维、哈、柯文版 Linux 操作系统关键技术的设计实现*)

苏建辉[1,2]　程　晶[1,2]　蒋同海[1]

(中国科学院新疆理化技术研究所　乌鲁木齐 830011)[1]　(中国科学院研究生院　北京 100039)[2]

摘　要　维吾尔文、哈萨克文、柯尔克孜文(以下简称民文)等多文种操作系统软件,对于提高我国西部少数民族地区的信息化水平,起着重要作用。本文首先阐述了民文与汉文、西文等语言在计算机处理方面的差异,然后通过分析 Linux 国际化框架,提出了民文版 Linux 的总体设计结构,最后分两个模块重点论述了民文变形显示、从右向左书写、民汉混合处理等主要技术难点在 Qt 层次上的实现。测试表明,民文版 Linux 在保证原有功能的基础上,全面支持了民文的使用习惯。

关键词　维、哈、柯文,国际化,Linux,变形显示,左向文本

Design and Implementation of Pivotal Technology in UKK Operating System Based on Linux

SU Jian-hui[1,2]　CHENG Jing[1,2]　JIANG Tong-hai[1]

(Xinjiang Technical Institute of Physics & Chemistry, CAS, Urumqi 830011, China)[1]

(Graduate School of Chinese Academy of Sciences, Beijing 100039, China)[2]

Abstract　Operating system software supporting minority languages such as Uigur, Kazak and Khalkhas (UKK), is playing an extremely important role in raising the informationization level of Xinjiang Uygur Autonomous Region. Firstly the processing differences among UKK, Chinese and English were discussed. Then based on analyzing the internationalization framework of Linux, architecture for the UKK Linux was proposed. Finally The main pivotal technology for minority languages processing was elaborated in detail in two subsystems. It's proved that the UKK Linux can not only reserve its original functions, but also accord with the special custom of UKK very well.

Keywords　Uigur, Kazak and Khalkhas (UKK), Iinternationalization, Linux, Transfiguration display, Right-to-left text

新疆维吾尔自治区是我国一个多民族聚居区。在该区内,汉文和维吾尔文、哈萨克文、柯尔克孜文(以下简称民文)等少数民族文字普遍地混合使用。目前国内支持民文的操作系统(OS)非常稀少,不能满足民文信息化应用的迫切需要。因此基于 Linux 开发可以同时处理民文、汉文和英文的 OS 软件有着非常重要的价值,对于发展我国西部少数民族地区经济,提升社会的信息化水平等都有着巨大的推动作用。

1　民文的书写规则和设计难点

维吾尔文、哈萨克文和柯尔克孜文这三种文字均属阿拉伯语系的文字,与汉文和西文差异很大。主要包括:(1)与上下文内容相关的字符显现形式。根据邻近字母属性的不同,民文字母会具有不同的显现形式。一般来说,可以分为 4 种显现形式:独写形、首写形、中写形和尾写形。(2)左向文本。汉字和西文的对齐方式是靠左对齐,书写方向是从左到右,它们称为右向输入文字。而民文的书写规则恰好与此相反,文字是按照靠右对齐,从右向左的规则来书写,属于左向文本。(3)民文之间也有部分差别,维、哈、柯文三种文字的字母数量和字母的组成也不相同,即使相同字母的变形规则也有一定的差别。

依据民文的以上特点,设计民文版 Linux 需要解决的技术难点有:(1)实现民文显示时自动选形,使系统在输入状态与修改状态下都可以根据民文字母在词中出现的位置,自动选定正确的变形显现字符。(2)解决民文行文方向相反的问题,在以汉英文为目标语言的图形界面中,实现民文靠右对齐,从右向左书写的编辑、排版和显示。(3)由于新疆地区的使用环境,民文版 Linux 必须可以同时处理汉文,因此系统要求支持民文与汉英文字的相互嵌套处理,并且两种文本都可以按照各自的文本书写方向显示。

2　Linux 国际化框架

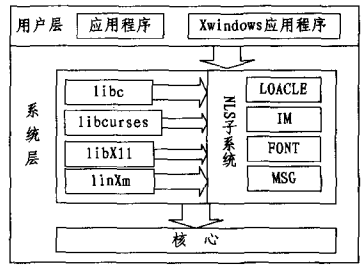

图 1　Linux 系统 NLS 体系结构图

Linux 国际化的核心是 NLS(National Language Support)子系统,它符合 POSIX(Portable Operating System Interface)标准,总体框架见图 1。该子系统建立在基于 ASCII 码的 Linux 核心上,系统中所有支持多国语言的实用程序,包括 X Windows 应用程序都是建立在这个基础上。在此基础

*)国家 863 计划项目(2003AA1Z2110),新疆维吾尔自治区高技术研究与发展计划资助项目(200412108)。**苏建辉**　硕士研究生,主要研究方向为多语言软件的开发与测试;**蒋同海**　硕士生导师,主要研究方向为多语言软件的开发与测试。

上可以建立支持各种不同的语言文化的民族特征数据库(locale)、输入方法(IM)、字体(FONT)和本地化文本信息(message)等。在 NLS 子系统中,locale 则是本地化工作的一个基石,因为不管是 glibc,还是系统的其它部分,都是通过读取系统当前的 locale 设定来识别当前的本地化区域,从而使用正确的本地数据。

3 民文版 Linux 设计结构

3.1 设计思路

民文版 Linux 建立在 NLS 基础上。首先通过选择适当的本地化环境变量,配置民文本地化环境,安装民文字体和民文界面翻译,在图形函数库中加入针对民文文本显示的支持,再在输入法服务器中集成一个民文输入模块,就可以建立一个比较完善的民、汉、英多文种操作系统平台,可以输入、显示输出民文、汉文,并具有全面的民文界面,符合民文习惯的使用环境。

3.2 设计结构

民文版 Linux 最大的不同就在于它的图形界面,因此其研发重点也主要集中在图形界面层。依据民文的使用特点及 Linux 国际化框架结构,把民文版 Linux 划分为 5 个子系统,即民文自动选形子系统、民文左向文本书写子系统、民文本地化环境子系统、界面资源子系统、多语种输入子系统等。各子系统关系如图 2 所示。

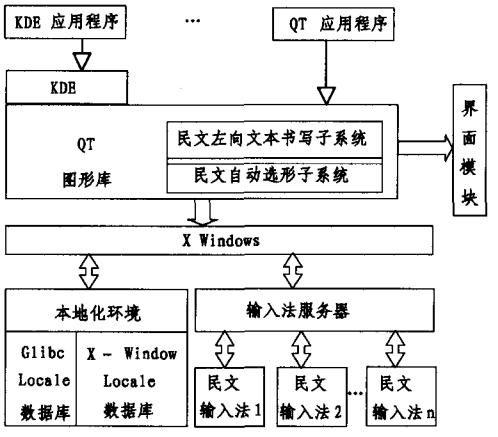

图 2 民文版 Linux 设计结构图

各部分的主要功能如下:

1)民文自动选形子系统:包括单字符显示字形的选择与变形组合字符的选形。主要功能是将系统中需要显示的文本划分为汉、英文片断与民文片断,并将民文片断中的民文名义字符转换为符合上下文的显示字符,从而使系统界面上显示的民文可以自动选择显示字形。

2)左向文本书写子系统:按照民文靠右对齐、从右向左书写的规则,将民文的逻辑顺序(即输入顺序)调整为显示顺序。同时支持民文与汉英文字的混合编辑排版。

3)多语种输入子系统:与具体的输入法及当前平台界面所使用的语言无关,支持在系统运行中自由切换民文、哈文、柯文、汉文和英文输入的功能,因此用户可以同时对汉文、英文、民文等多语种进行混合输入。

4)民文本地化环境数据库系统:包括 glibc 的 locale 和 X Window 的 locale 数据库,它们影响基本函数行为,提供了程序运行的本地化环境,为整个系统提供了与民文的文化特性有关的描述信息。

5)民文界面资源子系统:为系统提供民文界面信息,这一部分包括应用程序的民文界面的翻译和主菜单、桌面条目的民文翻译和民文字体安装与资源文体配置。

对于民文版 Linux,民文本地化环境数据库、界面资源的翻译合并、民文多语种输入法子系统等都与中文版本差别不大,在本文不做介绍,主要介绍自动选形、从右向左书写、民汉混合处理等主要技术难点的实现。

4 关键子系统的设计与实现

4.1 民文自动选形子系统

民文自动选形子系统有两种实现方案。第一种是放在输入法中实现的,通过在输入法中存储以前输入的字符,建立选形状态空间来选择民文显示字形,并输入到系统中,也就是说在此类系统中,输入与存储的民文都是显现字符的编码。第二种是采用以名义字符作为民文存储与输入字符的编码基准,在字符显示时才进行自动选形的设计方案。前者的好处是实现起来比较简单,但缺点是它不能解决修改状态下的自动选形,也就是说当光标移动到文本编辑区的前几行进行字符的插入、删除操作时,因为不能建立选形状态空间,则修改字符两侧及插入字符就不能选择正确的显现字符。后者克服了前者的弊端,因此我们在开发民文版本 Linux 系统时采用第二种方案来设计民文自动选形子系统,设计结构如图 3 所示。

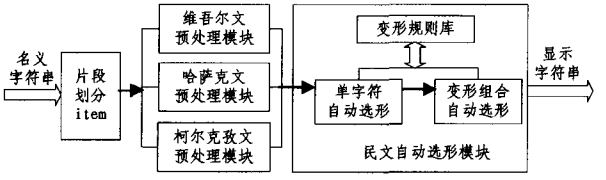

图 3 自动选形子系统结构图

4.1.1 片断划分

系统需要同时处理民文、汉文、英文等多种文字,为了使自动选形子系统只对民文字符进行处理,对于显示字符串,首先要进行文本片断化处理,依据字符的编码范围、方向属性、字体等特征将字符串划分成多个片断,文本中任何一个特征的改变都是一个新片断的开始,使每个条目中的字符具有同样的语言、字体、方向属性。接着系统会根据当前的本地化全局变量调用相应的民文预处理模块来处理民文片断的变形属性。

4.1.2 民文文本预处理模块

民文预处理模块是按照民文字母变形规则来定义民文字符的变形属性。民文文本之间的选形规则类似,但是它们之间存在部分编码相同但对应的显示字符不同的字母,对于这三种文字我们分别设计了预处理模块。

按照民文具有的显示字形的种类,可以划分为一种(只有独写形)、两种(独写形与尾写形)与四种(独写形、首写形、中写形、尾写形)三类,它们对应的连接类型分别是:(1) 独立型:不能与前后字符相连的字符;(2) 右连型:只能与前面字符相连的字符;(3) 双连型:可以与前后字符相连的字符。同时民文字符中还有一个 Unicode 编码为 0x0640 的特殊字符,它在民文编辑中经常用来拉长字符,所以它可以与前后字符相连,但是自身没有变形字符。我们将它的连接类型划分为全连型。

我们对字符的连接类型进行了归纳，将其划分两种连接超类型，分别是右连接超类型与左连接超类型。其中右连接超类型包括双连型与全连型。左连接超类型包括双连型、右连型与全连型。

民文预处理模块对输入的字符串进行分析，依据不同民族语言字母的变形类型，分析民文字符的变形类型与连接超类型，并绑定到该字符上。同时对字符的变形组合属性进行分析，对可以与相邻字符组合成变形组合的字符，将其组合属性位置位。

4.1.3 民文自动选形模块

经过上一步获得输入名义字符的连接属性后，就可以进行单字符自动选形计算，我们采用民文单字符选形算法进行单个名义字符到显现字符的转换。民文单字符选形算法的选形规则如下：

(1)字符的连接类型为右连型，右侧如果有一个右连接超类型字符，则显现字符使用尾写形。

(2)字符的连接类型为双连型，右侧如果是一个右连接超类型字符并且在左侧是一个左连接超类型字符，则显现字符使用中写形。

(3)字符的连接类型为双连型，右侧如果是一个右连接超类型字符，而在左侧没有左连接超类型字符，则显现字符使用尾写形。

(4)字符的连接类型为双连型，左侧如果是一个左连接超类型字符，而在右侧没有右连接超类型字符，则显现字符使用首写形。

(5)如果以上规则都不能应用在当前字符上，则显现字符使用独写形。

民文单字符选形算法依次处理名义字符串中的字符，首先取得字符的连接类型与前后位置字符的连接超类型，依据上述民文单字符选形算法规则由上到下的顺序与选形规则进行比较，当与其中一条规则匹配后，就把该字符通过映射表将名义字符编码映射为相应的显现字符的编码。例如，有一个由3个字符组成的民文词，第一个字符的连接超类型为右连接超类型，第二个字符的连接类型为双连型，第三个字符的连接超类型为左连接超类型，对第二个字符进行选形时，可以匹配到规则二，则显示字形就是中写形，然后通过映射表就可以将其名义字符编码转换为对应中写形字符的编码。

下面将讨论多字符组合字型的变形。首先讨论民文中强制类型合体字形的变形规则。

以X和Y代表变形组合字符序列，民文变形组合字型的组合规则为：

(1)X字符的尾写型在左、Y字符的首写型在右的序列将形成和体字形XY；

(2)X字符的尾写型在左、Y字符的中写型在右的序列将形成和体字形XY的尾写型。

上述变形规则本质上就是一个收敛的递归模型，其(1)规定了递归的终止条件，(2)规定了递归的规则。

单字符选形结束后，对字符的组合属性进行判断，对于组合属性位置(1)的民文字符则按照组合规则计算组合字型的编码。通过以上步骤，就可以实现民文名义字符到界面显现字符的转换。转换后的民文显现字形按照上下文就可以准确相连在一起了。

4.1.4 实现

在Linux图形库Qt中主要由QTextEngine类负责进行界面字符串的分析显示。在这个类中我们首先在Itemize()方法中将显示字符串分成片断(item)，每一个字符片断中的文字具有完全相同的属性。其次再经过QTextEngine成员函数shape()中插入民文自动选形模块的入口函数。在shape()中会调用scriptEngines数组(也被称为JumpTable，它的成员都是函数指针)中的函数对文本进行显示字符处理。我们就是在scriptEngines数组中插入民文自动选形入口函数：

```
const q_scriptEngine scriptEngines[] = {
    { basic_shape,basic_attrlbutes }    //Latin
    { minority_shape,minority_attributes }
    ...
```

在入口函数minority_shape中，依次调用了三个函数来实现民文自动选形的功能：

(1)charProcessing()：对应民文预处理模块，应用民文知识库分析字符串，得到民文字符变形类型与连接超类型并绑定到输入字符上。

(2)glyphConverter()：按照民文选形算法，利用绑定在字形上的属性，分析民文字符应该显示的字符，并按照民文字符映射表进行名义字符到显现字符的替换操作。

(3)combineGlyph()：进行变形组合字符的合成替换。

通过以上设计就可以实现系统中输入、存储的是民文的名义字符，到显示时再进行变形显示的变形方案。它可以彻底解决民文字符修改状态下的自动选形。

4.2 左向文本书写子系统

该子系统应实现的功能是：按照民文靠右对齐、从右向左书写的规则，将民文的逻辑顺序调整为显示顺序；当民、汉文字互有嵌套时，汉文需要从左向右阅读，民文从右向左阅读，在屏幕的显示的两种文字的字序都要符合各自的语言习惯。

在介绍实现过程之前，先要明确两个概念。一个是字符的逻辑顺序(logical order)，它与字符输入顺序一致，同时它也是文件中字符存储顺序。另一个是字符的可视顺序(visual order)。这就是字符屏幕显示顺序，如图4所示。

图4 双向文本的逻辑顺序和显示顺序比较

图4中大写字母代表另外一种从右向左书写的文本。

处理过程：分析逻辑顺序(也即输入顺序)的文本，将文本分为item，每个相邻的item的书写方向都不一样，而item内的字符的书写方向是一致的。再从外到内地确定文本的嵌套级。最外层的文本如果是从右向左书写的文本，级数为1，如果是从左向右书写的文本，级数为0，内层的item按照嵌套层次而以1为单位增加级数。在每一个item的级数都确定以后，再从内向外，从级数最高的item开始，子程序按级数对文本执行倒序操作。如果最高层的级数为n，则第一次将最高级数的item文本进行倒序，第二次将它相邻的级数为$n-1$的item文本和第一次已执行过倒序的item的文本一起进行倒序操作。依次类推，直到最外层，就可以得到民文与汉文混合文本的屏幕显示顺序。

在Qt图形函数库中，对民文和汉文的处理过程是由

QTextEngine 类完成的。首先通过 QTextEngine 类静态成员函数 static void bidiItemize (QTextEngine * engine, bool rightToLeft, int mode) 按文本方向将混合文本分成 item,并计算每个 item 的级数,每个 item 内的文本的方向都是一致的。然后通过 QTextEngine 类的成员函数 void QTextEngine ::bidiReorder(int numItems, const Q_UINT8 * levels, int * visualOrder) 进行 item 倒序操作,这样就可以得到文本的显示顺序。然后再对民文进行自动选型,计算每个 item 内字形大小和相对位置。最后计算每个 item 的绝对屏幕坐标位置,按 item 的显示顺序,将每个 item 依次显示在屏幕上。这样不论是对于段落向右对齐、从右到左书写的文本段,还是段落向左对齐、从左向右书写的文本段,处理的程序是一致的。不同的是每个 item 在屏幕上显示的坐标位置不同,对于段落向右对齐的文本,先按照向左对齐的模式计算它显示时的 X,Y 值,再对 X 值进行一个固定的坐标变换,使它右边界(item 的轮廓是一个矩形)靠文本书写区最右侧边界的距离与以前的 X 值一致,这样显示出的效果就是靠右对齐。在 Qt 中 QtextEngine 类成员函数 endLine 进行每个 item 的坐标全定位。

结束语 我们在 Redhat Linux 9.0 的基础上,按照上述方式对 Qt 库进行修改,加入民文自动选形模块和左向文本书写模块,重新编译 Qt 库,同时在对应目录下加入民文本地化数据库、输入法、字体以及界面翻译文件,修改系统默认的本地化 Locale 变量,就完成了维、哈、柯文本版 Linux 的开发,它在保证了系统原有功能的基础上,全面支持了民文的使用习惯。在使用 Qt 和 KDE 开发的图形程序中实现了民文自动选形、从右向左书写和民、汉文字混合编辑排版显示等功能。在图形方式下,可以按照民文的使用习惯实现民文文件名、文件夹、搜索维文文件等操作系统的文本处理功能,从而完成既定的设计目标。

文章论述的这些关键技术,不仅很好地实现了 Linux 下维、哈、柯文的信息化处理,而且对于与维、哈、柯文类似的阿拉伯语系文种的处理也具有普遍的指导意义。

参 考 文 献

[1] Arabic Code Chart. http://www.unicode.org/charts/PDF/U0600.pdf,2005,1

[2] Arabic Presentation Forms - A. http://www.unicode.org/charts/PDF/UFB50.pdf,2005,1

[3] Arabic Presentation Forms - B. http://www.unicode.org/charts/PDF/UFE70.pdf,2005,1

[4] Arabic Shaping. http://www.unicode.org//versions/Unicode4.0.0/ch08.pdf,2005,1

[5] LineBreakingProperties. http://www.unicode.org/reports/trl4.pdf,2005,1

[6] Bishop A, Brown D, Meltzer D. Supporting multilanguage text layout and complex scripts with Windows 2000. http://www.microsoft.com/typography/developers/uniscribe/intro.htm, 2003,12,12

[7] OpenType Specification Version 1.4. http://www.microsoft.com/typography/otspec/default.htm,2004

[8] 古丽拉·阿东别克,米吉提·阿布力米提. 维吾尔语词切分方法初探. 中文信息学报,2004,18(6):61-65

[9] 苏国平,缪成,夏国平. Linux 下维、哈、柯文多语种图形化处理平台的设计与实现. 中文信息学报,2004,28(4):88-93

[10] 芮建武,吴健,孙玉芳. 国际化文字处理综述. 中文信息学报,2006,20(2):87-93

[11] 靳简明,王华,丁晓青. 维汉英混排文档识别. 电子与信息学报,2006,28(7):1188-1191

[12] 缪成,袁保社,李莉. Linux 系统下开放式维、哈、柯、汉、英多语种混合输入法系统. 计算机应用,2003,23(11):36-38

[13] 马宁,于洪志. Linux 民文化技术. 西北民族大学学报:自然科学版,2005,26(1):58-63

[14] 卢有飞,张伟,等. 维文版 Office 设计中关键技术的研究与实现. 中文信息学报,2007,21(2):112-116

[15] International Dr. 国际化软件开发. 申凤,等译. 北京:机械工业出版社,2003

[16] 郑燕飞,刘岩,陈克非. Linux 文平台的实现关键技术及其发展方向研究[J]. 计算机工程,2002,28(1):14-15

[17] 董军. 基于 Java 平台的维文版永中集成 Office 的设计[D]. 北京:中国科学院研究生院,2005

(上接第 242 页)

取算法、相似性匹配算法甚至智能语义等都有密切的关系。我们下一步研究的一个主要目标就是实现模型的自动分类。

原型系统的检索方式中支持的关键字的个数有限,而文件检索方式则要求用户本地机存在查询模型才可以进行模型检索。这些检索方式在有些情况下并不能很好地满足用户的检索要求。因此,更加适宜用户使用的三维检索界面技术也是我们未来研究的内容之一。

参 考 文 献

[1] 杨育彬,林珲. 基于内容的三维模型检索综述. 计算机学报,2004,27(10):1297-1310

[2] Funkhouser T, Min P, Kazhdan M, et al. A search engine for 3D models. ACM Transactions on Graphics, 2003(22):83-105

[3] Chen D, Tian X, Shen Y, et al. On Visual Similarity Based 3D Model Retrieval. Computer Graphics Forum (EUROGRAPHICS'03), 2003, 22(3):223-232

[4] 郑伯川,彭维. 3D 模型检索技术综述. 计算机辅助设计与图形学学报,2004,16(7):873-881

[5] Zaharia T, Preteux F. Shape-based retrieval of 3D mesh models. IEEE International Conference on Multimedia and Expo (ICME 2002), 2002(8)

[6] Kazhdan M, Funkhouser T, Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3D shape descriptors // ACM SIGGRAPH'2003. 2003:156-164

[7] Vranic D. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres // IEEE International Conference on Image Processing (ICIP'2003). 2003(3):757-760

[8] Veltkam P R. Shape matching: Similarity measures and algorithms. Shape Modeling International, 2001(5):188-197