从汉字编码到西文状态下汉字显示

李 然*
(大连水产学院电子工程系)

摘 要 作者从汉字编码入手,分析了汉字编码的类型和规律,从而得出西文状态下显示汉字的方法。

关键词 汉字编码;字库;汉字显示
中图分类号 TP301.2

随着电脑在中国日益走进千家万户,汉字操作系统也应运而生,目前,成功的汉字操作系统有 UCDOS CCDOS等,其功能较好地满足了人们处理汉字的要求。但是,有时存在这样两个问题: (1) 人们在输出汉字时必须先运行汉字操作系统,而人们对于只有汉字显示没有汉字输入的程序有时并不想在汉字操作系统下执行。(2) 在汉字操作系统下显示的汉字、字型和大小均不能随意改变,满足不了人们在制作高质量界面时的需求。鉴于上述原因,本文介绍一种在西文状态下不需要运行汉字操作系统直接显示汉字的方法,并可以任意地放大和变形,从而使界面更加丰富多采,而做到这一点,首先,要了解汉字的编码从而得出西文状态下显示汉字的原理。

1 汉字的代码

我们从汉字的输入—存储—输出三个方面来讨论汉字的代码。

1.1 汉字的输入

在计算机上使用中文,首先必须把中文信息输入机器中,中文信息的输入通常可以分为自然码输入和编码输入两大类。

1) 自然码的输入

自然码的输入是指对中文文字和中文语音的识别,属于模式识别范畴,也是中文输入的最高形式和最终形式。

文字识别 有印刷体识别和手写体识别两种。主要通过扫描仪进行汉字输入。印刷体识别现在已有北大、清华、四达等几家单位研制成功并通过了技术鉴定,识别准确率

收稿日期: 1996-05-14
* 李然: 196年生,女,讲师,大连 116023

在 90% 左右，基本上可以使用。手写体目前较成功的应属“汉王”系统。

语音识别　随着多媒体技术的发展，汉字的输入有了新的突破，出现了“声数汉语系统”可以识别孤立词语、断续词语和连续词语三种。直观易学，准确率高。目前主要有 SH8型、SH5型、PS/2型等产品。

以上的汉字输入技术，没有充分利用计算机的资源，并且还需要增加象语音系统、扫描仪等价格昂贵的辅助设备，因而，目前不可能成为汉字输入的主流。汉字输入的主要方式应以编码输入为主。

2) 编码输入

编码输入是指将中文编成代码，利用微机上的键盘输入汉字。但是，键盘上只有英文字母、数字和各种符号，并没有汉字，要从键盘上输入汉字，就要用键盘上的各种符号表示汉字。这就是所说的汉字输入码。所谓汉字输入码，就是按照汉字的读音、字形等特征用键盘上一串西文按键给汉字编码，以便从西文键盘上输入汉字。

目前，国内已有几百种编码方案，它们可简要归为以下几种类型:

a. 流水码　按照某种顺序编排的编码。如电报码、国标码、区位码等。

b. 纯音码　直接用英文字母或数字代替拼音作编码。如全拼双音、双拼双音等。

c. 形　码　根据汉字的结构特征，或根据笔画形状进行编码。如首尾码、五笔字型等。

d. 音形码　在拼音的基础上，加上字形信息、部首信息、字义、或部首的读音组成编码。如全息码等。

根据不同的编码，改写原西文 DOS下的键盘输入处理模块，开发出了各种汉字输入软件，使人们可以在愈来愈短的时间内，愈来愈容易地掌握汉字输入技能。

1.2 汉字的存储

通过各种编码输入到计算机内的汉字要有一个统一的存放标准，这个标准就是汉字的“机内码”，也称“内码”。所谓汉字的内码就是计算机内部使用的统一表示汉字的代码。编码方案的不同，输入码的长短和格式也不同，差别很大。每一种编码都对应了专门的键盘输入处理模块，也就是说不管用哪一种编码输入，一个汉字对应的内码总是唯一的。

其实，西文也有内码，这就是 ASCII码，ASCII码是美国国家信息交换标准代码(American Standard Code for Information Interchange)的简称。用 7位二进制数代表 1个字母、数字或符号，这样 ASCII码可以表示 128个字符。其中包括 32个通用控制符，10个十进制数码，52个英文大小字母及 34个专用符号。

既然在计算机内部，汉字信息的处理及存储都是以内码形式进行的，那么内码存放有没有一个统一的标准呢?这个标准又是什么样的呢?为了弄清这个问题，我们还要了解另外两个汉字的代码，“汉字的国标码”和“汉字的区位码”。

为了在汉字系统或通信系统之间交换信息，必须给每个汉字规定一个统一的代码。1981年 5月，我国国家标准总局颁布了《信息交换用汉字编码字符集》，国家标准代号为 GB2312-80，作为汉字交换码编码的国家标准，简称国标码或区位码。同年 5月，已把该标准向 ISO/TC97/SC2进行了登记，这个汉字字型和编码的国家标准规定了 7745个图形字符及它们的二进制编码。其中非汉字有 682个，它包括符号 202个，序号 60个，数字 22个，

拉丁字母 52个, 日文假名 169个, 希腊字母 48个, 俄文字母 66个; 汉字有 6763个, 其中包括一级汉字 3755个, 是最常用的汉字, 一般都知道其读音, 因此按汉语拼音顺序排列 (多音字取它的常用发音, 同音字则以起笔的横、竖、撇、点、折为序), 起笔相同, 则按次笔, 依次类推。二级汉字 3008个, 都是平时不太常用的汉字, 一般不易熟记其发音, 故按部首和笔画排列。

在这种编码方案中, 全部汉字和符号排列在一张 94× 94的方形表中, 纵向为区码或国标码的第一个字节, 横向为位码或国标码的第二个字节, 区码和位码各为一个字节, 一个汉字用二个字节表示。区码在前 (高字节), 位码在后 (低字节)。区位码为十进制, 国标码为十六进制, 区位码均从 1到 94, 而国标码是从 21H到 7EH (00~ 20H及 7FH为空白)。因此, 国标码和区位码之间有着简单的换算方法: 先将区位码换成十六进制数, 再加上 20H即为国标码。即

国标码高字节 = 区码+ 20H,　　国标码低字节 = 位码+ 20H

如: "啊" 在第 16区第 1位, 则 "啊" 的区位码十六进制表示为 10H区和 01H位。因而 "啊" 的国标码高字节 = 10H+ 20H, "啊" 的国标码低字节 = 01H+ 20H, "啊" 的国标码可表示为 3021H。

那么, 内码与国标码或区位码的关系是怎样的呢? 同西文的内码 ASCII码一样, 我国的国家标准 GB2312-80规定, 一个汉字用二个字节表示, 每个字节也只用其中的后 7位, 为了不与 ASCII码相冲突, 保证中英文兼容, 同时又尽可能与国标码保持一致, 规定将国标码的每个字节的最高位置 1,以示此内码作为汉字内码。每个字节的最高位置 1即为每个字节加上 80H。因此

汉字内码高字节 = 国标码高字节+ 80H,　　汉字内码低字节 = 国标码低字节+ 80H

如 "啊" 国标码为 3021H, 则 "啊" 的内码高字节 = 30H+ 80H= B0H, "啊" 的内码低字节 = 21H+ 80H= A1H, "啊" 的内码可表示为 B0A1H。

汉字的机内码、国标码、区位码的关系如下:

区位码+ 20H - -→ 国标码+ 80H- -→ 机内码

即区码 = 内码高字节 - A0H, 位码 = 内码低字节 - A0H。

1.3 汉字的输出

在计算机中汉字用统一的内码表示, 但是 2个字节的内码是不能送屏幕显示, 也不能送打印机打印。

汉字显示时, 要将内码变成点阵码, 所谓汉字的点阵码, 即为汉字的字型信息, 即把一个方块划分为许多小方格 (一般是以 8的倍数为单位, 如 16× 16、24× 24等), 每一个小方格为点阵中的一个点, 把一个汉字置于这样一个方块上, 有笔划通过的小方格视为黑点, 无笔划通过的小方格视为白点。所有黑点就描出了该汉字的字型。用二进制的 1来表示点阵中的一个黑点, 0表示点阵中的一个白点, 那么点阵中的一行黑白点就可以用一个二进制的代码串表示, 若干个代码串就组成了整个汉字的点阵信息, 有时也称为汉字的字模。所有汉字和各种符号的点阵信息就组成了汉字的 "字模库" 简称 "字库"。字库通常有 16× 16、24× 24、32× 32、64× 64等类型, 下面是 16× 16点阵字库中以 "大" 字为

例的点阵和 32个相应的点阵字节[1]。

字节 1 字节 2

```
|—— —— —— —|  |—— —— —— —|
b7.......... b0b7.......... b0
0 . . . . . H H . . . . . .
1 . . . . . H H . . . . . .
2 . . . . . H H . . . . . .
3 . . . . . H H . . . . . .
4 H H H H H H H H H H H H H H H H
5 . . . . . H H . . . . . .
6 . . . . . H H . . . . . .
7 . . . . . H H . . . . . .
8 . . . . . H H . . . . . .
9 . . . . . H H H . . . . . .
A . . . . . H H . . H . . . . . .
B . . . . . H H . . . H H . . . . .
C . . . . . H H . . . . . H H . . . .
D . . . . H H . . . . . H H . . .
E . . . H H . . . . . H H H .
F . . H H . . . . . H H .
b7.......... b0b7.......... b0
|—— —— —— —|  |—— —— —— —|
```

字节	内容 (十六进制)	字节	内容 (十六进制)
1	03	2	00
3	03	4	00
5	03	6	00
7	03	8	00
9	FF	10	FF
11	03	12	00
13	03	14	00
15	03	16	00
17	03	18	00
19	03	20	80
21	06	22	40
23	0C	24	60
25	18	26	30
27	30	28	18
29	60	30	0E
31	C0	32	06

字节 31 字节 32

为了在屏幕上显示汉字,必须得到将要显示的汉字字模的首字节在字库中的位置,即汉字的地址码。它是用来提取汉字字模首字节在字库中存放的逻辑地址的编码。当要向屏幕输出汉字时,必须通过汉字的地址码,才能从汉字库中获取所需的汉字点阵信息,然后在屏幕上获得汉字字型的输出。汉字地址码的设计要考虑与汉字内码有一个简单的映射关系,以便输出汉字时,可根据汉字的内码简捷地从汉字库中获得其字模。通常汉字库内的汉字字模是按一定顺序存放的,故汉字地址码也是连续有序的。地址码与区位码的关系如下:

$$\text{地址码} = [(\text{区号} - m) \times 94 + (\text{位号} - n)] \times N$$

其中,m、n根据不同汉字操作系统下的字库,而取不同的值。N为一个汉字的点阵字节数。

字库可以存放在磁盘上,称为软字库。引导汉字操作系统时,字库全部或部分被送到内存的 RAM中驻留,以便使用。软字库要占用较大的内存空间。字库也可以装在可擦除只读存储器 (EPROM)或只读存储器 (MASK- ROM)中,就是所说的硬字库,俗称“汉卡”。把汉卡插在微机的扩展槽内,作为机器的一个扩充 ROM存储区使用,这种字库

不占内存, 提高了读取速度, 但占用一部分 ROM地址。

16× 16点阵字库主要用于屏幕显示,每个汉字字模占 32个字节, 由于显示刷新缓冲区的格式是水平排列的, 为了避免额外的转换过程,所以,一般做成横向排列的格式。如图 1为一个汉字字型的 16× 16方阵排列情况。

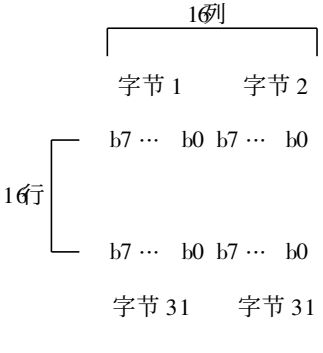

图 1　16× 16点阵字模排列

16点阵字库中每个汉字字模的首字节地址即地址码可通过下式得到:

地址码= 〔(区号 - m)× 94+ (位号 - n)〕× 32

24× 24点阵字库主要用于打印或制作软件界面, 每个汉字字模占 72个字节, 为了避免额外的转换过程而做成纵向排列格式。如图 2为一个汉字字型的 24× 24方阵排列情况。

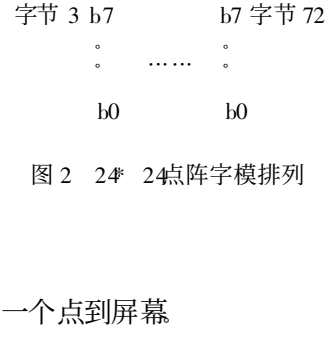

图 2　24× 24点阵字模排列

24点阵字库中每个汉字字模的首字节地址即地址码可通过下式得到

地址码= 〔(区号 - m)× 94+ (位号 - n)〕× 72

2　西文状态下汉字的显示原理

2.1　对点阵信息的处理

对 16点阵字模的处理, 根据图 1, 程序中可用三重循环, 外层分别控制 16行, 中层分别控制每行中的两个字节, 内层分别控制每个字节中的 b7- b0八个位, 判断每个位是 1或是 0,决定写点还是不写点,从而可在屏幕上的相应位置上显示出字型。对 24点阵字模的处理, 根据图 2, 也用三重循环, 外层分别控制 24列, 中层分别控制每列中的三个字节, 内层控制每个字节中的八个位。

2.2　往屏幕上写点的方式

在 IBM PC系列微机及兼容机上, 一般可用两种方法写一个点到屏幕。

1) 利用 ROM BIOS

这种方法的优点是与显示类型无关,通过 BIOS的 10H中断的 12号功能调用实现,编程简便, 但显示速度较慢。例如下面的函数 Writepoint ( ) 就是用该方法将指定颜色为 Color的点写到第 X列,第 Y行的位置上。

```
writepoint (int x, int y)
{
union REGS r;
r. h. ah= 12;
r. h. al= COLOR
```

```
r. x. cx= x;
r. x. dx= y;
int86 (0x10, & r, & r);
}
```

2) 利用 C语言写点函数 Putixel ( )[2]

C语言中函数 Putixel (int x, int y, int color) 的功能是将指定颜色为 Color的点写到第 *X* 列,第 *Y* 行的位置上。

例 下面语句在坐标 (10, 20) 位置上写绿色的点。

```
Putixel (10, 20, GREEN)
```

下面程序将以 Putixel ( ) 函数为基础, 对其稍加变形, 可对 UCDOS中 24点阵字库的汉字进行显示及放大, 并可显示各种颜色、各种字型的汉字, 放大后的汉字无锯齿形状, 平滑美观 程序运行后, 将显示界面 “版权所有, 翻版必究”, 字型为 24点阵黑体字。

本程序使用 Turbo C 2. 0编译, 在 PC /XT 286 386 486及兼容机上运行通过, 有兴趣的读者不访一试。

```
# include<stdlib. h>
# include<dos. h>
# include <graphics. h>
# include<stdio. h>
void printhz1 (int x, int y, char* s, int c);
char sh [] = "版权所有";
char sh1 [] = "翻版必究";
char* sss, * sss1;
main ()
{
int* s, p [] = {130, 140, 510, 140, 510, 320, 130, 320, 130, 140};
int* s11, p1 [] = {130, 140, 510, 140, 520, 130, 120, 130, 120, 330, 130, 320, 130, 140};
int* s2, p2 [] = {510, 140, 520, 130, 520, 330, 120, 330, 130, 320, 510, 320, 510, 140};
int* s3, p3 [] = {120, 130, 520, 130, 530, 120, 110, 120, 110, 340, 120, 330, 120, 130};
int* s4, p4 [] = {520, 130, 530, 120, 530, 340, 110, 340, 120, 330, 520, 330, 520, 130};
int driver, mode;
driver= DETECT;
mode= 0;
initgraph (& driver, & mode, "");
setbkcolor (1);
setcolor (0);
setlinestyle (0, 0, 1);
setfillstyle (1, 4);
s= p;
fillpoly (4, s);
setfillstyle (1, 7);
s11= p1;
fillpoly (6, s11);
setfillstyle (1, 15);
s2= p2;
```

```
fillpoly (6, s2);
setfillstyle (1, 15);
s3= p3;
fillpoly (6, s3);
setfillstyle (1, 7);
s4= p4;
fillpoly (6, s4);
sss= sh;
printhz1 (180, 6, sss, 10);
sss1= sh1;
printhz1 (230, 6, sss1, 10);
setcolor (15);
settextstyle (1, 0, 2);
outtextxy (330, 275," Tel 4671025");
getch ();
closegraph ();
}
void printhz1 (int x, int y, char* s, int c)
{
char buffer [72];
int i, j, k, l [32] [32], p, length;
long ml, mr, m;
FILE* fp;
fp= fopen (" hzk24h"," rb");
if(fp= = NULL) {printf("can not open hzk24h! ");
    exit (0);
    }
setcolor (14);
length= strlen (s) /2;
for (k= 0; k < length; k+ + ) {
ml= s [k* 2] + 256;
mr= s [2* k+ 1] + 256;
m= (ml- 175) * 94+ mr- 254;
fseek (fp, (long) ( (m- 1) * 72), 0);
fread (buffer, sizeof (char), 72, fp);
for (j= 0; j< 23; j+ + )
for (i= 0; i< = 2; i+ + )
{
p= buffer [j* 3+ i];
if ( (p& 0x80)! = 0) l [i* 8] [j] = 1;
else l [i* 8] [j] = 0;
if ( (p& 0x40)! = 0) l [i* 8+ 1] [j] = 1;
else l [i* 8+ 1] [j] = 0;
if ( (p& 0x20)! = 0) l [i* 8+ 2] [j] = 1;
else l [i* 8+ 2] [j] = 0;
if ( (p& 0x10)! = 0) l [i* 8+ 3] [j] = 1;
else l [i* 8+ 3] [j] = 0;
if ( (p& 0x08)! = 0) l [i* 8+ 4] [j] = 1;
else l [i* 8+ 4] [j] = 0;
if ( (p& 0x04)! = 0) l [i* 8+ 5] [j] = 1;
else l [i* 8+ 5] [j] = 0;
if ( (p& 0x02)! = 0) l [i* 8+ 6] [j] = 1;
else l [i* 8+ 6] [j] = 0;
if ( (p& 0x01)! = 0) l [i* 8+ 7] [j] = 1;
else l [i* 8+ 7] [j] = 0;
}
for (j= 0; j< 23; j+ + )
for (i= 0; i< 23; i+ + )
if (l [i] [j] = = 0)
putpixel (2* (j+ 24* k+ 20* (y- 1)), i+ x, 10);
}
fcloseall ();
}
```

参 考 文 献

1 戴梅萼.微型计算机技术及应用. 北京: 清华大学出版社, 1993
2 谭浩强. C语言程序设计. 北京: 清华大学出版社, 1994

From Chinese Code to Showing Chinese in English State

Li Ran
(Deparment of Electronic Engineering, DFU)

Abstract　In this paper, author analyses the kind and law of Chinese code, and finds the method that can show Chinese in English state.

Key words　Chinese code; CClib; Chinese showing