编译原理课程实验报告

**实验一 程序语言认知实验**

指导教师：王贵珍老师

班 级：07111507

学 号：1120151880

姓 名：廖汉龙

邮 箱：[liaohanlong@outlook.com](mailto:liaohanlong@outlook.com)

2018 年 3 月 2日

一、实验目的

了解程序设计语言的发展历史，了解程序设计语言的多样性，了解不同程序设计语言的各自特点；感受编译执行和解释执行两种不同的执行方式，初步体验语言对编译器设计的影响，为后续编译程序的设计开发典型良好基础。

二、实验内容

给定一个特定的功能，分别使用 C/C++、 Java、 Python、 Haskell 和 MIPS/X86汇编实现该功能，对采用这几种语言实现的编程效率，程序的规模，程序的运行效率进行对比分析。例如分别使用上述几种语言实现一个简单的矩阵乘法程序，输入两个矩阵，输出一个矩阵，并分析相应的执行效果。

三、实验步骤

**1.计算机系统硬件配置情况**

|  |  |
| --- | --- |
| **配置** | **参数** |
| CPU核心数 | 4（Inter(R)Core(TM) i5-4200H CPU） |
| CPU主频 | 2.8GHz |
| 已安装的内存（RAM） | 8G |
| Cache（三级缓存） | 256KB-1.0MB-6.0MB |
| 操作系统 | Window 10 64位操作系统 |

**2.程序设计语言的开发环境配置**

|  |  |  |
| --- | --- | --- |
| **语言** | **编译器/解释器版本** | **官方网站** |
| C/C++ | Vistual Studio code | https://www.visualstudio.com/ |
| Java | Java 1.8.0\_141 | http://www.oracle.com/ |
| Python | Python 3.6 | https://www.python.org/ |
| Haskell | Haskell 8.2.2 | https://www.haskell.org/ |
| X86汇编 | Vistual Sdudio 2015  Masm32 | https://www.visualstudio.com/  http://www.masm32.com/ |

**3.主要实现代码片段**

（1）C/C++ 主要代码

|  |
| --- |
|  |

(2) Java 主要代码

|  |
| --- |
|  |

（3）Python 代码

|  |
| --- |
|  |

（4）Haskell 主要代码

|  |
| --- |
|  |

（5）x86 汇编主要代码

|  |
| --- |
|  |

**4.运行结果截图与过程说明**

整个实验有C++代码生成了三份txt文件中的数据是由空格分开的随机整数，并且三份文件对应的数据量分别是10000，50000，100000。所有测试性能的代码都读入这三份文件进行性能测试与对比。

实验过程中，计时部分不包括数据的读入时间，只测试在读入数据之后，进行对数据排序的时间，下面是各代码的程序运行结果截图。在实际实验过程中，运行了10次取平均值为最后的测试结果。

（1）C/C++ 运行结果截图

|  |
| --- |
|  |

(2)Java 运行结果截图

|  |
| --- |
|  |

(3)Python 运行结果截图

|  |
| --- |
|  |

(4)Haskell 运行结果截图

|  |
| --- |
|  |

(5)x86 汇编 运行结果截图

|  |
| --- |
|  |

**5.程序规模对比**

|  |  |  |
| --- | --- | --- |
| 语言 | 代码行数 | 工程环境 |
| C/C++ | 62 | 有 |
| Java | 76 | 有 |
| Python | 41 | 无 |
| Haskell | 28 | 无 |
| X86 汇编 | 284 | 无 |

由上述可知，单从程序的规模来看（忽略个人编码风格差异），x86 汇编代码的行数远远大于其它几类语言，不需要工程环境。在高级语言中，程序语言规模由高到低依次是java, C/C++， Python, Haskell，由于Haskell语言的特性，其为函数式编程，所以在递归排序等涉及函数操作的程序中，相对的规模小了很多，Haskell语言在编写矩阵相乘时候表现就不那么好了。其中值得一提的是Python, 由于采用接近自然语言的编程方式，并且整个语言的格式采用的是缩进，所以，其代码的规模要小很多。并且Python和Haskell语言代码是不需要工程环境的配置的。

**6.语言易用性对比**

针对C/C++、Java、Python、Haskell, x86五种种语言，本人对于他们的易用性排序为（C/C++）>= (Python) >= (Java) > (Haskell)>(x86 汇编)

x86 汇编作为汇编语言，处理起来是非常繁杂的。Haskell是公认的较为难用的程序语言，但是由于其是函数式的原因，所以在特定的领域有很好的作用。其它三个高级语言是就个人掌握情况而定的。很多高校都有分别以其中一门语言作为信息学专业的第一门编程语言。

**7.代码性能对比**

|  |  |  |
| --- | --- | --- |
| 数据规模 | 语言 | 10次平均运行时间（ms） |
| 10000 | C/C++ | 2.00 |
| Java | 90.86 |
| Python | 83.51 |
| Haskell | 130.88 |
| X86 汇编 | 0.76 |
| 50000 | C/C++ | 23.80 |
| Java | 857.29 |
| Python | 456.74 |
| Haskell | 747.54 |
| X86 汇编 | 12.11 |
| 100000 | C/C++ | 105.10 |
| Java | 4828.90 |
| Python | 966.44 |
| Haskell | 1535.17 |
| X86 汇编 | 43.79 |

由以上表格记录可知，相对来说，x86汇编语言和C/C++语言的运行效率要远远优于其它几种语言。而x86汇编语言又大幅优于C/C++。这可以解释一些大型的游戏都是使用C/C++开发的，而有些对速度要求非常高的程序则会嵌入汇编程序。

Python程序和Haskell在运行速率上相差不大，但是经过比较其他的一些数据，比如运行大整数的运行，Haskell的运行速度却要优于Python。Python的作为一种脚本语言的优势非常明显，简单易懂，占用内存低，现在各个领域运用非常广泛。

Java 相对则要慢很多，阅读了网上一些文档，发现，Java的运行速度与代码的简介性一直被大家所诟病，但是目前其确实应用最为广泛的程序语言，这得益于其在Android和Web方面的应用。

四、实验心得体会

通过本次实验，我接触了平时较为少用的汇编语言和从来没有接触Haskell程序语言。并且通过做了归并排序的实验，初步对各种程序语言的运行效率，易学易用等房买你有了直观的认知。

本次实验遇到的困难主要在于入门Haskell语言以及使用汇编语言完成归并程序的编写。

我认识到各种语言各有风格，比如Python，其接近自然语言的特性深受大家的欢迎。C语言是面向过程的语言，而C++，Java，Python等都是面向对象的语言，而在Haskell中一切都是函数。

我也认识到计算机专业是一个需要终身学习的学科，但是就程序语言来说，我现在所接触与了解到的仅是九牛一毛，应当不断的提高自主学习的能力。

五、附录

使用C++语言编写的随机数生成。

