计算机体系结构

PC性能测试实验报告

|  |  |
| --- | --- |
| 学 院： | 计算机与信息技术学院 |
| 专 业： | 计算机科学与技术 |
| 指导教师： | 吴为民 |
| 班 级： | 计科1602 |
| 姓 名： | 麻锦涛 |
| 学 号： | 16281262 |

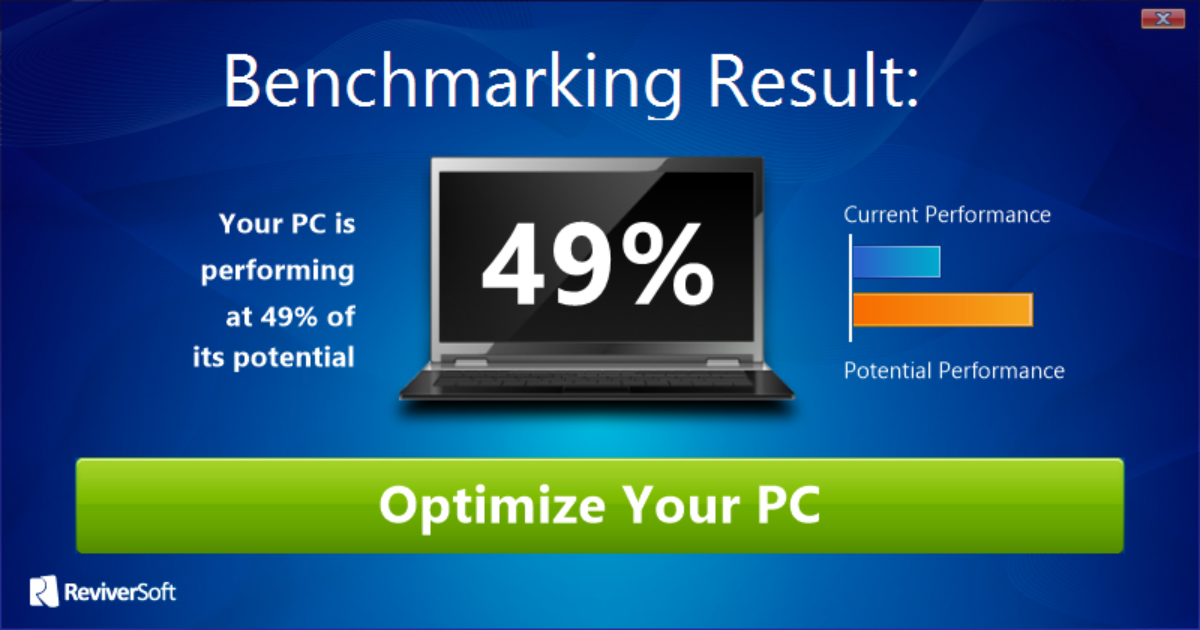
2019年3月16日

**题目1：PC性能测试软件**

请在网上搜索并下载一个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），并对你自己的电脑和机房电脑的性能进行测试。并加以比较。

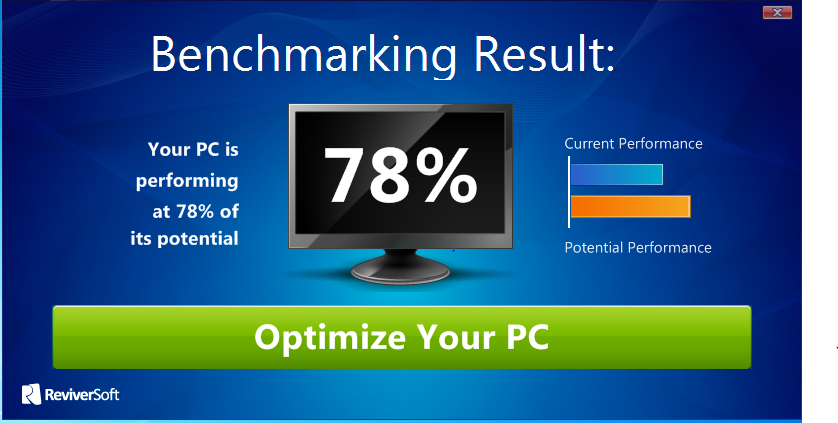
**我分别用PC 性能 benchmark和鲁大师两款软件做了对比，下面是实验过程及结果：**

1. **我的电脑：**





1. **机房电脑：**





**单独就评估软件**鲁大师**对两台电脑性能进行比较**

|  |  |  |
| --- | --- | --- |
|  | 我的电脑 | 机房电脑 |
| 处理器性能 | 31851 | 45688 |
| 显卡性能 | 11586 | 11916 |
| 内存性能 | 8247 | 9888 |
| 磁盘性能 | 8580 | 2100 |

由上面跑出的结果，可见机房电脑的性能在处理器，显卡，内存性能上要高于我的电脑，但我的电脑的磁盘性能要高于机房的。

综上分析：分析pcbenchmark所得数据为电脑的current performance与其potential performance的比值，值大表明计算机目前运行良好，性能好，由测试结果数据可得比较出机房的电脑当前运行的性能更好。分析鲁大师性能测试结果：我的电脑得分60264机房电脑得分69592，通过分析我们可以得出CPU占总得分的比重最大，表明了其对计算机性能的影响是最大的，其次显卡性能和内存性能也很关键，另外机房的电脑内存性能较弱，所以拉低了整体得分，机房电脑各项得分基本上均领先于我的电脑，可以得出机房电脑性能更好的结论。

**题目2：toy benchmark的编写并测试**

可用C语言编写一个程序（10-100行语句），该程序包括两个子程序，两个子程序完成相同的功能，但一个子程序主要采用整数操作完成，另一个子程序主要采用浮点操作完成。两个子程序执行的频率（频率整数，频率浮点）可调整。请在你的计算机或者在机房计算机上，以（0.5，0.5），（0.1，0.9），（0.9，0.1）的频率运行你编写的程序，并算出三种情况下的加权平均运行时间。

下面给出两个例子作为提示：

**提示1：比如求某个浮点数（小数点后的位数固定）的若干次幂。你可以采用：**

**（1）整数方法：先将浮点数转换成整数（乘以10的若干次方），然后求整数的幂，最后再将结果转换成浮点数（除以10的若干次方）。**

**（2）浮点方法：直接计算double数的幂，该方法会比第一种方法用更长的时间。**

**注意：**

**（1）两种方法分别作为子程序写在一个程序中。两种方法的子程序都需要运行多次才能看出差距。比如，对于频率（0.1，0.9），你可以将第一个子程序运行1000次，第二个子程序运行9000次，然后看总的运行时间。**

**（2）由于整数表示范围有限，所以第一种方法的运算结果可能因超过整数表示范围而不正确。不过，这里只关注运行时间，运行结果可不考虑。**

**提示2：再比如求一定精度的π值。你可以采用：**

**（1）级数法：利用如下公式：**

**π/4 = 1 -1/3 + 1/5 - 1/7 + 1/9 .....**

**（2）概率法：在边长为1的正方形中画半径为1的内切圆。随机产生0-1范围内一定数量的坐标点，落入内切圆的个数和落入整个正方形个数之比即为π/4的近似值。当然，你要用到C语言中求伪随机数的功能。**

**实验过程及结果：**

一、按照题目要求编写程序，代码如下

#include <stdio.h>

#include <time.h>

int main ()

{

int x,y,i,a;

double b;

clock\_t start,end;

printf("please input the times of these two parts!\n");

scanf("%d%d",&x,&y); /\*控制两部分的运行频率\*/

start=clock();

for(i=0;i<x;i++) /\*整数操作\*/

{

a=1234567\*5678123;

}

end=clock();

printf("int time is %f\n",(double)(end-start)/CLK\_TCK);

start=clock();

for(i=0;i<y;i++) /\*浮点操作\*/

{

b=12345678.5678\*56781230.1234;

}

end=clock();

printf("double time is %f\n",(double)(end-start)/CLK\_TCK);

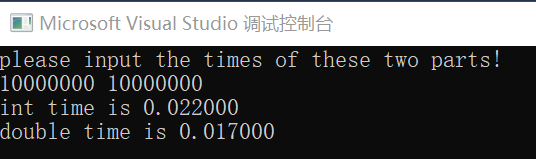
return 0;

}

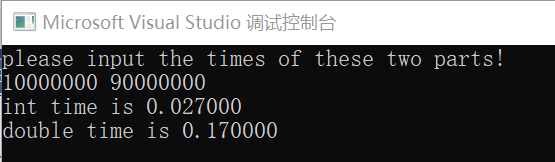
二、运行程序

分别输入符合（0.5，0.5），（0.1，0.9），（0.9，0.1）频率的次数，得到运行结果，如下图所示

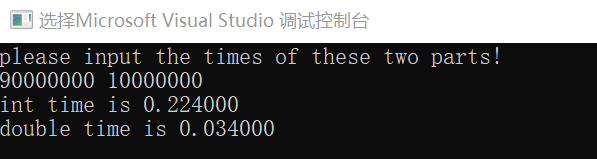
1.频率符合（0.5，0.5），为了使double型的时间变量顺利表示出两个简单操作的运行时间，输入的次数可尽量大



2.频率符合（0.1，0.9）



3.频率符合（0.9，0.1）

****

三、运行结果计算与分析

根据加权算数平均值的计算公式：操作A\*权值A+操作B\*权值B，分别计算三种频率不同（即权值不同）情况下的加权算数平均值得到三种情况下的加权平均运行时间：（单位均为S）

1. 在频率为（0.5，0.5）的情况下，加权平均运行时间为0.022\*0.5+0.017\*0.5=0.0195

2. 在频率为（0.1，0.9）的情况下，加权平均运行时间为0.027\*0.1+0.017\*0.9=0.018

3. 在频率为（0.9，0.1）的情况下，加权平均运行时间为0.224\*0.9+0.034\*0.1=0.205

分析：由于程序运行的不稳定性导致时间上的微偏差，但还是可以分析得到，**时间较长的操作所占权重相对较大的情况下，加权平均运行时间会偏大，反之亦反。**

**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)。

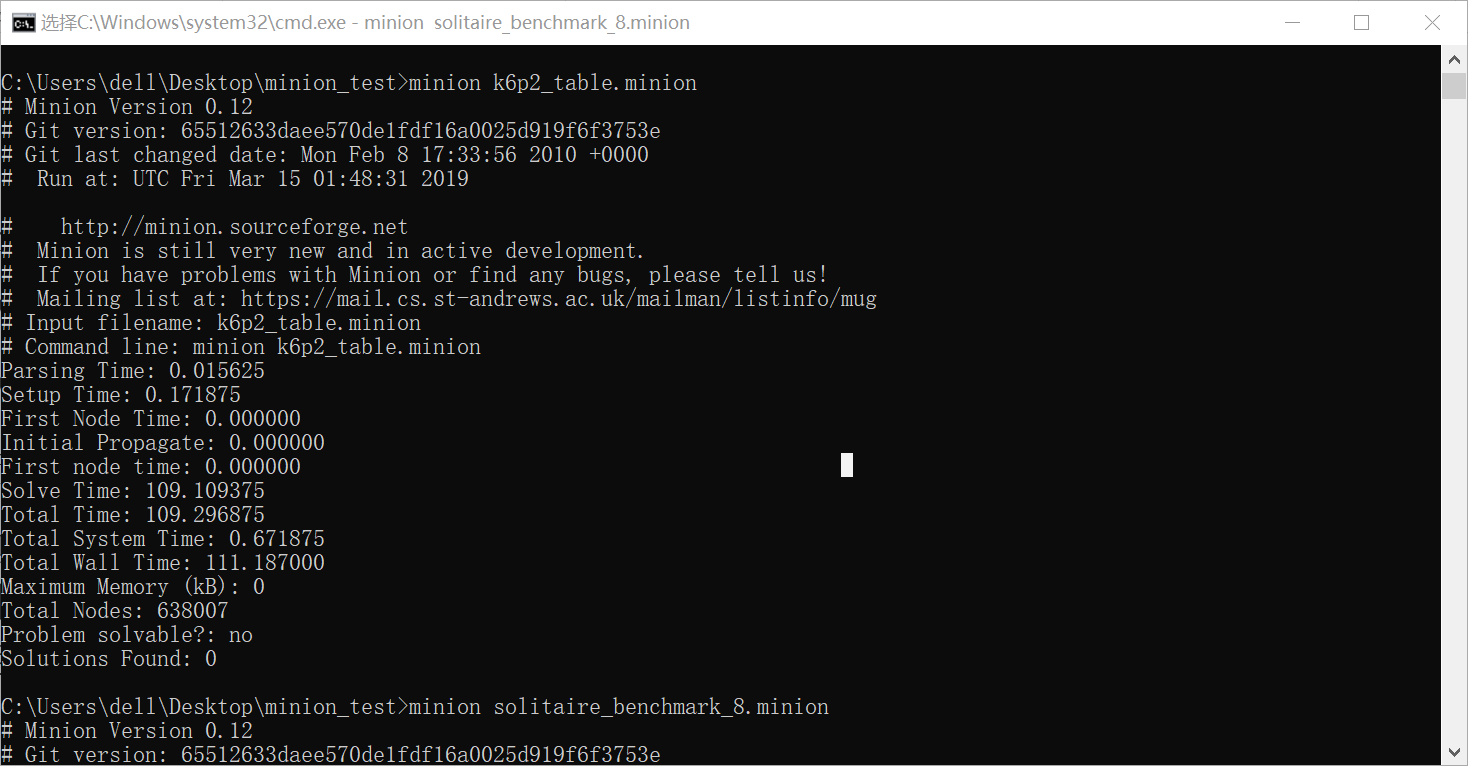
2. 进入你的程序目录，其中有可执行程序minion.exe，以及两个测试实例。

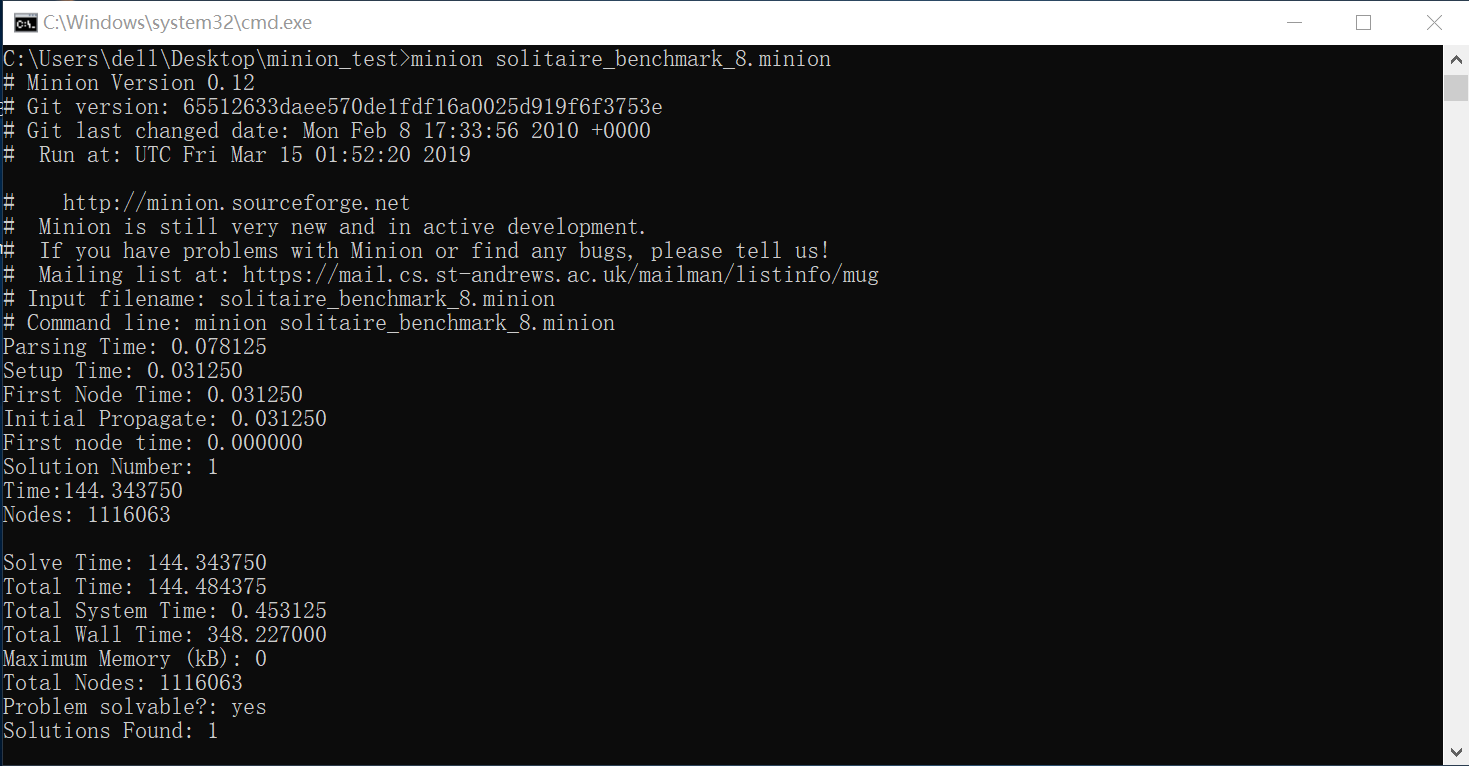
3. 运行：*minion 某实例*。（需要运行一段时间，比如5分钟，你要耐心等待）

请分别以2个测试实例为输入，分别在你自己的计算机和机房计算机上运行，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。

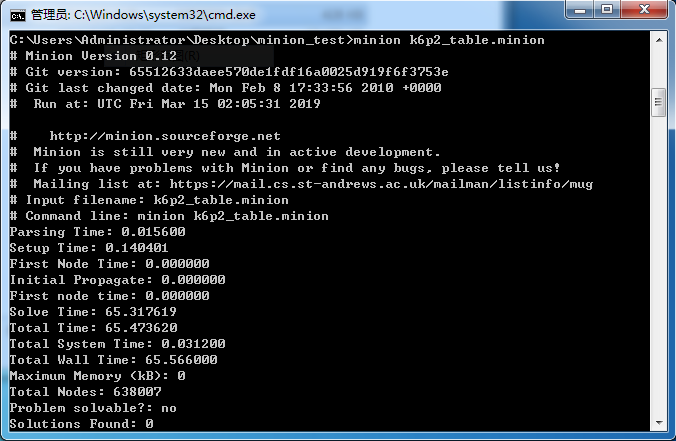
**注意：同学们的笔记本大都是64位的，有的程序可能运行不了，如minion就未必能运行。如果不能运行，建议你给自己的电脑安装一个虚拟机（如VMware），再在虚拟机中安装一个32位的windows系统。**

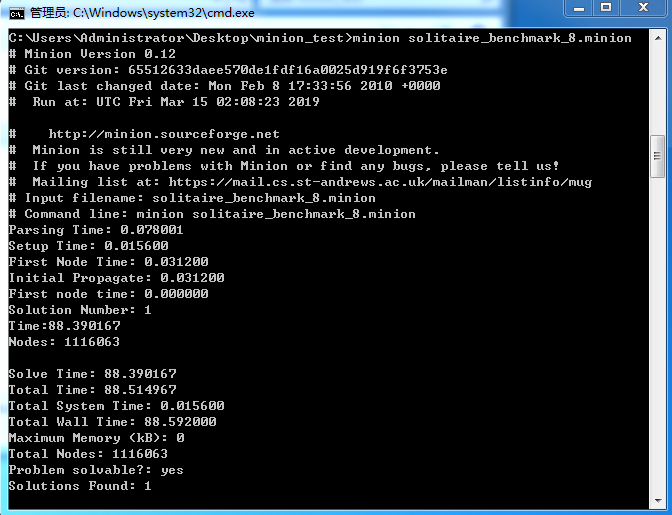
**自己电脑运行截图：**





**机房电脑运行截图**





二、运行结果计算与分析

根据实例运行结果得到：（数据单位均为S）

在我的电脑上，实例1的Total Wall Time是111.187；实例2的Total Wall Time是348.227；在机房电脑上，实例1的Total Wall Time是65.566；实例2的Total Wall Time是88.592；

根据归一化概念，参考机的选择不影响计算结果。在这里我选择机房电脑为参考机，

即机房电脑两个实例的运行时间分别为1、1，几何平均值为1；

相应得出我的电脑两个实例的运行时间分别为1.696、3.931，几何平均值为2.582。

**比较两个几何平均运行时间：1和2.582，可以得到机房电脑运行实例耗时更短，性能更好。**

（为方便计算和记录，以上值均进行四舍五入保留到小数点后三位。）

分析：这个实验的两台机子性能相差比较远，两个程序的运行都以同样的方向拉开了很大的距离，所以其实不用处理分析数据便可得到两台机子的性能比较结果。但遇到其他复杂的情况，当一台机子的几个程序与另一台的几个程序相比得到的差别类型也不同时，归一化处理就能体现它的价值。利用归一化处理，选择一个座位参考机，只需要计算处理另一个机子的几个程序运行时间，计算其几何平均值，得到一个结果与参考机的几何平均值对比即可，即把多个程序之间多个数据的比较化作两个数据之间的比较，简单明显地得出结论。