文章编号:1005-8451(2005)02-0030-04

Java编程中的中文乱码问题的研究

赵 陬,魏慧琴
(北京交通大学 计算机科学与信息技术学院,北京 100044)

摘 要:在目前解决Java编程中的中文乱码问题技术的基础上,提出了通过分析Java程序设计中Java编译器对Java源文件和JVM Java Virtual Machine,对字节码文件的编码/解码过程,以便正确而及时地预测、发现和检查问题的新思路,并给出相应的解决问题的方法。经过大量开发实践表明,该思路可以准确找出问题产生的原因并解决问题,从而提高程序设计的效率。

关键词:中文乱码;Unicode;GB2312;GBK;jspSmartUpload
中图分类号:TP39 **文献标识码**:A

Research on Chinese incorrect codes of programming in Java

ZHAO Yang, WEI Hui-qin
(School of Computer Science and Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: Based on the present technology used in the resolution of Chinese incorrect codes of programming in Java, an original approach of predicting, finding the problems exactly by analyzing how does javac complier encode and decode Java Source Code and how does JVM(Java Virtual Machine) encode and decode ByteCode file, as well as the resolution was given. As the experiments show, this approach could found finding problem, quickly and resolved problems of Chinese incorrect codes better and improved efficiency of programming.

Key words : Chinese incorrect codes; Unicode; GB2312; GBK; jspSmartUpload

Java编程语言具有面向对象、可跨平台运行和分布应用等特点,因此使得Java语言成为主流的网络编程语言。但是在基于Java编程语言的应用开发中,经常碰到中文乱码问题。例如,浏览基于JSP技术的网站看到的是乱码、带有中文汉字的JSP文件打开后看到的也是乱码、Java程序和数据库之间数据传递出现乱码等,给开发者和使用者带来了很大的不便。因此,我们需要了解Java编程时中文问题产生的原因,以找到解决Java中文问题的方法。但是,由于Java技术涉及内容非常广,面向Java的Web服务器、应用服务器以及JDBC数据库驱动等都没有官方的标准,所以Java应用在处理中文过程中所出现的中文问题具有多变性,增加了问题的复杂度。这就需要有一种方法,能够快速地发现和检查问题,并能准确地解决问题。

1 计算机常用字符集及汉字字符集

在计算机中,字符是以编码来表示的。每个字符使用哪种编码表示,取决于所使用的字符集。

ASCII字符集是Internet上最初使用的字符集,它使用7 bits来表示一个字符,总共表示128个字符,其中包括了英文字母、数字和标点符号等常用字符。之后,又进行了扩展,使用8 bits表示一个字符,可以表示256个字符,主要在原来的7 bits字符集的基础上加入了一些特殊符号,例如制表符。

Unicode字符集固定使用16 bits(2 bytes)来表示一个字符,共可以表示65 536个字符。该字符集将几乎所有语言的常用字符收录其中,方便了信息交流。Unicode字符集有多种编码形式。标准的Unicode称为UTF-16,后来为了双字节的Unicode能够在现存的处理单字节的系统上正确传输,出现了UTF-8,使用类似MBCS的方式对Unicode进行编码。UTF-8是一种Unicode编码,它属于Unicode字符集,即它的编码的字符集和Unicode是一致的,但编码的方式不一样。UTF-8是以8bits即1 byte为编码的最基本单位,被广泛应用在文件存储和网络传输中。

GB2312-80是在国内计算机汉字信息技术发展初始阶段制定的,其中包含了大部分常用的一、二

收稿日期:2004-06-09
作者简介:赵 陬,在读硕士研究生;魏慧琴,副教授。

级汉字和 9 区的符号。该字符集是几乎所有的中文系统和国际化的软件都支持的中文字符集，是最基本的中文字符集。

GBK 是 GB2312-80 的扩展，是向上兼容的。它包含了 20 902 个汉字，其编码范围是 0x8140～0xFEFE，剔除高位 0x80 的字位，其所有字符都可以一对一映射到 Unicode。

1.1 中文问题的来源

目前，大多数国际性的软件内部均采用 Unicode 编码，在软件运行时，它获得本地操作系统默认支持的编码格式，然后再将软件内部的 Unicode 转化为本地系统默认支持的格式显示出来。Java 的 JDK 和 JVM 即是如此。Java 语言内部是用 Unicode 表示字符的，所以在 Java 程序运行时，就存在着一个从 Unicode 编码和对应的操作系统及浏览器支持的编码格式转换的问题，这个转换过程通过一系列步骤完成，如果其中任何一步出错，则显示出来的汉字就会是乱码，这就是我们常见的中文问题。

1.2 Java编程转换的详细过程

Java 程序的执行过程是："编写源程序代码 -> Java 文件；Java 源代码-> Java 字节码；虚拟机->操作系统 -> 显示设备。"上述过程中的每一步骤，都必须正确地处理汉字的编码，才能够使最终的显示结果正确。

"编写源程序代码 ->Java 文件"，在 Java 编程过程中，需要对中文字符进行从 Unicode 到 GBK 编码的转换。

"Java 源代码-> Java 字节码"标准的 Java 编译器 javac 使用的字符集是系统默认的字符集，如在中文 Windows 操作系统上就是 GBK 编码，而在 Linux 操作系统上就是 ISO-8859-1，所以在 Linux 操作系统上编译的类中源文件中的中文字符都出了问题。

"Java 字节码->虚拟机->操作系统"，Java 运行环境（JRE）分英文版和国际版，但只有国际版才支持非英文字符。Java 开发工具包（JDK）支持多国字符，并非所有的计算机用户都安装了 JDK。很多操作系统及应用软件为了能够更好地支持 Java，都内嵌了 JRE 的国际版本，为自己支持多国字符提供了方便。

"操作系统 -> 显示设备"，对于汉字来说，操作系统必须支持并能够显示。英文操作系统如果不搭配特殊的应用软件的话，是不能够显示中文的。

1.3 解决方法

根据 Java 处理文件的原理，本文给出了一套解决问题的方法。

常见的 Java 程序中和用户直接交互、用于输入和输出字符的有：直接在 console 上运行的类、JSP 代码、Servelet 类 3 种。其具体的解决方案如下。

1.3.1 针对直接在console上运行的类

在程序编写时，如果从用户端接收可能含有中文的输入或中文的输出，程序中应该采用字符流来处理输入和输出，具体来说，应该用以下的面向字符型节点流类型：

对文件：FileReader FileWrieter

其字节型节点流类型为：FileInputStream，FileOutputStream

对内存 数组：CharArrayReader CharArrayWriter

其字节型节点流类型为：ByteArrayInputStream，ByteArrayOutputStream

对内存 字符串：String-Reader StringWriter

对管道：PipedReader PipedWriter

其字节型节点流类型为：PipedInputStream，PipedOutputStream

同时，应该用以下的面向字符型处理流来处理输入和输出：

BufferedWriter BufferedReader

其字节型的处理流为：BufferedInputeStream，BufferedOutputStrea

InputStreamReader OutputStreamWriter

其字节型的处理流为：DataInputStream Data-Output

Stream

其中 InputStreamReader和InputStreamWriter 是用于将字节流按照指定的字符编码集转换到字符流，例如：

```
InputStreamReader in = new InputStreamReader
(System.in,"GB2312");
OutputStreamWriter out=new OutputStreamWriter
(System.out,"GB2312");
```

另外，在编译程序时，通过指定确定的编码机制来实现其编译结果对中文的支持。例如，对于需要支持简体中文应用可以通过javac-encoding gb2312 Read.java来编译源程序。

1.3.2针对Servlet类

在编译Servlet类的源程序时，用-encoding指定编码为GBK或GB2312。

在对用户输出时，用response对象的setContent-Type("text/html;charset=GBK")或GB2312来设置输出编码格式。

在接收用户输入时，用request.setCharacterEncoing("B2312")。

这样，无论Servlet类移植到什么操作系统中，只要客户端的浏览器支持中文，就可以正确显示。

1.3.3 针对JSP代码

（1）在JSP源文件中指定其编码格式，以保证JSP编译器能正确地解码含有中文字符的JSP文件。其方法是：在JSP源文件头上加入<%@page pageEncoding="GB2312"%>或<%@page pageEncoding="GBK"%>90

（2）定义JSP页面的字符输出集，从而实现内码的自动转换，以保证JSP向客户端输出时是采用中文编码方式输出的。用法是，在JSP源文件头上加入<%@page contentType="text/html;charset=gb2312"%>。

（3）在JSP源文件头加入<%request.setCharacterEncoding("GB2312");%>，以保证JSP能正确获得传入的参数。

2 应用举例

我们已经利用以上的方法并结合具体的应用程序，解决了在开发几个大型网站（新闻出版署的扫黄打非网站、高等教育出版社的理工教育资源网）的过程中所出现的乱码问题。例如，在这些系统中，上传下载功能是必备的一个功能模块，开发过程中，使用jspSmartUpload上传下载组件来完成其功能。这个组件虽然能下载文件，但对中文支持不足。若下载的文件名中含有汉字，则浏览器在提示另存文件时，显示的却是乱码，影响了正常工作。这是因为浏览器在传递参数时默认是以UTF-8编码格式来传递，应用服务器返回给浏览器的内容是Unicode从而使得浏览器不能正确显示中文名字。所以我们对返回给浏览器的另存文件名进行了UTF-8编码。

jspSmartUpload组件的SmartUpload类是完成下载并保存功能的，在此类中增加了toUtf8String这个方法，直接利用Java语言提供的编码转换方法获得汉字字符的UTF-8编码，之后将其转换为%XX的形式。增加源码如下：

```
/**
     * 将文件名中的汉字转为UTF-8编码的字符串,以便下载时能正确显示另存的文件名.
     * @return 重新编码后的文件名
*/
publicstaticStringtoUtf8String(Strings)
    {
     StringBuffer sb = new StringBuffer();
    for(int i=0; i<s.length(); i++)
        {
      char c = s.charAt(i);
       if(c >= 0 && c <= "\377")
            {
        sb.append(c);
       } else{
        byte b[];
        try
            {
          b = Character.toString(c).getBytes("utf-8");
            }
        catch(Exception ex)
            {
         System.out.println(ex);
          b = new byte[0];
            }
       for(int j = 0; j < b.length; j++)
            {
        int k = b[j];
        if(k < 0)
          k += 256;
         sb.append("%"+ Integer.toHexString(k).toUpperCase());
            }
        }
    return sb.toString();
}
```

然后在downloadFile方法中引用该方法，将文件名返回给浏览器之前进行转换。改动部分如下：

```
public void downloadFile(String source
```

文章编号:1005-8451(2005)02-0033-03

旅客列车编组管理系统的开发与应用

李 红
(太原铁路分局 临汾站,临汾 041000)

摘 要:旅客列车编组管理系统的开发与应用,改变了以往旅客列车编组顺序表的人工作业模式,提高了生产效率和工作质量,减轻了作业人员劳动强度。主要阐述了该系统的设计目标、实现方法、系统功能及数据流程。

关键词:旅客列车编组;管理系统;应用;开发
中图分类号:U292.91;TP391 文献标识码:B

Development and application of Passenger Train Marshalling Management System

LI Hong
(Linfen Station of Taiyuan Railway Subadministration, Linfen 041000, China)

Abstract: With the development and application of Passenger Train and Management System, the manner of manual work was changed, the efficiency was increased, the quality was improved and the work intensity was reduced. It was described the objective, the method ,the function and the data flow of the System.

Key words: passenger train marshalling; Management System; development; application

随着计算机技术越来越广泛地应用于铁路运输管理,特别是全路 TMIS 的投入运用后,面临的问题就是维护和完善,以及进行后续开发,这样系统才能不断进步,提高性能,简化操作,拓宽系统资源的使用范围,系统才将更具有生命力。必将有大量信息管理系统进入生产领域。

收稿日期:2004-07-25
作者简介:李 红,助理工程师。

旅客列车编组管理系统的开发与应用,弥补了现有 TMIS 中只能管理货物列车编组而不能管理旅客列车编组的不足,改变了以往手工填写客车编组的作业方式,提高了作业效率及准确率。

1 系统简介

本系统的主要功能是对旅客列车编组信息进行

FilePathName,StringcontentType,StringdestFileName, intblockSize)
throws SmartUploadException, IOException, ServletException
{
toUtf8String(getFileName(s)));
}

根据以上方法,开发过程中所出现的中文问题已经得到了很好的解决。

4 结束语

Java 处理中文时所产生的问题,其根本原因就是由于被操作的中文字符(或变量)的编码格式与目标的编码格式不同造成的,所有这些问题都是发生在字符的输入、输出过程中。本文给出了 Java 在处理源程序过程中的详细转换过程,使得我们可以更好地通过分析来确定产生问题的环节,并给出了具体的解决方案。

参考文献:

[1] Allamaraju,S.(American) Professional Java Server Programming J2EE 1.3 Edition[M].Wrox Press,2003.
[2] Paul J.Perone,et al. Building Java Enterprise Systems With J2EE,[M].SAMS Publishing,2003.
[3] 段明辉. Java 编程技术中汉字问题的分析及解决[EB/OL],http://www-900.ibm.com/developerWorks/cn/java/java_chinese/index.shtml,2003-06-12