输入相对应的顺序写入到std。

1.3 Sudoku算法

  本次实验我们小组使用了 DANCEING LINK 算法

2. 性能测试

实验环境：MacOS 4core 4 thread

型号名称： MacBook Air

型号标识符： MacBookAir7,2

处理器名称： Intel Core i5

处理器速度： 1.8 GHz

处理器数目： 1

核总数： 2

L2 缓存（每个核）： 256 KB

L3 缓存： 3 MB

内存： 8 GB

Boot ROM 版本： MBA71.0176.B00

SMC 版本（系统）： 2.27f2

序列号（系统）： FVFTXQURJ1WK

硬件 UUID： 8FD84269-876D-55DD-A276-74D60B79AEF9

以下是测试内容：

实验代码4线程解决1个数独用时： 0.000530s

原始代码单线程解决1个数独用时： 0.000483s

实验代码4线程解决10个数独用时： 0.001195s

原始代码单线程解决10个数独用时： 0.001230s

实验代码4线程解决1000个数独用时： 0.018120s

原始代码单线程解决1000个数独用时： 0.044272s

可以看到当使用4线程时解决数量比较大的数独问题时比原来的单线程快了一倍多。

同时解决1000个数独问题时，使用不同线程数的用时比较：

1线程： 0.044272s

2线程: 0.029070s

3线程: 0.024895s

4线程: 0.018120s

5线程: 0.016162s

6线程: 0.023295s

7线程: 0.024913s

从上面也可以看到线程也不是开得越多越好, 因为线程开太多对cpu反而会带来负提升。