**主题：高级语言的特点描述**

**任务：每种高级语言的是如何和计算机交互的，每种语言的擅长之处，为什么擅长**

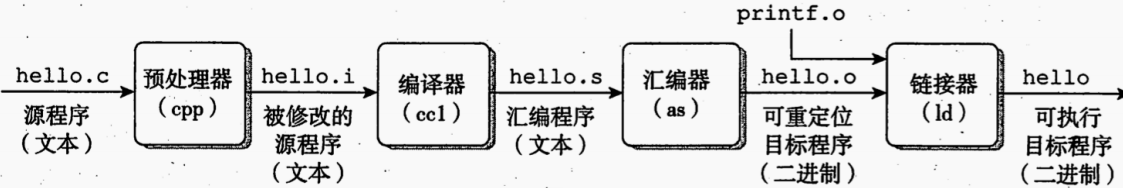
1. **C/C++**

C是一门纯面向过程编程的语言，C++在C的基础上增加面向对象的有关内容与规则。

1、实现原理

通过将源程序翻译成可执行目标程序后直接由计算机的操作系统来运行，具有较强的平台相关性，属于编译型语言。

C与C++由源文件到可执行目标程序的过程如下



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 将#include的系统文件加入到程序文件中 |  | 将C语言翻译成汇编语言 |  | 将汇编语言翻译成机器语言 |  | 合并库函数文件与源文件 |  |

2、特点

C/C++无需通过虚拟机执行，所以它们较高的性能，更加贴近底层，许多操作系统与office是由C/C++编写而成。

1. **Java**

Java是一种面向对象语言

1、实现原理

Java首先通过前端编译将Java源文件编译为字节码文件（class文件），再由Java虚拟机JVM来执行字节码文件。

（1）Javac

其中前端编译通过Javac完成，其过程如下

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| Java字符流 | 词法分析 | 记号流 |
| 记号流 | 语法分析 | 语法树 |
| 语法树 | 语义分析 | 注解抽象语法树 |
| 注解抽象语法树 | 字节码生成器 | 字节码（class文件） |

（2）JVM

得到class文件后，JVM加载class文件并得到相应执行结果。

JVM主要由三个功能系统构成

* 类加载系统

加载必要的class文件到JVM内存中

* 执行引擎子系统

读取并执行所加载的class文件

Ps：这一步使用了JIT编译，JIT编译主要由C1，C2两个编译器组成，C2的功能较为复杂，在此不做详述，C1编译器则对字节码不断进行优化，依次得到HIR（高级中间代码表示）、LIR（低级中间代码表示），然后进行分配寄存器等工作最后得到机器码。

* 垃圾收集子系统

自动释放没有用的对象

JVM内存结构

* 程序计数器

帮助进程切换

* 栈

方法被调用时创建，用于存储局部变量表、操作栈、动态链接、操作出口等

* 堆

存放对象实例

* 方法区

存储类信息、常量、静态变量、即时编译器编译后的代码等

2、特点

由Java的实现原理，可以得出其相对C++的实现方式具有以下优点

1. 开发效率高

面向对象语言，易维护，不必由程序员进行垃圾回收等。

1. 平台无关性

“一次编译，处处运行”，class文件是平台无关的而JVM则是平台相关的，在不同的平台提供一致的JVM虚拟指令集，一致的操作与访问形式，由此实现“一次编译，处处运行”。

1. 特殊的优化

可以根据运行时性能监控进行一些特殊的优化，如调用频率预测。

Java的开发效率是以性能上的劣势为代价的，JIT编译器占用用户程序运行时间，而它的动态检查与动态优化等消耗许多时间与资源。

其带来的具体性能劣势如下：

1. JIT编译占用用户程序运行时间，难以引入大规模的优化技术；
2. 动态安全要求频繁进行动态检查，时间消耗大；
3. 虚方法使用频率大于C/C++，提高了编译器优化难度；
4. Java可以进行动态扩展，有些全局优化难以进行，运行时变化引起撤销或重新进行一些优化；
5. 对象分配在堆上，需要垃圾收集器进行回收资源管理，垃圾回收器自身占用资源；
6. python

1、实现原理

python的实现原理与Java类似，python源文件首先被翻译为字节码文件，然后通过python虚拟机PVM执行。

2、特点

python语言自身具有简洁易懂的特点，而且相对其他语言python具有丰富的基础工具与类库，使用便利。

python的PVM相比JVM更加灵活与抽象，相对来说性能的损失也更大。

四、**参考**

[[零] Java 语言运行原理 JVM原理浅析 入门了解简介 Java语言组成部分 javap命令使用](https://www.cnblogs.com/noteless/p/9510252.html)

<https://www.cnblogs.com/noteless/p/9510252.html>

[浅谈JVM编译原理->.java文件转变为.class文件的过程](https://www.cnblogs.com/pengx/p/10330455.html)

<https://www.cnblogs.com/pengx/p/10330455.html>

Java 源代码编译成 Class 文件的过程分析

<http://www.codeceo.com/article/compile-java-source-code-into-a-class-file.html>

Java编译（三）Java即时编译(JIT编译)：运行时把Class文件字节码编译成本地机器码

<https://blog.csdn.net/tjiyu/article/details/53948009>

[JVM的粗略简述](https://www.cnblogs.com/wade-luffy/p/5752533.html)

<https://www.cnblogs.com/wade-luffy/p/5752533.html>

Python解释执行原理

<https://www.cnblogs.com/dj0325/p/7966182.html>

基于Java与Python的面向对象编程的基本特征研究\_韩宏峰

Python语言的探讨\_史梦楚

浅析面向对象语言C\_与Java区别\_王艳娟

面向对象程序计的平台\_C\_与JAVA的比较\_高建州