它摆脱了硬件平台的束缚,实现了"一次编写,到处运行"的理想

Java优点

提供了一个相对安全的内存管理和访问机制,避免了绝大部分的内存泄露和指针越界问题

它实现了热点代码检测和运行时编译及优化,这使得Java应用能随着运行时间的增加而获得更高的性能

它有一套完善的应用程序接口,还有无数来自商业机构和开源社区的第三方类库来帮助它实现各种各样的功能.....

Java程序设计语言

各种硬件平台上的Java虚拟机

Class文件格式

JavaAPI类库

来自商业机构和开源社区的第三方Java类库





Java Card: 支持一些Java小程序(Applets)运行在小内存设备(如智能卡)上的平台

Java技术体系 可以分为4个平台 Java ME (Micro Edition): 支持Java程序运行在移动终端 (手机、 PDA)上的平台,对JavaAPI有所精简,并加入了针对移动终端的支持,这个版本以前称为J2ME。

Java SE (Standard Edition): 支持面向桌面级应用(如Windows下的应用程序)的Java平台,提供了完整的Java核心API, 这个版本以前称为J2SE。

Java EE (Enterprise Edition):支持使用多层架构的企业应用 (如ERP、CRM应用)的Java平台,除了提供Java SE API外,还对其做了大量的扩充ill并提供了相关的部署支持,这个版本以前称为J2EE。



实战:自己编译JDK:

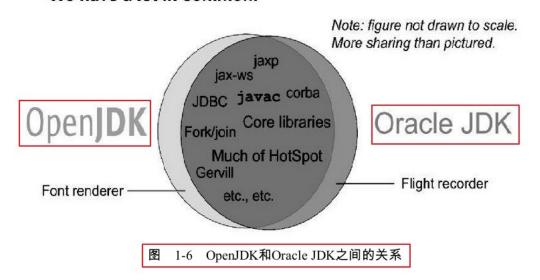
考虑到Sun系列的JDK是现在使用得最广泛的JDK版本,笔者选择了OpenJDK进行这次编译实战。 在本次编译实践中,笔者选用的项目是OpenJDK7u,版本为7u6

1、获取JDK源码

OpenJDK和Sun/OracleJDK之间的关系与区别:

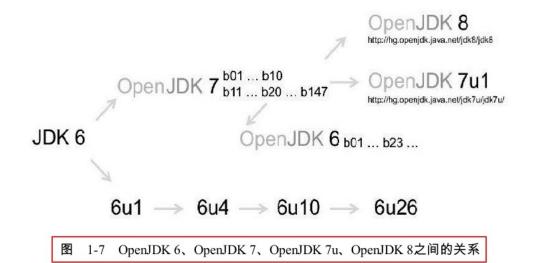
OpenJDK 7和Oracle JDK 7在程序上是非常接近的,两者共用了大量相同的代码,所以我们编译的OpenJDK,基本认为性能、功能和执行逻辑上都和官方的Oracle JDK是一致的。

"We have a lot in common."



OpenJDK 6、OpenJDK 7、OpenJDK 7u和OpenJDK 8等项目之间的关系:

OpenJDK7是始于JDK6时期,当时JDK6和JDK6 Update 1已经发布,JDK7已经开始研发了,所以OpenJDK7 是直接基于正在研知 但考虑到OpenJDK7的状况在当时还不适合实际生产部署,因此在OpenJDK7 Build 20的基础上建立了OpenJDK6分支,剥离掉JI 个可以通过TCK6测试的独立分支。



在本次编译实践中,笔者选用的项目是OpenJDK7u,版本为7u6

	获取OpenJDK源码的方式
方式1	通过Mercurial代码版本管理工具从Repository中直接取得源码(Repository地址: http://hg.openjdk.java.net/jdk7u/jdk7u) 获取过程如以下代码所示: hg clone http://hg.openjdk.java.net/jdk7u/jdk7u-dev cd jdk7u-dev chmod 755 get source.sh . get_source. sh
方式2	直接下载官方打包好的源码包,读者可以从Source Bundle Releases页面(地址: http://jdk7.java.net/source.htm1)取得打包好的源码

未完......