



北京交通大学
BEIJING JIAOTONG UNIVERSITY

《计算机体系结构》 实验报告

实验名称:	实验 1_ PC 性能评测实验报告
学 号:	
姓 名:	
学 院:	计算机与信息技术学院
日 期:	2022 年 9 月 23 日

目录

1. 实验目的	错误!未定义书签。
2. 实验内容	错误!未定义书签。
3. 实验思路	错误!未定义书签。
4. 程序实现	错误!未定义书签。
4.1. 相关环境介绍	错误!未定义书签。
4.2. 内置命令选择和结构体设计	错误!未定义书签。
4.3. 重要函数说明和流程图	错误!未定义书签。
5. 程序验证	错误!未定义书签。
6. 实验汇总	3
6.1. 实验要求完成情况	错误!未定义书签。
6.2. 技术难点及解决方案	错误!未定义书签。
6.3. 实验感想和经验总结	错误!未定义书签。
6.4. 参考链接	错误!未定义书签。

题目 1：PC 性能测试软件

请在网上搜索并下载一个功能较全面的 PC 机性能评测软件（比如：可在百度上输入“PC 性能 benchmark”，进行搜索并下载，安装），利用该软件对你自己的电脑进行性能测试，并加以详细分析。在条件允许的情况下，将你的电脑与另外一台电脑的性能均进行性能测试和分析比较。

本人电脑基本情况：

电脑型号	联想 Lenovo 小新 Air-14IIL 2020 笔记本电脑
操作系统	Windows 10 家庭版 64 位 (Version 21H2 / DirectX 12)
处理器	英特尔 Core i5-1035G1 @ 1.00GHz 四核 八线程
主板	联想 LNVNB161216 (I/O - 3482 for Intel 495 Series 芯片组)
显卡	NVIDIA GeForce MX350 (2 GB / 联想)
内存	16 GB (三星 DDR4 3200MHz 8GB x 2)
主硬盘	三星 MZVLB512HBJQ-000L2 (512 GB / 固态硬盘)
显示器	群创光电 CMN1408 (14 英寸)
声卡	瑞昱 High Definition Audio
网卡	英特尔 Wireless-AC 9560

测试软件：鲁大师、CPU-Z、GPU-Z、

以下是具体情况：

CPU

处理器	英特尔 Core i5-1035G1 @ 1.00GHz 四核
速度	1.00 GHz
处理器数量	核心数：4 / 线程数：8
核心代号	Ice Lake
生产工艺	10 nm
插槽/插座	FCBGA1526
一级数据缓存	4 x 48 KB, 12-Way, 64 byte lines
一级代码缓存	4 x 32 KB, 8-Way, 64 byte lines
二级缓存	4 x 512 KB, 8-Way, 64 byte lines
三级缓存	6 MB, 12-Way, 64 byte lines
指令集	MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, HTT, EM64T, EIST, Turbo Boost

比对测试：

基准 CPU 为 intel i9 12900K，本地 CPU 性能相当于其 7%的性能。(来自 CPU-Z)

GPU

主显卡	NVIDIA GeForce MX350
显存	2 GB
频率	核心: 1354MHz / 显存: 1752MHz
显卡制造商	联想
芯片制造商	Nvidia
BIOS 版本	86.07.90.00.6e
驱动版本	31.0.15.1659
显卡	英特尔 UHD Graphics
显存	128 MB
显卡制造商	联想
芯片制造商	Intel
驱动版本	30.0.100.9955

对比测试：

基准显卡 Nvidia 4090

时钟频率为其 60% 1354/2235 (MZ) .

浮点运算能力为其 2.3% 1.88/82.58 (TFLOPS)

总体性能为其 17.5% (来自 VERSUS)

内存

ChannelA-DRAM	三星 DDR4 3200MHz 8GB
型号	M471A1G44AB0-CWE
厂商	SAMSUNG
模块位宽	64 bits
模块电压	SSTL 1.2V

硬盘

产品	三星 MZVLB512HBJQ-000L2 (固态硬盘)
大小	512 GB
固件	3L1QEXF7

缓存	512 MB
接口	PCI-E 3x4
数据传输率	4000.00 MB/秒
特性	S.M.A.R.T
硬盘已使用	共 643 次, 累计 3347 小时

题目·2: toy benchmark 的编写并测试

用 C 语言编写一个程序, 该程序包括两个子程序, 两个子程序具有相同的功能, 但采用的算法不同, 因而执行速度有较大差异 (分别标记为快速和慢速)。在主程序中, 要将两个子程序调用很多次 (以突显其运行时间差距), 调用的频率 (频率快速, 频率慢速) 可调整。比如, 对于频率 (0.3, 0.7), 可将两个子程序分别运行 30000 次和 70000 次 (或者等比率增减)。

请以 (0.5, 0.5), (0.1, 0.9), (0.9, 0.1) 的频率分别运行你编写的程序, 并计算出三种情况下的加权平均运行时间。

这里按题目提示采用了使用两种不同的方法求最大公约数的函数, 赋予不同比例求最终的加权时间。

满方法为暴力遍历, 快方法为辗转相除法。

代码如下:

```
1. #include <iostream>
2. #include <ctime>
3. #include <random>
4. using namespace std;
5.
6. int slow_gcd(int a, int b){
7.     for(int i=b; i>=2; --i){
8.         if(a%i==0 && b%i==0){
9.             break;
10.        }
11.        else{
12.            continue;
13.        }
14.    }
15. }
16.
17. int fast_gcd(int a, int b){
18.     if(a%b==0){
19.         return 0;
20.     }
21.     else{
```

```
22.         fast_gcd(b, a%b);
23.     }
24. }
25.
26. double run(double k){
27.     clock_t start, end;
28.     int a, b;
29.     int count = k*10000;
30.
31.     start = clock();
32.     for(int i=0; i<count; ++i){
33.         a = (rand()<<5) + rand();
34.         b = (rand()<<5) + rand();
35.         if(a<b){
36.             int tem = a;
37.             a = b;
38.             b = tem;
39.         }
40.         slow_gcd(a, b);
41.     }
42.     end = clock();
43.
44.     double slow_gcd_time = (double)(end-start)/CLOCKS_PER_SEC;
45.
46.     count = (1-k)*10000;
47.     start = clock();
48.     for(int i=0; i<count; ++i ){
49.         a = (rand()<<5) + rand();
50.         b = (rand()<<5) + rand();
51.         if(a<b){
52.             int tem = a;
53.             a = b;
54.             b = tem;
55.         }
56.         fast_gcd(a, b);
57.     }
58.     end = clock();
59.     double fast_gcd_time = (double )(end - start)/CLOCKS_PER_SEC;
60.
61.     double total_time = fast_gcd_time + slow_gcd_time;
62.     return total_time;
63. }
64. int main() {
```

```

65.     cout << "0.5  " << run(0.5) << "\n";
66.     cout << "0.1  " << run(0.1) << "\n";
67.     cout << "0.9  " << run(0.9) << "\n";
68. }

```

运行结果：

```

0.5  3.771
0.1  0.761
0.9  6.792

```

第一个数为慢速算法所占比例，后边为时间秒。

题目 3：minion 程序的运行

Minion 是约束满足问题的求解器。运行的方法是：

1. 进入 DOS 环境(点击开始，然后在最下面一栏输入 cmd 后回车)。
2. 进入你的程序目录，其中有可执行程序 minion.exe，以及两个测试实例。
3. 运行：minion 某实例。(需要运行一段时间，比如 3 分钟，你要耐心等待)

请分别以 2 个测试实例为输入，分别在你自己的计算机和另一台计算机上运行，得出四个运行时间。然后，归一化到你自己的计算机，计算出几何平均值，并由此比较两台机器的性能。如果你手头没有另一台电脑，可用另一位同学电脑的运行数据。

测试结果：

本次结果：

```

D:\BJTU\subject\Junior First\Computer Architecture\lab1\实验1\minion_test>minion.exe k6p2_table.minion
# Minion Version 0.12
# Git version: 65512633daee570de1fdf16a0025d919f6f3753e
# Git last changed date: Mon Feb 8 17:33:56 2010 +0000
# Run at: UTC Fri Sep 23 01:51:21 2022

# http://minion.sourceforge.net
# Minion is still very new and in active development.
# If you have problems with Minion or find any bugs, please tell us!
# Mailing list at: https://mail.cs.st-andrews.ac.uk/mailman/listinfo/mug
# Input filename: k6p2_table.minion
# Command line: minion.exe k6p2_table.minion
Parsing Time: 0.000000
Setup Time: 0.125000
First Node Time: 0.000000
Initial Propagate: 0.000000
First node time: 0.000000
Solve Time: 67.531250
Total Time: 67.656250
Total System Time: 0.046875
Total Wall Time: 67.789000
Maximum Memory (kB): 0
Total Nodes: 638007
Problem solvable?: no
Solutions Found: 0

```

```
D:\BJTU\subject\Junior First\Computer Architecture\lab1\实验1\minion_test>minion.exe solitaire_benchmark_8.minion
# Minion Version 0.12
# Git version: 65512633daee570delfdf16a0025d919f6f3753e
# Git last changed date: Mon Feb 8 17:33:56 2010 +0000
# Run at: UTC Fri Sep 23 01:53:35 2022

# http://minion.sourceforge.net
# Minion is still very new and in active development.
# If you have problems with Minion or find any bugs, please tell us!
# Mailing list at: https://mail.cs.st-andrews.ac.uk/mailman/listinfo/mug
# Input filename: solitaire_benchmark_8.minion
# Command line: minion.exe solitaire_benchmark_8.minion
Parsing Time: 0.078125
Setup Time: 0.015625
First Node Time: 0.031250
Initial Propagate: 0.031250
First node time: 0.000000
Solution Number: 1
Time: 87.406250
Nodes: 1116063

Solve Time: 87.406250
Total Time: 87.531250
Total System Time: 0.062500
Total Wall Time: 87.771000
Maximum Memory (kB): 0
Total Nodes: 1116063
Problem solvable?: yes
Solutions Found: 1
```

同学机器的结果:

```
C:\Users\Lenovo\Desktop\体系结构\lab1\实验1\minion_test>minion.exe k6p2_table.minion
# Minion Version 0.12
# Git version: 65512633daee570delfdf16a0025d919f6f3753e
# Git last changed date: Mon Feb 8 17:33:56 2010 +0000
# Run at: UTC Wed Sep 21 08:40:17 2022

# http://minion.sourceforge.net
# Minion is still very new and in active development.
# If you have problems with Minion or find any bugs, please tell us!
# Mailing list at: https://mail.cs.st-andrews.ac.uk/mailman/listinfo/mug
# Input filename: k6p2_table.minion
# Command line: minion.exe k6p2_table.minion
Parsing Time: 0.000000
Setup Time: 0.078125
First Node Time: 0.000000
Initial Propagate: 0.000000
First node time: 0.000000
Solve Time: 43.171875
Total Time: 43.250000
Total System Time: 0.031250
Total Wall Time: 43.885000
Maximum Memory (kB): 0
Total Nodes: 638007
Problem solvable?: no
Solutions Found: 0

C:\Users\Lenovo\Desktop\体系结构\lab1\实验1\minion_test>
```

```
# Minion Version 0.12
# Git version: 65512633daee570delfdf16a0025d919f6f3753e
# Git last changed date: Mon Feb 8 17:33:56 2010 +0000
# Run at: UTC Wed Sep 21 08:46:25 2022

# http://minion.sourceforge.net
# Minion is still very new and in active development.
# If you have problems with Minion or find any bugs, please tell us!
# Mailing list at: https://mail.cs.st-andrews.ac.uk/mailman/listinfo/mug
# Input filename: solitaire_benchmark_8.minion
# Command line: minion.exe solitaire_benchmark_8.minion
Parsing Time: 0.046875
Setup Time: 0.015625
First Node Time: 0.015625
Initial Propagate: 0.015625
First node time: 0.000000
Solution Number: 1
Time: 60.671875
Nodes: 1116063

Solve Time: 60.671875
Total Time: 60.750000
Total System Time: 0.062500
Total Wall Time: 60.988000
Maximum Memory (kB): 0
Total Nodes: 1116063
Problem solvable?: yes
Solutions Found: 1
```


时间表格

	k6p2_table	solitaire_benchmark_8
本地	67.789	87.771
同学机器	43.885	60.988
归一化	1.544696366	1.439151964

本地 CPU 为 Core i5-1035G1,
同学 CPU 为
我的