

61-64

WINDOWS 汉化与中文应用软件开发

沈浩 张然

(复旦大学计算机科学系 上海 200433)

TP316

TP311.52

A 摘要 近几年,Windows 中文平台纷纷涌现,Windows 的中文应用日益增多。本文讨论了 Windows 中文平台的两种形式:核心汉化平台和外挂汉化平台,比较了不同中文平台的特点,介绍了几种中文应用软件开发方法并针对 Windows 的 DBCS 处理机制指出了中文应用软件开发时需要注意的问题。

关键词 汉化, 汉化平台, Windows, 中文应用软件, 开发方法。

应用程序 Windows 3

软件开发

1. 汉化回顾

1990年5月,Microsoft 公司推出了 Windows 的第一个成熟版本 Windows 3.0。彻底改变了 PC 机的操作界面,在短短两三年里很快成为图形用户界面事实上的工业标准。当人们纷纷转向 Windows 时,汉字处理技术也就主要成了 Windows 汉化与中文应用软件开发问题了。我们看到,Windows 的特性降低了开发中西文软件之间的差异,汉化后的 Windows 是一个比较理想的中文平台。

Windows 主要面向英文处理,早在 Windows 3.0 版本之前,国内的一些有识之士就注意到了 Windows 的发展潜力,开展了汉化工作^[6]。90 年底到 91 年初,国内先后出现了长城、北大、江帆等多个版本,台湾方面也出现了资策会、图容等汉化版本。1991 年 7 月台湾 Microsoft 公司还推出了他们的 Windows 3.0 中文版(CWIN3.0),北大则推出了 BD-WIN3.0。1992 年 3 月,Microsoft 公司正式推出了 Windows 3.1 版,Windows 的汉化技术逐渐成熟,形成了外挂式和核心法两种主要的汉化方法并立的局面。当前,性能比较完善且用户较多的主要有 Microsoft 公司于 93 年 10 月推出的 PWIN3.1、新天地推出的 CSTAR(1.3 版和 2.0 版)、北京四通利方公司的 RichWin 和美国双桥咨询公司的双桥中文视窗。本文将以前两种产品为例进行分析和比较。

纵观 Windows 汉化的历程,我们可以把 Windows 中文环境的发展划分为以下三个阶段。第一阶段为汉化阶段。人们看重 Windows 的图形功能、多窗口管理功能、多任务处理能力和友好的界面,对 Windows 汉化的要求仅仅是使之具备基本的中文

处理能力,而更高级的中文处理则留给中文应用软件来完成。第二阶段为中文系统阶段。以中西文兼容为突破口,目标是使 Windows 中文环境和中文功能与西文相当,使之成为一个实用的开放的中文系统,支持中西文应用软件开发。第三阶段为中文平台阶段。中文平台通常具备以下三项功能:汉字功能;可以增加字体资源的开放系统;支持应用开发。

中文平台已经成为 Windows 中文环境的主流。

2. Windows 与中文化

所谓 Windows 中文化,其工作的实质就是使西文系统能处理双字节内码,实现中文输入和中文输出。Windows 自身的一些特点给汉化工作提供了便利,在很大程度上减轻了 Windows 汉化的工作量。

第一、大量的应用功能转换到系统层。Windows 是建立在 DOS 基础上的操作系统,它不仅增强了作为一个操作系统所必需的内存管理,进程管理等基本功能,吸收了多媒体,网络、笔式计算等新技术,还扩充了许多原本由应用软件提供的字体、文字处理、对话和编辑等功能。大量的功能由应用级转移到系统级,保证了风格的统一,简化了应用软件开发。功能的实现方式与应用软件无关,这也是软件兼容性的重要保证。

第二、系统不与外设直接发生联系。Windows 的设备无关性技术,使各种外设均以 DRIVER 形式接入系统,对系统而言,各种设备是由 DRIVER 描述的“虚拟设备”。Windows 提供了一个与具体设备无关的图形设备接口 GDI,应用软件通过调用 GDI 函数实现输入输出。这样的三级结构实现了设备无关性,也为 Windows 支持多文种(包括中文)奠定了

基础。

第三、保留了对双字节内码的支持。利用 Windows 中支持双字节内码的 API(应用软件编程接口)函数,我们可以开发支持 DBCS 内码的软件,用运行在 Windows 的中文平台上的软件处理汉字。

第四、便于语言间转换的新风格。Windows 提倡把提示信息(如菜单、对话、字符串等)放到软件资源表中的设计风格,有利于软件在不同语言版本间进行转换。

3. Windows 的汉化方法及比较

有两种不同的构造 Windows 中文平台的途径:核心汉化法和外挂汉化法^[6,7]。为理解这两种方法的不同之处,我们首先来看一下 Windows 的核心程序结构^[7]。

3.1 Windows 的核心程序结构

Windows 是一个国际化的操作系统。为兼顾各种语言,Windows 的核心程序均由与语言无关的固定基本处理模块和可更换的特定语言模块(动态链接库形式)组成。例如图形设备接口(GDI)由固定的基本 GDI 模块和可更换的特定语言处理模块组成(图 1),基本处理模块是一个单字节处理模块,对于各种文字,只更换不同的链接库即可。

Windows 的核心程序的上述特点,使外挂中文平台得以实现。外挂中文平台不必修改核心程序,汉化工作仅限于增加一个中文处理模块,以便提供中文输入功能和输出功能,同时保留西文的语言处理能力。当然也可以直接修改输入输出模块,使之能够同时处理中文和英文。

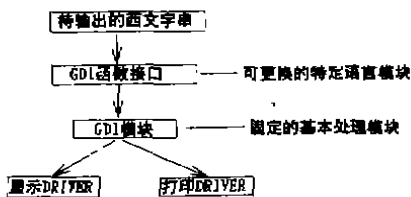


图 1 Windows 的核心程序结构

3.2 外挂汉化法

外挂汉化法是将一个独立的中文模块挂在 Windows 系统上,不更改任何西文处理模块。这种方法,以 CSTAR 为代表。

以中文输出为例(图 2)。外挂汉化法把一个独立的中文处理模块,外挂到 GDI 模块上。

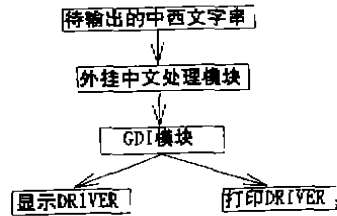


图 2 外挂汉化法示意图

采用外挂汉化法具有以下优点:

(1)具有中文处理的独立性,最大好处是独立于英文 Windows 系统,不仅能支持英文、中文、日文、德文等版本的 Windows,还能支持 Windows For Workgroup(WFW)网络系统。

(2)广泛的适应性。外挂汉化法不更改系统的核心和外设驱动程序,保持英文 Windows 的单字节属性。国外众多优秀软件是以单字节为基础的,其中的一些软件在中文 Windows 环境下运行会存在一些问题,而采用外挂汉化法则可以有效地克服这些问题。

(3)灵活性。外挂汉化法的外挂程序可在任何时候外挂到 Windows 系统上,也可在任何时候从 Windows 系统上摘除。摘除后,Windows 或 WFW 又完全恢复到原来的西文状态。在升级时只需要更换外挂汉化模块即可。

但是采用外挂汉化法也有缺点:

(1)因为核心系统是单字节的,光标位置和删除均以单字节进行,所以如果应用软件直接调用了核心的有关功能,在换行时可能会产生乱码。

(2)显示速度较慢,且无法通过修改 Windows 核心来解决。一般采用在内存中的缓冲区存放已显示过的汉字的方法以在一定程度上提高显示速度。

3.3 核心汉化法

核心汉化法就是修改 Windows 的核心和外设驱动程序,核心不保留原来的单字节部分^[6]。这种方法,以 PWIN 为代表。

我们也以中文输出为例(图 3)。

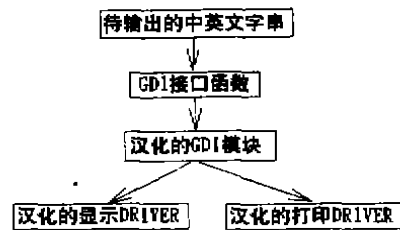


图 3 核心汉化技术

核心汉化法的优点如下:

(1) Windows 核心汉化,使其从系统内核到操作外壳均为中文环境,菜单提示、应用软件、联机帮助全部为中文方式。

(2) 通过修改核心程序,使汉字输入、中文写屏速度大大提高,基本上可达到与英文 Windows 相同的速度。

(3) 由于中文 Windows 是核心汉化的,判断光标位置、删除键的移动都以汉字为单位进行。在外挂汉化环境下出现乱码的程序,在核心汉化环境下一般不会出现乱码。

核心汉化法的缺点是:

(1) 兼容性较差。一些西文 Windows 下的 Driver 不能在汉化的核心下使用。

(2) 灵活性较差。用户要么选择中文 Windows,要么选择 Windows 的其它版本或 Windows For Workgroup;而采用外挂法则可以在不同文种之间相互切换。此外,在升级时用户需更换整个系统。

3.4 两种汉化方法代表的比较

PWIN 和 CSTAR 分别是当前核心汉化法和外挂汉化法的代表。就中文处理能力来说,使用完全实现 DBCS 处理机制的软件,均不会出现乱码,处理中文的能力相当。但就目前的情况来看,完全实现 DBCS 处理机制的应用软件并不多,有时候必须依靠核心功能。这样的应用软件在 CSTAR 下仍会出现乱码。从这一点来看,PWIN 可以支持最为广泛的中文应用软件和提供更好的中文处理功能。

表 3-1

	PWIN3.1	CSTAR2.0
启动速度	较慢	较快
显示速度	快	较快
是否存在乱码	是	是
输入法	3种	9种
字体	2种 (可扩展至6种)	6种 (可扩展至22种)
与西文软件兼容性	好	很好
中文应用软件支持	多	多
扩展性	好	较好
中文提示	好	一般

如果用户是地地道道的中文用户,主要业务是进行字处理,则 Windows 中文版是较好的选择;如果用户要兼顾中、西文环境,而且不作大量的字处理工作,则 CSTAR 还是比较方便的。尤其是 CSTAR2.0 对 CSTAR1.3 作了较大的改进,很多不能在 CSTAR1.3 上运行的软件已能够在 CSTAR2.0 上运行。

4. 中文应用软件的开发

Windows 中文应用软件大体分为两类。一类是对西文软件进行汉化。许多西文软件直接支持或部

分支持双字节内码,在 PWIN 或 CSTAR 下即具备了初步的中文处理能力。如果再对其提示信息进行中文处理,就可以满足人们处理中文的要求。另一类是专门开发的中文应用软件。这种应用软件可以在特定的中文平台上开发,比如说,PWIN 或 CSTAR。利用 PWIN 或 CSTAR 提供的适合中文处理的 API 编程接口,程序开发比较容易,但在不同平台上可能不兼容,也可以只利用 WIN3.1 标准的 API 进行开发,那样就可以在所有的 Windows 平台上工作。表 4-1 是对这两类中文应用软件的比较。

表 4-1

	汉化西文软件	开发中文软件
中文化效果	一般	好
是否存在乱码	是	一般没有
兼容性	好	较好
工作量	小	大

4.1 中文应用软件的开发策略

我们开发 Windows 下的中文应用软件时有以下几种方案可供选择:

(1) 直接使用或组合使用现有软件。当前有数以千计的英文软件,其内核支持双字节内码,可以进行中文处理而无需汉化。而且,可以利用 Windows 的多任务、多窗口的特性以及数据文档的功能对这些软件进行组合使用。

(2) 利用现有的软件提供的编程功能^[4]。很多 Windows 应用软件具备宏指令的功能,如 EXCEL、AMIPRO 等,使用这些宏指令也能进行编程。

(3) 专业软件开发人员可以使用 Windows3.1 的 SDK 和 DDK,或 PWIN3.1 的 SDKE 和 DDKE 开发工具直接编程。

前两种方法比较简单,但效果不是很好;最后一种方法比较专业,可以开发出各种实用的软件。

PWIN 的附件 NOTEPAD、PWIN 的附件 WRITE 和中文 WORD5.0 等三个软件在测试环境 CSTAR1.3+Windows3.1 和 PWIN3.1 上运行的结果如表 4-2 所示。

表 4-2

环境	CSTAR1.3+Windows3.1			PWIN3.1
所运行软件	NOTEPAD (PWIN3.1)	WRITE (PWIN3.1)	中文 WORD5.0	三种软件之一
启动	好	好	不能	好
打开文件	好	较好	—	好
删除	无乱码	存在乱码	—	无乱码
综合	好	一般	—	好

PWIN3.1 的两个字处理附件在同一个中文平台 CSTAR1.3+Windows3.1 上为什么会有如此不同的结果呢? 它们虽然都是字处理软件,都用 API 编程,但不同的开发方法产生了不同的运行结果。

NOTEPAD 短小精悍,功能单一,只能用来编辑较小、较简单的文档,没有排版功能。它利用了 Windows3.1 的 Edit 类来实现。CSTAR 接管了 Edit 类的消息,因此,在 CSTAR1.3 下能够较好地处理中文,不需要在汉字间加上空格以避免出现乱码。

WRITE 较 NOTEPAD 复杂,它有自己的文件格式,可以进行简单的排版,并没有在应用软件本身完全实现 DBCS 的处理,而是直接利用了 Windows 核心的判断光标位置、删除等功能。如果核心只支持单字节编码,必然会出现删除半边汉字和出现乱码的现象。这类中文应用软件依靠核心支持 DBCS 处理。

中文 WORD5.0 是由 Microsoft 公司推出的中文 OFFICE 中的字处理软件,它继承了西文 WORD 的强大功能,其排版的功能更强,能够达到美观的效果,如果这些功能完全由 Windows 3.1 的 API 函数来实现难度很大。WORD5.0 选择在 PWIN3.1 下开发,不仅因为 PWIN 是 Microsoft 公司的产品,还因为 PWIN3.1 中包含一个动态链接库 WINNLS.DLL,即 Windows 的国际化语言支持动态链接库。DLL 中包含了一些支持和实现 DBCS、MBCS 的 API 函数。WORD5.0 利用了这些 API,而 CSTAR1.3 中则不存在 WINNLS.DLL 文件,因此无法运行。

由以上三个软件,我们能够得到启发。一般来说,用 Windows3.1 的 API 函数开发的中文应用软件可以根据字处理功能的要求大体分为以下三类:软件只要求简单的中文输入;软件不仅要求输入中文,还要有一定的编辑功能,但字体单一;专门的字处理软件,有字体变化,包含复杂的编辑排版功能。

我们可以根据软件的功能、工作量、商业化程度来选择开发平台和开发方法。表 4-3 比较了用不同方法开发上述三类软件的工作量。

表 4-3

	标准 API 实现 DBCS	利用核心功能	用中文编程接口
第一类	1	0.7	0.7
第二类	3	2.5	2.2
第三类	10	9	8

4.2 Windows 的 DBCS 处理机制

中文是大字符集文字,用双字节字符编码,因此开发中文应用软件的关键在于处理好双字节内码的字符串。为此必须了解 Windows 的 DBCS 处理机

制。

在一个 DBCS、MBCS 等扩展内码体系中,使用最多的是不等长内码。也就是说,一个字符可以由一个、两个、或多个字节组成。而一个字符串中每个字节的意义,又与字符串中的前后字节相关。这就是不等长内码体系的基本特点^[6]。

不等长内码体系的最大优点是系统的兼容性和编码的高效性,所以得到了广泛的应用。CCDOS 等系统便是不等长内码的一个实用例子。但是,不等长内码体系也有很大的缺点,这就是字符串中单一字节的二意性,系统处理较为复杂。

在 Windows 系统的设计中,考虑了这种不等长内码体系的未来扩充,在 API 中保留了有关函数,并在整个体系中充分照顾了这一点。这就是前面所说的 Windows 的 DBCS 支持能力。

4.3 中文 Windows 下应用软件开发应注意的几点

由于中文 Windows 采用了核心汉化法,因此开发应用软件时可以采用核心部分判断光标位置、删除等功能。还可以采用附加的 API 函数,以加快开发进度。但要注意兼容性问题。

Microsoft 开发的 PWIN 为了实现处理双字节的内核,编制了一些特殊的 API 函数,也就是说 PWIN 的 API 函数是 Windows3.1 API 函数的一个超集。这些特殊的 API 函数一方面方便了软件开发人员开发 Windows 的中文应用软件,但同时也引入了不兼容的问题。到目前为止,由 Microsoft、Lotus 等公司委托开发的中文应用软件已经有许多,比如说,中文 OFFICE(包括中文 WORD、中文 EXCEL、中文 POWERPOINT),中文 AMIPRO 等等。这些程序在 PWIN 下开发,在 PWIN 环境下测试,因此这些软件在 PWIN 下一般能正常运行;而在 CSTAR 下运行时则会发生许多意想不到的问题。比如说:中文 AMIPRO 在 CSTAR1.3+Windows3.1 起动的情况下无法打开 AMIPRO 格式的文件等,因此兼容性是一个必须考虑的问题。

4.4 CSTAR 下应用软件开发问题

CSTAR1.3 版针对中文编程的基本任务和难点,设计和开发了一套中文编程接口(CSAPI)^[7],提供了中文编程服务。利用 CSAPI,软件开发人员可以加快开发进度。CSTAR1.3 版 CSAPI 以动态链接库的形式提供,其中定义了中文逻辑字型的基本结构,提供与中文处理有关的基本输入输出函数。

由于 CSTAR 采用外挂汉化法,因此在开发中文应用程序时应尽可能避免利用核心部分的单字节功能。