DOI:10.13657/j.cnki.gxkxyxb.1994.02.013

第10卷 第2期 广西科学院学报 Vol.10,No.2
1994年11月 Journal of the Guangxi Academy of Sciences Nov,1994

软件国际化机制的设计技术

石文昌

(广西计算中心 南宁市星湖路 530022)

摘要 通过分析UNIX系统国际化机制的实现方法,讨论软件国际化机制的设计技术。以便采取通用的方法来处理不同国家不同语言不同格式的信息。

关键词 国际化 本地化 场所 消息数据库 UNIX

1 引言

随着软件技术的发展,为使软件更好地进入国际市场,人们提出了软件的国际化思想。软件国际化的关键所在是解决不同国家与地区不同语言的信息处理问题。通俗地说,软件的国际化机制可以接收和处理世界上所有的语言文字信息,使软件产品不受任何语种的限制,用户只要通过简单的设置或修改,就可方便地选用自己所需的语言文字处理功能和人机界面表示形式。

软件的国际化机制要解决两个主要问题。其一,不同的语言可能需要用不同的字符集表示,软件的国际化机制要具有接收、处理和输出相应字符集的功能;其二,不同国家或地区,因语言和文化习惯的不同,对信息的表示可能采用不同的方式;比如,作为典型的例子,时间和日期的表示方法因不同国家语言文化习惯的不同就有一定的区别,软件系统与用户会话时,国际化机制必须确保系统人机界面的表示方法与用户的传统文化习惯一致。

UNIX是在世界范围内广泛流行的一个开放式标准操作系统,新版的UNIX系统已实现了国际化的处理能力。本文通过分析UNIX系统国际化环境的实现方法来探讨软件国际化机制的设计技术。

2 语言环境的描述

软件的国际化环境,是软件设计者们要考虑的问题。对于用户而言,他们最关心的问题恐怕是系统应按照他们所用的语言的要求和习惯来接收和处理信息。系统尽管提供多种语言的处理能力,用户往往只需要一种语言的处理能力。比如,中国用户希望系统提供的是中文处理能力,而很少关心系统是否支持对其他语种的处理。因而,在软件的国际化研究中,涉及到国际化和本地化两个相辅相成的概念。软件设计者提供国际化环境,而用户使用软件时选用本地化环境。

世界上很多标准化机构对软件国际化应用环境的建立进行了广泛的研究,其中,较有影

1993-11-01 收稿。

响的 ANSI X3J11 C 和 POSIX P1003. 1 这两个机构都为国际化的程序定义了标准。UNIX SVR 4.0 就是采用这两个机构所制定的标准设计国际化机制的。在国际化应用环境标准的制订方面，ANSI C 委员会定义一个称为“场所”（locale）的核心概念来描述语言环境。

一个场所是一个具体的人机交互会话环境，其内容包括语言、消息格式和消息表示方法等。会话语言在很大程度上代表着一个场所的主要特征，因而有时用会话语言来称呼场所，如“当前场所是汉语”，这意味着当前的语言环境是汉语环境，用户界面信息按照汉语的习惯给出。

给出了场所的定义后，语言环境的确定便等价于场所的确定。ANSI X3J11 C 标准草案定义一个数名为 setlocale 的函数，用以指定当前场所，执行这个函数以后，任何操作如果与场所有关，就遵从这个指定场所的有关规定。这个函数的调用接口是：

```
#include <locale. h>
setlocale (int category, const char * locale);
```

第一个参数 category 指定受场所影响的操作范围，也就是指定到底哪些操作将遵从该函数指定的场所的规定。这个参数可以取下列值：

(1) LC __ ALL：所有与场所有关的部分；

(2) LC __ COLLATE：核对序列；

(3) LC __ CTYPE：字符类型；

(4) LC __ MONETARY：货币单位；

(5) LC __ NUMERIC：数目表示；

(6) LC __ TIME：时间表示法。

第二个参数 locale 是场所名称，实际上就是用户所使用的语种的名称。当函数 setlocale 返回后，由 category 指定的所有操作将按 locale 指定的场所的有关规定进行。例如，用如下方式调用该函数：

```
setlocale (LC __ ALL," French")
```

它指明场所名称为“法语”，第一个参数表明以后所有与场所有关的操作都按法国习惯进行。如果把第一个参数改为" LC __ TIME"，则表明只有时间表示法采用法国的习惯表示，其他操作不受影响。

3 语言的处理方法

语言的处理包含两方面的内容，以汉语为例，一个方面是怎样对待汉字字符，另一个方面是怎样在程序中用汉语显示提示信息。一个具有国际化机制的系统需要处理多种语言的字符信息和提示输出信息。

3.1 字符信息的处理

众所周知，ASCII 码字符集的每个字符占用一个字节，有效位是其中的低 7 位。使用 ASCII 码字符集的用户习惯上总是把字符和字节当作同义词来使用。在国际化环境中，这种观点不再适用。例如，不能把“文件系统路径名的每个成分最多能占 14 个字节”说成是：“文件系统路径名的每个成分最多包含 14 个字符”，因为能占用 14 个字节的字符个数是随场所而定的。不同场所字符集的一个字符占用的数据位数不同。如：

ASCII 码字符集：每个字符占用一个字节（用其低 7 位）；

欧洲代码集：每个字符占用一个字节（8 位全用）；

汉字字符集：每个字符占用两、三个或更多个字节。

为在国际化环境中对不同的字符集作统一的处理，ANSI C 语言标准草案定义两种字符表示法：

(1) 宽字符

宽字符是一个内部数据类型，任何字符集的字符都可由它表示出来，程序中的字符操作用宽字符表示。ANSI C 标准用 typedef 定义的宽字符类型是 wchar __ t。

(2) 多字节字符

多字节字符是字符的外部表示法，用来实现多语种字符集的字符或字符串的输入/输出。一个 wchar __ t 类型的对象的多字节表示可以占多个字节， 其中包括移位状态编码。在 C 语言程序中用字符数组表示多字节字符。

3.2 显示信息的处理

在国际化环境中统一处理多种语言的一种方法是将不同语言的消息组织成各种消息数据库，系统与用户会话时，从相应的消息数据库文件中取所需的信息。UNIX 系统消息数据库在目录结构中的组织形式如图 1 的例子所示。目录/usr/lib/locale 包含法语和英语两个场所目录，每个场所目录包含一个称为 LC __ MESSAGES 的消息子目录，这个子目录下存放消息数据库文件。场所目录“法语”的 LC __ MESSAGES 消息子目录下有 app1 和 app2 两个消息数据库文件；场所目录“英语”的 LC __ MESSAGES 消息子目录下有 app1、app2 和 app3 三个消息数据库文件。

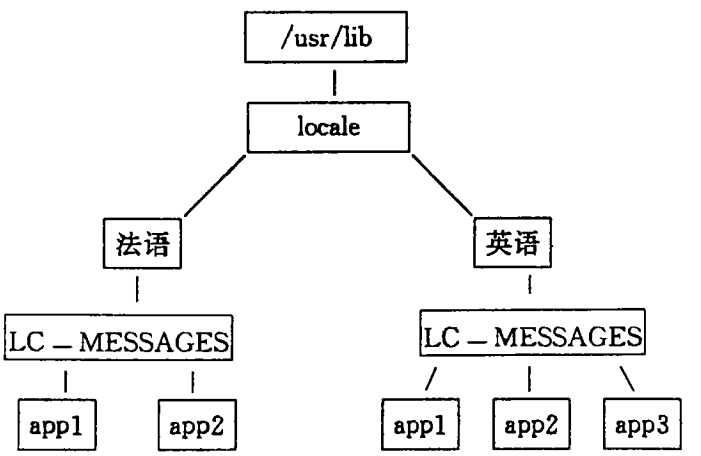

图 1　UNIX 系统消息数据库在目录结构中的组织形式示例

引进消息数据库以前，程序中需输出信息时，使用打印语句印出字符串常量，引进消息数据库后，程序中不再使用字符串常量，而改为对消息数据库文件中的消息的访问，这样使得程序能独立于具体语言。

UNIX SVR 4.0 提供两个消息管理工具 mkmsgs 和 srchtxt 对消息数据库进行操作，mkmsgs 用于建立新的消息数据库，srchtxt 用于查询消息数据库。

4 消息数据库技术的实施

4.1 消息数据库的应用

由非国际化环境转变到国际化环境，需要对系统中的程序作必要的修改。通过考察程序的修改方法，可以获得对消息数据库的应用的感性认识。在程序的修改过程中，系统中所有与场所有关的程序需要修改，程序中与场所有关的部分需要修改，其他与场所无关的程序或程序中的组成部分不必改动。在ANSI C的国际化应用标准中，C语言的关键字和符号等原封不动地保留，C语言的各种控制语句，如循环语句等无需改动。

在考虑实现国际化方案以前，如果要在程序中根据不同的语言习惯处理信息，我们也许可以采用如程序1中所示程序设计方法。

```
...
char *name;
/* 向用户索取文件名 */
if (西班牙语)
    name=span__ask__file ();
else if (法语)
        name=fren__ask__file ();
else if (德语)
        name=ger__ask__file ();
else if (汉语)
        name=chi__ask__file ();
else /* 缺省认为是英语 */
        name=eng__ask__file ();
...
```

程序1 不同语言习惯处理信息的程序设计方法示例

很显然，这种方法的最大缺点是使程序显得十分冗长，而且，程序中所能列举的情况是有限的，如果要支持一种新的语言环境，就得对程序作修改，而且改动起来并不容易。例如，为增加朝鲜语处理功能，需要编写一个新的子程序kor__ask__file，它用朝鲜语提示用户输入文件名并接受用朝鲜语表示的文件名。另外，程序中所有与场所有关的其他代码也都得作适当的改动，以符合朝鲜语习惯。

采用国际化实现方案以后，处理同样的问题，程序的面貌得到了彻底的改变。我们考察UNIX系统的作法。UNIX SVR 4.0借助于消息数据库提供一种新的简洁有效的方法。SVR 4.0提供两个消息数据库信息查找服务程序，一个是C语言函数，另一个是可执行的命令，它们的名称都是gettxt。函数gettxt调用接口是：

```
extern char *Msgdb;
char *gettxt (char *msgid, char *defaultstr);
```

该函数的返回值是指向消息正文首地址的指针，当找到消息正文时，指针指向找到的内容，找不到时，指向第二个参数给出的消息。

第一个参数是消息标识符，其格式是“消息文件名：消息标识号”，如“APP：57”，这表

示要在消息数据库文件 APP 中找标识号等于 57 的消息。当前场所确定消息数据库的入口，也就是说，如果当前场所是“汉语”则 gettxt 在汉语消息数据库中检索所需消息正文。

为合理解决独立于场所的消息处理问题，将程序中对 ASCII 字符串常量的引用替换成对通用正文查找服务程序的调用。用 gettxt 代替字符串常量后，使程序支持新的场所就是比较简单的事了，只要将程序的消息数据库翻译成新的语言，建立新的消息数据库便是。

程序 1 中的程序段改用 gettxt 来实现就简洁多了（见程序 2）

```
...
#define M__GETFNAME " APP: 57"
main ()
{char *p, *name;
/* 向用户索取文件名 */
p=gettxt (M__GETFNAME, " Enter file name:");
name=ask__file (P);
...
```

程序 2　改用 gettxt 实现不同语言习惯处理信息的程序示例

程序中的 gettxt 从当前场所的 APP 文件中找标识号为 57 的消息，若找到则指针指向该消息，若找不到则返回指针指向英文语句:" Enter file name:"。

4.2　旧程序向新环境的过渡

为了便于对原有的用户界面程序作修改，UNIX SVR 4.0 提供 exstr 消息管理工具，它的功能是自动查找程序中带引号和字符串（摘录功能），并把对这些字符串的引用改为对 gettxt 的调用（替换功能）。exstr 的任选项" -e" 表示执行摘录功能，任选项" -r" 表示执行替换功能。

下面我们通过一个例子来说明怎样用 exstr 把原来支持一种语言的程序改为支持多种语言的程序。

```
1 # include <fcntl. h>
2
3 main (int argc, char * *argv)
4 {
5    int fd;
6
7    if ( (fd=open (argv [1], O__RDONLY)) <0) {
8        fprintf (stderr,"%s: can't open file %s/n", argv [0], argv [1]);
9        exit (1);
10   }
/* 程序的其余部分 */
  }
```

程序 3　支持英语环境的程序示例

设原有程序 progl. c 的内容如程序 3 所示。程序中的第 8 行含有字符串常量。

执行摘录操作时，exstr 找出源文件（作为 exstr 的参数）中的所有字符串常量，并送到标

准输出，程序中的每个字符串常量在标准输出中有一个输出行与其对应，对应的输出行中除字符串常量外，还有标识信息，如源文件名、行号和字符串的第一个字符在行中的位置等，各种信息由冒号隔开。例如，执行下列命令：

exstr -e progl. c

将得到如下输出：

progl. c: 8: 18:::%s: can't open file %s\n

因为源文件中只有一个字符串常量，因而 exstr 的输出只有一行，输出行中有两个空的域，分别被用于定义消息文件名和消息标识号。为利用查找结果实现对程序的修改，需要对 exstr 的输出结果定向到指定的文件中，例如，执行如下命令：

exstr -e progl. c>progl. msgs

该命令把 exstr 命令的输出按原来的格式定向到文件 progl. msgs 中，在 progl. msgs 的各行的两个空域上填上相应的消息文件名和消息标识号（以便 gettxt 使用），在我们的例子中，progl. msgs 中只有一行，在第一个空域中填入" APP"，在第二个空域中填入" 1"，表示将该字符串常量映象成 APP 消息文件中的第 1 号消息。在此基础上再执行如下命令：

exstr -r progl. c<progl. msgs

便把程序 3 中仅支持英语的程序改成程序 4 中支持多种语言的程序。

```
extern char *gettxt ();
1 #include <fcntl. h>
2
3 main (int argc, char **argv)
4 {
5       int fd;
6
7       if ( (fd=open (argv [1], O__RDONLY)) <0) {
8        fprintf (stderr, gettxt (" APP: 1","") , argv [0], argr [1]);
9          exit (1);
10      }
       /* 程序的其余部分 */
  }
```

程序 4 支持多种语言环境的程序示例

5 结语

至此，通过以 UNIX SVR 4.0 这个现实系统为参照对象，我们分析了软件国际化机制的设计思想与实现方法。一言以蔽之，软件国际化应用环境的设计，一种切实可行的策略是，采用 ANSI C 所定义的场所概念描述语言环境，以宽字符和多字节字符来提供多种语言的字符集的处理能力，通过消息数据库和一整套有效的软件工具对不同语言的信息进行统一的处理。软件国际化机制的这种设计思想在系统软件和应用支撑软件的开发中具有重要的现实意义，同时，在应用软件的研制中，对于开发通用的应用系统也有一定的借鉴作用。

（下转第64 页）

它的最低频率约 180Hz；X 的最小数只能取 1，用相同的方法可算出它的最高频率约为 46KHz。这样，在 I/O 接口的第 15 脚和第 8 脚之间接 100KΩ 电位器，从电位器的中心抽头就获得输出脉冲重复频率范围为 180Hz－46KHz，幅度范围为 0～＋3.5V 的脉冲信号。这样宽的频率范围和幅度已基本满足实验的一般要求。如果需要更大的输出幅度，可另加驱动级。

Getting Adjustable Pulse Signals with Apple- Ⅱ Computer

Liu Detian
(Guangxi University)

Abstract The method of getting adjustalbe pules signals with Apple- Ⅱ computer was studied by using program. It is shown that Apple- Ⅱ can be used as an adjustalbe pulse source.

Key words Apple- Ⅱ computer, program, pulse source

(上接第60 页)

Design Technology of Software Internationalization Mechanism

Shi Wenchang
(Computing Centre of Guangxi, Xinghu Road, Nanning, 530022)

Abstract So as to use a general scheme to process various formats' information in different languages from different countries, The design technology of software internationalization mechanism is discussed by analysing the implementation method of the internationalization facilities of UNIX system.

Key words internationalization, localization, locale, message database, UNIX