

基于 FLTK 多种编码字体的量宽式分行显示研究

李 凯¹ 饶天贵² 胡士杰¹ 曾 升¹

¹(株洲中车时代电气股份有限公司轨道交通技术中心 湖南 株洲 412001)

²(株洲中车时代电气股份有限公司数据与智能技术中心 湖南 株洲 412001)

摘 要 针对 FLTK (Fast Light Tool Kit) 控件自身不能对文本串自动分行显示的问题,提出一种可兼容 GB2312、UTF-8 等多种编码方式,根据测量预显示文本串在控件中所占像素宽度来分割文本显示的方法。通过与现有文本分行显示方法的对比分析,该方法在减少文本编辑工作量和提高文本分行显示效果上具有明显的优势,为文本智能分行显示技术提供一种新的参考。

关键词 文本分行 FLTK 量宽式 显示技术 UTF-8 GB2312

中图分类号 TP301.6 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2018.12.060

BRANCH DISPLAY OF WIDTH MEASURE METHOD BASED ON FLTK MULTIPLE ENCODING FONT

Li Kai¹ Rao Tiangui² Hu Shijie¹ Zeng Sheng¹

¹(Rail Transit Technology Center, Zhuzhou CRRC Times Electric Co., Ltd., Zhuzhou 412001, Hunan, China)

²(Data and Intelligent Technology Center, Zhuzhou CRRC Times Electric Co., Ltd., Zhuzhou 412001, Hunan, China)

Abstract Aiming at the problem that FLTK widgets could not automatically branch display the text string, we proposed a coding method compatible with GB2312, UTF-8, etc. The text display was segmented according to the measured pixel width of the pre-displayed text string in the widgets. Compared with the existing methods of text branch display, the method has obvious advantages in reducing the workload of text editing and improving the effect of text branch display. It provides a new reference for text intelligent branch display technology.

Keywords Text branch FLTK Width measure method Display technology UTF-8 GB2312

0 引 言

目前基于 Linux 系统的 FLTK 工具广泛应用于国内外机车、地铁等车辆显示器 UI 软件开发。FLTK 为 Linux 系统下轻量级的图形数据开发库^[1],开发的 UI 软件具有占用系统资源少、协议兼容性强、运行稳定可靠、可配置等优点^[2],所以目前在轨道交通领域使用广泛。跟随智能化的发展方向,显示器不仅要能实时显示车辆的基本信息,还要实时播报和记录车辆发生的故障情况,并且能向司机和维护人员给出每条故障可能发生的原因和处理措施。通常这类信息描述的文字很多,而每个界面中 FLTK 控件宽度有限,且显示的文

字不能自动换行,再加上不等宽字体每个字符所占的像素大小不同^[3],普通的分行方法会导致每行文字显示参差不齐,影响界面美观。较好的折行显示算法尤为重要,因此为了解决 FLTK 控件不能自动分行显示文字的问题,并针对不等宽字体,提出了一种兼容多种编码方式的自适应分行显示方法。

1 传统分行显示方法

1.1 固定字符数分行显示方法

将文本文件中的每条信息按固定字节数进行分割,然后分别在各控件中显示,是固定字符数分行显示

的主要处理思路,如图 1 所示。根据 FLTK 控件宽度,通过测试,可以找到分行的固定字符数值,可以保证固定宽度的控件能完整显示文字,但对于不等宽字体,比如汉字、数字、标点符号、英文等所占像素宽度并不相同,会使得相同字符数的不同文字显示得参差不齐,严重影响了界面的美观。

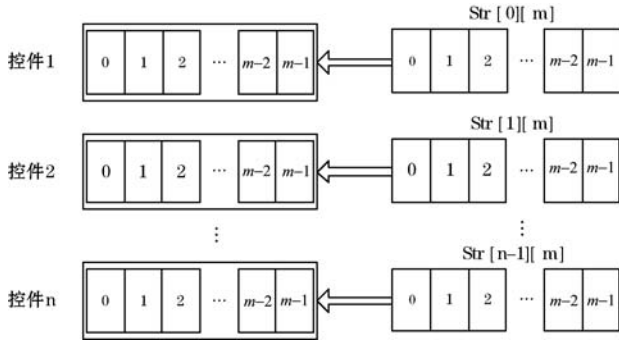


图 1 固定字符数分行处理方法

1.2 插入标签分行显示方法

该方法的处理思路是在文本文件中的每条信息适当位置上插入特定的字符,然后根据这些标签进行分割,最后分别在各控件中进行显示,如图 2 所示。根据 FLTK 控件宽度,通过多次测试,在要显示的文本串中插入特殊的分行符来进行差别化分行处理可以解决每行显示参差不齐的问题,但是对于上千条的信息,一一测试并在每条信息中插入分行符,工作量极大,效率极低,会严重影响项目交付进度。

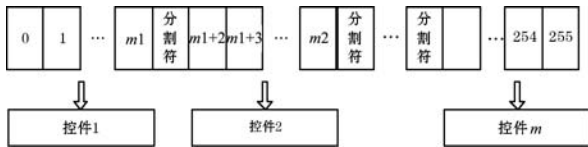


图 2 插入标签分行处理方法

1.3 其他分行处理方法

文献[3]给出了针对不等宽字体分行处理的基本思路,但是只针对 GB2312 编码、ASCII 编码字体的分行处理方法,分行处的英文单词未做分行处理,而是直接将整个单词另起一行,这样可能会导致每行显示参差不齐,影响显示效果。文献[4]在 Android 平台中设计了一种扩展的文本显示控件,可以实现对不等宽字体的分行处理,同样在兼容文字的多编码方式、英文分行处理上并未作相关处理。

2 分行显示方法设计

2.1 整体设计思路

该方法是针对单个 FLTK 控件无法显示大量文字(包含中英文、符号等)而作出的一种处理方法。是将

预显示的文字串分割成多个文字片段,再用多个相同的控件来显示,从而达到分行显示的效果。整体的文字串分割思路如图 3 所示。

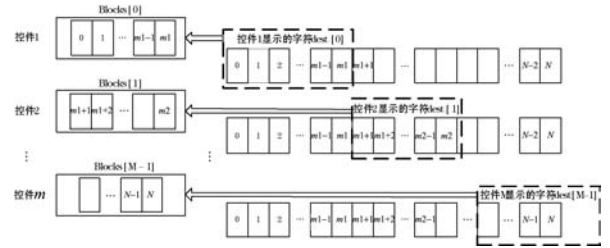


图 3 字符串分割再分行的处理思路

目前常用的字体,包括点阵字体和矢量字体,基本都是不等宽字体。不等宽字体的汉字、字母、标点符号等在控件中显示所占的像素宽度并不相等,使得相同宽度的控件能显示的字符总数不同,所以分配给每个控件显示的字符数需要灵活调整,即图 3 中每个虚线框所分割字符串并不相同。

2.2 针对多种编码的分割处理机制

2.2.1 基于 GB2312 编码的处理机制

基于常用的 GB2312 编码的分割处理机制,根据分割处字符所属文字类型的不同,可分为两种情况,如图 4 所示。如果分割处是汉字,根据 GB2312 编码规则,每个汉字占用 2 个字节^[5],则需回跳 2 个字节后截断字符串;如果分割处是字母、数字或者普通符号,根据 GB2312 编码规则,只占用 1 个字节^[5],只需回跳 1 个字节后截断字符串。

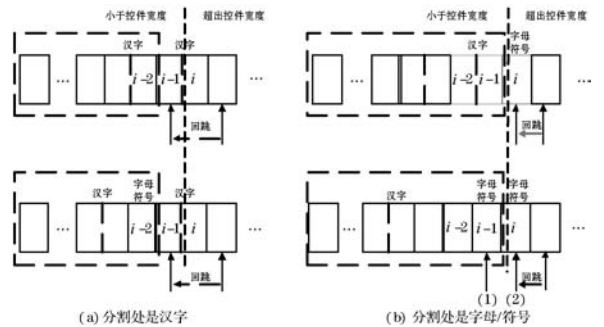


图 4 基于 GB2312 编码的分割处理机制

2.2.2 基于 UTF-8 编码的处理机制

UTF-8 编码是变长字节编码^[6-7],汉字、字母、数字、符号等所占字节数不相同,不正确的编码解析易导致乱码显示^[8-9],使得字符串分割处的情况变得相对复杂,如图 5 所示。

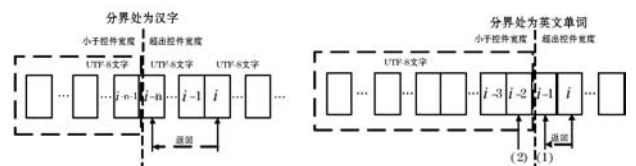


图 5 基于 UTF-8 编码的分割处理机制

1) 当分割处是汉字时,获取的该段字符数减 $n(n$

为分割处汉字所占字节数, $1 \leq n \leq 6$);

2) 当分割处是字母/字符时,分两种情况:(1) 一般情况下获取的该段字符数为能正常显示的字符数减 1。(2) 对于英语单词的截断换行,需考虑在行末添加短横线,这样获取的该段字符数需减 2;对于该行最后一个字符为单词首字母的情况,需将该字母移划分到下一个字段中,同时添加空格字符,这样获取的该段字符数也需减 2。

2.3 软件实现流程

具体实现的流程图如图 6 所示。

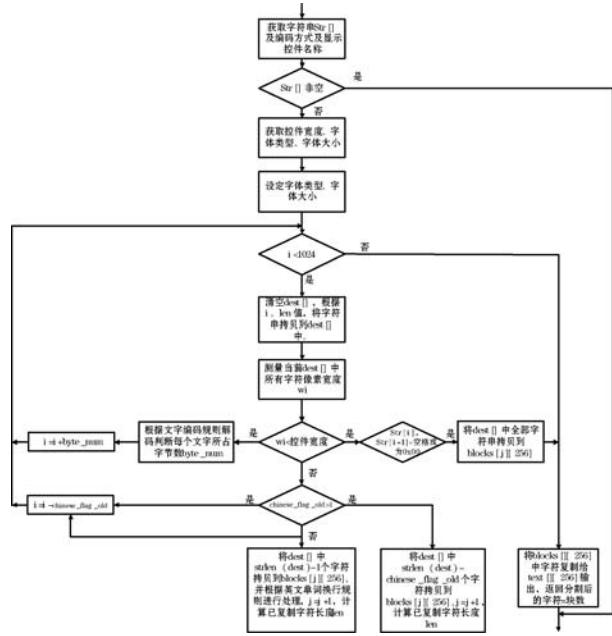


图 6 基于 FLTK 的字符分割处理流程

字符分割处理流程如下:

- 1) 判断预分割处理的字符串 $str[]$ 是否为空,若为空,则退出,否则向下执行。
- 2) 通过 FLTK 函数库中的函数,分别测量出显示字符的控件宽度 $widge_width$ 、字体类型 $font_index$ 、字体大小 $font_size$ 参数。
- 3) 通过 FLTK 函数库中的函数,以及步骤 2 测量的字体类型 $font_index$ 、字体大小 $font_size$ 参数来标定测量属性,这样后续测量的字符串所占像素宽度才是正确的。
- 4) 设字符串 $str[]$ 的下标为 i ,分段数为 j ,已拷贝至 $blocks[j][256]$ 的字符总数为 len 。如果 $i < 1\ 024$,根据 i, len 值将下标为 len 至 $i - len + 1$ (不包含)间字符拷贝至用于临时存取本次截取的片段的数组 $dest[]$ 中。
- 5) 使用 FLTK 函数库中函数测量当前 $dest[]$ 中所有字符所占像素值 wi 。
- 6) 比较 wi 与 $widge_width$,若 $wi < widge_width$,且

$str[i], str[i + 1], str[i + 2]$ 同时为空格,此时默认内容结束,则将 $dest[]$ 中所有字符拷贝至 $blocks[j][256]$ 中,并跳出循环,将 $blocks[j][256]$ 中字符串全部输出。若 $str[i], str[i + 1], str[i + 2]$ 不同时为空格,新的汉字标示 $chinese_flag$ 赋给上一个中文标示 $chinese_flag_old$,且根据文字的不同编码规则获取当前文字所占字节数 $byte_num$, $byte_num$ 赋值给 $chinese_flag$, $chinese_flag$ 等于 3 或 4 为汉字, $i = i + byte_num$ 。返回步骤 4。如果 $wi \geq widge_width$,则执行步骤 7。

7) 如果汉字标志 $chinese_flag$ 大于 1, $i = i - chinese_flag_old$,同时将 $dest[]$ 中前 $strlen(dest) - chinese_flag_old - 1$ 个字符拷贝至 $blocks[j][256]$ 中, $len = len + strlen(char_blocks[j])$, $j = j + 1$,返回至步骤 4,否则,执行步骤 8。

8) 如果汉字标志 $chinese_flag$ 小于等于 1, $i = i - 1$,如果同时 $dest[i - len - 1]$ 为字母, $dest[i - len]$ 不为“,”、“.”或空格,则 $dest[i - len]$ 变换为“-”, $len = len - 1$,即解决单词分割后,前半部分结尾处添加短横线的问题。如果汉字标志 $chinese_flag = 0$,同时 $dest[i - len]$ 为字母, $dest[i - len - 1]$ 为“,”或“.”或空格,则 $dest[i - len]$ 变换为空格, $len = len - 1$,即解决分割的单词前半部分仅有一个字母的显示的问题。将 $dest[]$ 中前 $strlen(dest) - 1$ 个字符拷贝至 $blocks[j][256]$ 中, $j = j + 1$, $len = len + strlen(char_blocks[j])$,返回至步骤 4。

以上介绍了该方法的整体思路、处理机制、具体实现流程。

3 实现效果与应用法

为了证明本文提出的文本分行显示算法在 FLTK 控件上实现的优越性,与目前应用较多的分行方法在文本编辑格式和显示效果上进行了比较。

固定字符数分行方法对文本格式举例如下(左右两项内容各固定分四行):

- 1001:1. X 出现故障原因 1;:1. 请检查 X1;.
- 1001:2. X 出现故障原因 2;:2. 请检查 X2;.
- 1001:3. X 出现故障原因 3;:3. 请检查 X3;.
- 1001:4. X 出现故障原因 4;:4. 请检查 X4;.

该方法需预先确定分行数目以及固定分行的字节数,文本编辑时需严格确保分行字符对齐。使用该方法的显示效果如图 7 所示,从图中可明显看到相同字符数分行有时会因为不等宽字体的差异导致行末参差不齐,影响美观。

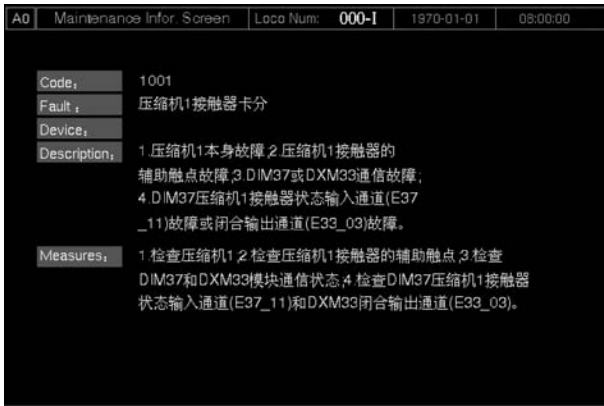


图7 固定字符数分行显示方法的显示效果

少乱码显示方面,也具备较强的能力。

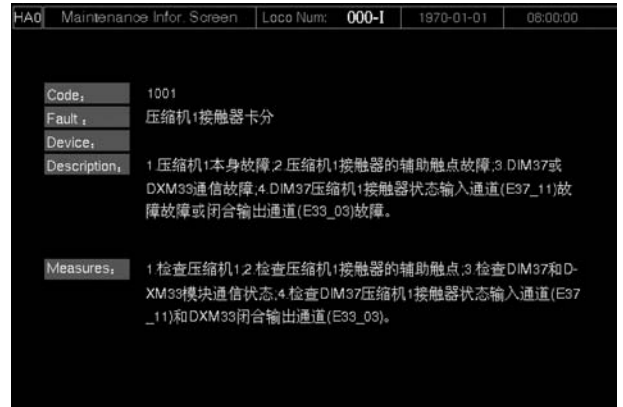


图9 量宽式分行方法的显示效果

文本中手动插入标签分行方法的文本编辑格式如下:

1001:1. X 出现故障原因 1;:2. X 出现故障原因 2;:3. X 出现故障原因 3;:4. X 出现故障原因 4。:

1001:X 故障导致的结果 1;: X 故障导致的结果 2。:

1001:司机对应的处理措施。:

1001:维护人员对应的处理措施。:

该方法需根据控件实际显示情况调整分行符标签位置,这样使得每条信息文本分行位置并不固定。该方法分行显示效果较好,但对文本编辑的工作量太大,如图8所示。

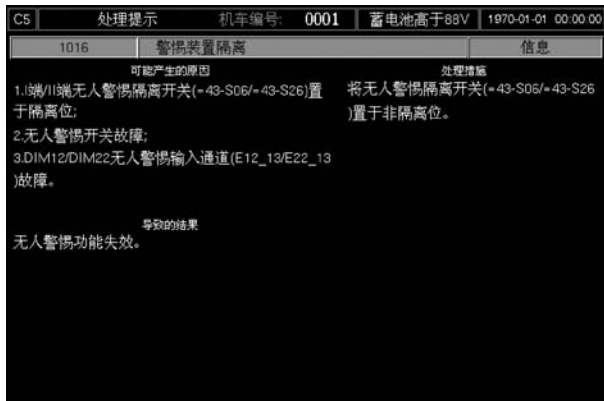


图8 文本中手动插入标签分行方法的显示效果

本文提出的新方法对文本编辑格式要求如下:

1001:1. X 出现故障原因 1;2. X 出现故障原因 2;3. X 出现故障原因 3;4. X 出现故障原因 4。:

1001:处理措施 1;处理措施 2;处理措施 3。:

从文本格式可以看出,只需要将每条文本信息单独在一行列出,无需添加任何分行标签或其他处理,明显降低了对文本格式的要求。量宽式的文本分行显示方法不仅大大减少了文本编辑工作量,而且在显示效果上也保持了较高的质量,行末文字参差不齐的现象基本解决,同时也具备对英文单词换行处理的能力,如图9所示。同时,该方法兼容多种文本编码格式,在减

通过对比分析,本文提出的新的量宽式分行显示方法,在有效降低文本编辑工作量和提高文本分行显示效果上,均得到了提高。

目前,该方法已在出口埃塞俄比亚电力机车等项目上进行了应用。

4 结 语

根据控件的像素宽度、字体类型、字体大小属性来对预显示的字符串进行自适应分割处理再分行显示的方法,不仅可以充分利用控件宽度大小来进行文字显示,而且不会出现文字的漏显问题,保证了文字显示的正确性、美观性。因为该方法只需要获取控件名称,就会自动获取其基本属性,并对需要显示的文字串进行自适应分割处理,使得对文本格式的要求降低,大大减少了对文本的处理时间;该方法还解决了每行显示参差不齐的问题,较好地保证了文本显示的美观性。该方法在对待编号分行处理上的能力还有待提高,这也是今后需要进一步研究的方向。

参 考 文 献

- [1] 宋杰. 基于 FLTK 的工程车显示屏 GUI 系统设计与实现[J]. 电力机车与城轨车辆, 2013, 36(4): 28-30.
- [2] 吴运智, 李广军, 郭志勇. 嵌入式图形工具集 FLTK 的研究与应用[J]. 实验科学与技术, 2007(4): 42-44, 50.
- [3] 杨帆, 曾志勇, 罗威. 不等宽字体的折行显示算法[J]. 硅谷, 2009(12): 12, 39.
- [4] 朱明东. 一种扩展的 Android 文本显示控件实现[J]. 软件导刊, 2015, 14(1): 33-35.
- [5] 倪晓军. 高效 Unicode/GB 编码转换算法的设计和实现[J]. 计算机技术与发展, 2009, 19(9): 21-24.
- [6] 席春梅. 初探 Unicode 编码[J]. 科技经济市场, 2008(9): 17-18.

4 结 语

本文提出了以总旅行预算费用为约束条件获取最大旅行体验的旅游景点规划问题,建立了景点综合评价指数模型。通过 0-1 背包算法求得费用约束条件下综合评价指数最高的景点集合,再利用旅行商算法遍历景点获取最短游览路径以降低交通费用,最后通过二分法循环优化上述过程得到景点规划最优解。

本文以南京主要景点为例给出其景点评价指数和景点距离矩阵,通过上述优化算法分别对预算总费用为 1 000 元和 500 元两种情况进行规划,其规划景点及游览路径合理,满足预算费用要求。

参 考 文 献

- [1] 袁光辉, 谢科, 邓林胜, 等. 旅游路线动态规划问题研究——以西安市出发为例[J]. 数学的实践与认识, 2016, 46(15):125-133.
- [2] 王艳, 印国成, 孙茂圣. 最佳游览路线生成方案的设计与实现[J]. 物联网技术, 2015, 5(12):87-89.
- [3] 杨丽萍. 最短路径算法在校园导游系统中的应用[J]. 计算机时代, 2014(2):31-32.
- [4] 邹时林, 阮见, 刘波, 等. 最短路径算法在旅游线路规划中的应用——以庐山为例[J]. 测绘科学, 2008, 33(5):190-192.
- [5] 徐婷婷, 王柱, 徐海洋. 旅游路线规划数学模型的建立与应用探讨[J]. 廊坊师范学院学报(自然科学版), 2016, 16(1):23-26.
- [6] 蓝雯飞, 吴子莹, 杨波. 背包问题的动态规划改进算法[J]. 中南民族大学学报(自然科学版), 2016, 35(4):101-105.
- [7] 王乐, 王世卿, 张静乐. 基于 Matlab 的 0-1 背包问题的动态规划方法求解[J]. 计算机技术与发展, 2006, 16(4):88-89.
- [8] 王永静. 基于动态规划法和模拟退火算法求解旅行商问题[J]. 商丘职业技术学院学报. 2016,15(5):5-7.
- [9] 王敏. TSP 问题及几种常见算法的比较研究[J]. 长春理工大学学报, 2010(5):184-185.
- [10] 张子寒, 张落成. 基于多种模型的旅游线路规划探讨——以南京主要景区游览为例[J]. 计算机应用, 2016, 36(S1):278-280.
- [11] 携程. 南京景点推荐[DB/OL]. <http://you.ctrip.com/sight/nanjing9.html>.
- 研究[D]. 北京:北京邮电大学, 2014.
- [3] 胡朝建. 一种物联网开放平台认证授权机制的设计与实现[D]. 广州:华南理工大学, 2014.
- [4] Hasan R, Winslett M, Conlan R, et al. Please permit me: stateless delegated authorization in mashups [C]//Annual Computer Security Applications Conference(ACSAC). 2008:173-182.
- [5] OpenID OAuth Extension Website[EB/OL]. http://step2.googlecode.com/svn/spec/openid_oauth_extension/latest/openid_oauth_extension.htm.
- [6] 鲁金钿, 尧利利, 何旭东, 等. 改进的 OpenIDConnect 协议及其安全性分析[J]. 计算机应用, 2017, 37(5):1347-1352.
- [7] 韩妍妍. 可视密码技术的研究[D]. 西安:西安电子科技大学, 2009.
- [8] 杨健, 汪海航, 王剑, 等. 云计算安全问题研究综述[J]. 小型微型计算机系统, 2012, 33(3):472-479.
- [9] Ranjbar N, Abdinejadi M. Authentication and authorization for mobile device [D]. Sweden: University of Gothenburg, 2012.
- [10] Jones M, Handt D. The OAuth 2.0 authorization framework: bearer token usage [EB/OL]. <https://tools.ietf.org/html/rfc6750>.
- [11] Zhang J L, Lu J T, Wan Z Y, et al. Security analysis of OpenID connect protocol with cryptoverif in the computational model[C]//Proceedings of the 11th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. Berlin:Springer, 2016:925-934.
- [12] 杨雪松, 王书文, 刘勇, 等. 一种基于视觉密码的云平台访问控制方案[J]. 甘肃科技, 2014, 30(3):11-13.
- [13] 曹晟, 陈峻, 崔喆, 等. 基于视觉密码的无线网络远程身份认证[J]. 计算机应用, 2008, 28(6):39-42.
- [14] 冯国柱, 李超, 吴翊. 基于视觉密码的身份认证方案[J]. 计算机应用, 2006, 26(10):2318-2319.

(上接第 328 页)

(上接第 312 页)

- [2] 马德新. 基于 Web 的物联网体系结构和感知域关键技术

- [7] 张鹏伟, 李建文. 数据库系统开发中字符编码问题的研究[J]. 陕西科技大学学报, 2013, 31(5):139-143.
- [8] 闫静, 王天宝, 罗浩. 应用开发中的中文乱码原因及其解决方案[J]. 成都信息工程学院学报, 2012, 27(5):458-461.
- [9] 谭园园. WEB 开发中的乱码及其解决方法[J]. 数字技术与应用, 2012(7):86-88.