

2.13H 五笔词汇库自动大容量扩充

宋运康 (重庆市大足师范)

摘要: 本文对 2.13H 五笔字型模块和 WM5.0 汉字内部词汇文件进行了全面分析,并在此基础上实现了 2.13H 词汇库自动大容量扩充。实现方案适用于多种汉字系统的五笔字型或非五笔字型词汇的扩充。对相关文件的分析在多方面也具有参考价值。

一、引言

2.13H 是目前国内使用最普遍的优秀汉字系统之一,它的增强屏幕显示、丰富多采的打印功能、内存的低消耗、内部接口的充分开放等优点是其它多种汉字系统不能取代的。由于它挂接了相应的五笔字型模块,使五笔字型输入成为 2.13H 系统多种汉字输入法中速度最快的一种。这些模块成为该系统不可缺少的部分。但是,这些模块容纳的内部词汇较少,据笔者程序统计,共 3113 个。虽然系统也挂接了五笔字型外部造词模块,但这个造词模块只能在系统级使用,用户对外部词汇的管理仅限于增、删,且增和删都只能一个词一个词的手工键入。词汇量少和词汇管理功能的薄弱,大大影响了 2.13H 的汉字输入效率,尤其是 WM5.0 推出之后,较之 WM5.0 的一万多词汇带来的高速输入,这个矛盾更为突出。但另一方面,对于 WM5.0 词汇的大容量带来的汉字输入的高速度,使内存的占有量大大增加,虽有灵活的动态环境取舍相应模块,但这一万多内部词汇作为一个整体却是不能“动态”的,要么至少一万多,要么一个词也没有。且这一万多词汇只能驻于常规内存,使不少软件不能运行。在无扩展内存的机器上,WM5.0 这个优秀的汉字系统基本上不能使用。可见,构造一个用户可以任意切割、拼接的词汇库(它的前提是词汇总量应充分大),以适应不同的用户、不同的软硬件环境需要,显然是很有意义的。

本文以 2.13H 为例,以其原有内部五笔词汇为核心,探索五笔词汇的大容量扩充,以及在大容量扩充基础上的词汇库动态管理。它也适用于多种具备五笔字型汉字输入方案的其它汉字系统。

二、词汇文件分析

解决词汇大容量扩充必须解决:

1. 扩充词汇的组织形式,这种形式应是 2.13H 系统五笔字型模块易于接收的,又方便于词汇动态管理。且最好系统不作或少作修改。

2. 扩充的词汇的获取,词汇的选取应有一定的科学依据。且扩充词汇不应与 2.13H 的五笔内部词汇重合。

对于 1,一个自然的想法是 2.13H 的五笔字型外部词汇文件 WBZXCH.DAT,它不仅可使系统软件基本上不作修改,且易于实现动态切割、拼合。对于 2,可以考虑“移植”WM5.0 的一万多内部词汇到 2.13H,这不但可以使选取扩充词汇的工作量大大减少,而且可能由程序完成移植,从而实现自动扩充。按严格的科学性要求,WM5.0 的词汇的选取可能还有一些问题,但考虑到最近才由国家有关部门批准的词汇库国家标准远未普及,且 WM5.0 在全国范围内有较广泛使用,已经历了一定程度的实践检验,目前选取它作为扩充词汇还是比较合理的。

这里,在编制自动扩充程序前,需要对 WM5.0 的内部词汇所在文件、2.13H 的五笔字型内部词汇所在文件以及 2.13H 的外部词汇文件 WBZXCH.DAT 进行分析。以下简要介绍笔者对这三个文件的有关分析。使用的工具软件是汉化 DEGUG。

WM5.0 的内部词汇所在模块是 WCHRIN.WM5。内部词汇基本上是集中存放于偏离文件头的 02F1H 至 A85DH 处。依次是重码词、两字词、三字词、四字词、多字词,这五类词之间以连续四字节“FF”作为分

界。每一个词都用常规汉字机内码存放,但五类词的存放格式有一点差别。每一个重码词和多字词以一个小于 AOH 的字符打头,作为两词的分界符。三字词和四字词两类,由于每一个词都是等长的,所以词与词间无分界符。二字词有一点特殊,尽管每个二字词等长,因为词首汉字相同的二字词较多,系统为了节省内存,采取按词首汉字分组成串,每组只存一个词首的方式,词首后跟一个

小于 AOH 的字符,再接若干个与词首汉字成词的汉字。这个小于 AOH 的字符,既作为一串二字词的词首汉字标识,又作为不同词首词汇“串”的分界符。五类词的每一类完全按五笔字型外码的字母顺序 A-Y 存放。下面是用 DEBUG 看到 WCHRIN.WM5 的一个片断。

-d3000

5D12:3000	D1 CC 09 B2 DD B4 D1 CC-A8 D2 B5 09 CE F1 D3 E0	烟.草囱台业.务余
5D12:3010	BC A8 D5 A8 09 B5 AF D2-A9 BB D9 B1 AC 07 D5 A8	绩炸.弹药毁爆.炸
5D12:3020	B7 A2 B2 D3 05 C0 C3 B4-D6 0D D0 C4 B1 A9 D7 B3	发灿.烂粗.心暴壮
5D12:3030	B2 DA CF B8 B6 CF 05 B6-A8 B7 DB 05 CB E9 C5 DA	糙细断.定粉.碎炮
5D12:3040	07 B5 AF B1 F8 C8 BC 07-C9 D5 C1 CF C8 DB 05 BB	.弹兵燃.烧料熔.化
5D12:3050	AF D1 D7 07 C8 C8 CF C4-D4 E3 07 B8 E2 CC A3 BB	炎.热夏糟.糕踢火
5D12:3060	F0 0D B3 B5 D1 E6 B2 F1-C2 AF BB A8 CA FD 0F D1	.车焰柴炉花数.学
5D12:3070	A7 C1 BF BE DD C4 BF D7-D6 D6 B5 CC C7 05 B9 FB	量据目字值糖.果

2. 13H 五笔字型内部词汇所在文件是 WBZX.EXE,内部词汇位的起始地址偏移文件头 5BDCH,共 3113 个,由于词汇量小,它似乎未选用重码词汇。它不先以词的汉字个数分组,而是以五笔字型外码的首字母为序,先分为 A-Y25 组,组间有分隔符。每

一组再以二、三、四、多字词的顺序排列。所有内部词汇的每一个词只最后一个汉字用常规汉字机内码存放,其他汉字都采用国际区位码高低字节分别+32 存放。词末汉字与非词末汉字的不同存放形式,兼作为各词间的分隔符。下面是用 DEBUG 看到的一个片断。

-d5ee0

5D12:5EE0	52 3D C9 FA 51 30 52 3D-D1 A7 50 38 52 3D D2 A9	R =生 Q0R =学 P8R =药
5D12:5EF0	50 50 52 3D D4 BA 3A 79-52 55 CA F5 35 87 53 22	PPR =院:yRU 术 5.S''
5D12:5F00	B9 FA 35 45 53 22 C3 F9-34 83 53 22 D0 DB 37 27	国 5ES''明 4.S''雄 7'
5D12:5F10	53 22 D3 EF 42 A4 53 2A-D1 F8 41 E7 53 2A D2 B5	S''语 B ǐ S * 养 A 纆 S * 业
5D12:5F20	1A 2B 56 78 C3 FB 1A F4-56 78 D7 F7 59 34 38 4A	.+Vx 名.触 Vx 作 ¥48J
5D12:5F30	4B 60 CA A1 5E F8 38 6F-43 7C BB AF 5E 05 38 6F	K'省'8oC1 化:8o
5D12:5F40	43 7C BC D2 51 47 39 24-33 4C CA A6 41 E7 39 24	C1 家 QG9 ¥3L 师 A 纆 ¥
5D12:5F50	45 29 D2 B5 55 E7 39 24-49 4C D2 B5 3E F8 39 24	E)业 U 纆 9 ¥IL 业 > 9 ¥

2. 13H 的外部词汇 WBZXCH.DAT 文件由文件头、词汇键码表、词汇表三部分构成。文件头从 00H-PFH 共 16 字节,0-1 字节表示文件词汇总数(低位在前,高位在后,下同),2-3 字节表示词汇键码表首址,4-5 字节表示词码表首址(即键码表末址加 1)。6-7 表示文件中所有词汇的汉字个数;8-9 字节表示文件长度。A-B 字节未用;C-F 为文件标志(5AH、44H、

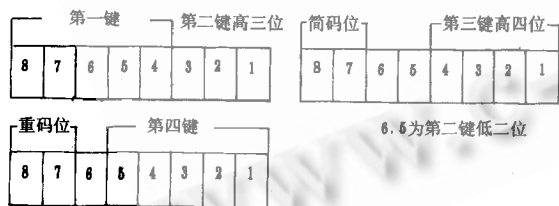
5AH、20H)。

紧接着是词汇键码表,每个词占三个字节。词汇键码表之后是词汇表,每个词占有 2×L 个字节(L 是该词汉字个数),每一个词的存放格式刚好与内部内部词汇表相反,非末尾汉字按常规汉字机内码存放,词尾汉字按国际区位码高低字节分别+32 存放(作为各词间的分界符)。词汇键码表和词汇表的元素顺序成一一对应关系

(三个字节表示的键码对应一个词)。下面是由 WBZXCH.DAT 文件。
DEGUG 看到的一个只 8 个词构成的一个

```
-d100 15e
5D12:0100 08 00 10 00 28 00 1B 00-5E 00 D7 06 5A 44 5A 20 ....(^.^桩.ZDZ
5D12:0110 15 0C 39 22 35 01 A4 38-07 94 03 10 A2 03 03 96 ..9^5.き8.....
..
5D12:0120 1C 23 3D 08 07 CB 24 21-C0 A9 D5 B9 B4 CA BB E3 #=#..水≡!扩展词汇
5D12:0130 B9 DC C0 EB B3 CC 50 72-B4 F3 D7 E3 CA A6 37 36 管理程 Pr 大足师 76
5D12:0140 CE A2 BB FA 4A 52 CC BD-4B 77 D7 D4 36 2F C0 A9 微机 JR 探 Kw 自 6/扩
5D12:0150 33 64 CE E5 B1 CA D7 D6-50 4D B4 CA 3B 63 00 3d 五笔字 PM 词;c.
```

词汇键码表中一词占三个字节,这三个字节遵从五笔字型编码方案表示该词的四次击键。每一键占五个字节位,基本上与五笔字型的主编码表(即国际一、二级共 6764 个汉字的五笔字型编码)相同。第一字节的高五位表示五笔字型的第一个键;第一字节的末三位与第二字节的第五、第六位表示第二 C 键;第二字节的低四位与第三字节的第六位表示第三键;第三字节的低五位表示第四键。另外,第二字节的高两位是简码位,最高位是二级简码位(有二级简码为 1,无二级简码为 0),次高位是三级简码位(有三级简码为 1,无为 0);第三字节的高两位是重码位,最高位表示有无重码(有为 1,无为 0),次高位表示该重码汉字的重码位置(0 表示处在第一位,1 表示不在第一位)。如下图:



只是在词汇键码表中,重码位(2.13H 未取重码词)和简码位(词汇无简码)全为 0。

三、词汇库容量扩充的实现

在上述三个文件分析的基础上,构造一个 2.13H 五笔字型词汇自动扩充程序,解决自动生成词汇键码的问题,是实现词汇自动扩充的关键之一,而解决这个关键必须找到主编码表。容易想象,任何一个可进行五笔字型输入的汉字系统都必须有五笔字型的主编码表。由于五

笔字型编码已经固定,这个编码表一般不应变化。据笔者查找得知,2.13H 的五笔字型主编码表起始位置在 WBZX.EXE 偏离文件头 02bdH 处,WM5.0 主编码表在 Xofn.wm5,超始地址偏离文件头 747 字节。词汇自动扩充程序流程如下:

- 1.读取.13H 全部内部五笔词汇,进而获得这些词汇的键码,为便于与扩充词汇比较,按键码首字母分组;
- 2.读到全部汉字容错码(键码);
- 3.从扩充词汇源文件中读取一个扩充词,进而获取这个词的键码;
- 4.这个键码是否内部词汇键码?是否是汉字容错码?是否与另一扩充词键码相同?如都不是,则按 2.13H 外部词汇文件格式写入外部词汇文件,否则转入上一步,直到读完扩充词汇源文件中的所有词汇。

这里,有几个重要问题必须指出:

1.WBZXCH.DAT 的容量问题。从上面对该文件的内部结构分析可知,该文件在理论上可达 64K。但经笔者多次检验证实,由于系统本身的原因,它最多不能超过 62K(这时,2.13H 的总词汇量约是 11000 个)。超过 62K 但不超过 64K 的 WBZXCH.CAT 可以被外部词汇装载程序 LOAD.COM 调入,且能在输入时正常使用。但超过 62K 的 WBZXCH.DAT 不能接受词汇管理软件 WCH.COM 的管理,一运行 WCH.COM 即死机。要使超过 62K 的外部词汇文件接受 WCH.COM 管理,一是修改 WCH.COM,但最好的是重写 WCH.COM。事实上,WCH.COM 功能也需要增强(特别是它在删除词汇时竟然无被删词汇显示,只能盲删,常常无论正确或错误都频繁鸣笛),在本文对 WBZXCH.DAT 分析的基

基础上,编写一个功能更强的词汇管理程序应是相当简单的了。

2.重码问题。重码包括字词重码和词间重码,字词重码又包括词与字的标准编码相重、词与字的容错码相重。原 2.13H 的内部词汇量少,即无字词重码,又无词间重码。虽然系统允许外部词汇产生重码,但在使用时,对于词间重码,只有提示行正确提示,但此时并不鸣笛,却又要求用户必须输入重码序号,否则系统不断鸣笛报警而不能继续输入,这显然是严重影响快速盲打的。而对于字重码,系统只承认字码而不承认词码,即对该词根本不予理会。所以,必须认真对待重码问题。在 WM5.0 的动态环境中激活显示字词重码项,可知与字的标准码相重的只“深远”一词(与“浣”字相重),但与字的容错码相重的词(比如“等等”与“拜”)以及扩充词与另一扩充词相重的问题必须由自动扩充程序予以解决。

3.自动扩充的速度问题。自动扩充程序的时间花费较多的主要部分是检查扩充词汇是否是原 2.13H 的内部词汇以及是否是重码,一个编制不好的程序可能要运行很长时间。应该进行分组比较,这个分组可按该词或该字的外码(击键码)的首字母分类,这样,可大大缩短时间,使整个扩充工作只 3 分钟左右。

另外,由于 WBZXCH.DAT 中的两个表(键码表与词汇表)元素一一对应,且两表相连,也可以考虑把两个表分别写入两个文件,再进行简单拼接。但要注意的是不能使用文件 COPY 命令,因为键码表中有很多“1AH”,COPY 命令会把这些字节当作文件尾标志而使拼接失败。笔者推荐由扩充程序一次完成。

附自动扩充程序于后(由于较长,不再注释)。它将自动扩充词汇 7713 个(其中二字词 5372 个,三字词 1243 个,四字词 1060 个,多字词 38 个),加上原内部词汇 3113 个,使 2.13H 的总词汇量达 11000 个左右,非重码的 W5.0 全部内部词汇扩充到了 2.13H。且这一万多个词汇几乎无一字词重码和词词重码。显然,扩充词汇的来源不限于 WM5.0,也不限于五笔字型词汇。在完成自动扩充以后,丝毫未减弱原有的造词功能,只是要注意(以前也同样注意)一次进入内存的外部词汇不要大于 64K 罢了。

这个容量充分大的词汇库为词汇库的动态管理提供了一个基点,但 64K 外部词汇并不是动态管理的极限,

而只是一次可进入内存的极限。对于更大的词汇量,显然可以构造一个或多个大于 64K 的词汇库,在使用时由程序自动切割拼凑。根据不同的需求,每个用户都可按自己的要求在进行系统进进行动态切割或拼接。但是,为了程序处理的方便,应该先对所有词汇库的全部词汇进行分类或分级,并把一个或多个外部词汇库中的分类或分级的标志记入一个标志文件或记入动态管理程序本身。动态管理程序在进入系统时给出多种分割、拼装选择,用户选取需要的选择项即可。为了对全部词汇分类或分级的方便,必要时可调入 dbase、Foxbase、wordstar 等软件环境中,以便进行需要的排序、调整、增删等种种操作,满意后再根据自动扩充的思路形成一个或多个外部词汇文件。这样,用户就取得了以系统外部词汇的全部控制,大大方便了词汇的合理利用。

```
label 10,20,30,40;
const s3:array[1..12] of string[6]=('蒙古族','平方米','直辖市',
'宣传科','反对党','众议院','旅行社','科威特','各部委',
'交易所','总代表','准确性');
s4:array[1..6] of string[8]=('持续增长','相互信任','顶天立地',
'尊重人才','义不容辞','身心健康');
var f,f1,f2,f3:file of byte;
p,n,n2,n3,n4,n5,nn,k,aa,bb,i,j:integer;
k1,k2,k3,k4,a1,a2,a3,b1,b2,c1,c2,c3,a,b,c,x:byte;
d:array[1..3113,1..4] of byte;
re:array[1..443,1..4] of byte;
m:array[1..26] of integer;
s1:string[6];
s2:string[8];
r:array[1..200] of byte;
y:array[1..4]of byte;
begin
hires; assign (f1,'d:wbzx');reset(f1);
bb:=0;aa:=1;m[1]:=1;n:=0;p:=¥5bdc;
repeat
seek(f1,p); j:=0;repeat j:=j+2; read(f1,r[j-1], r[j]) until
r[j-1]>160;
n:=n+1;p:=p+j+2;
if j>6 then bb:=1;
if (j=6) and (bb=1) then
begin for k:=1 to 4 do
r[k]:=r[k+2];j:=4;bb:=0;aa:=aa+1;m[aa]:=n end;
if j=4 then
begin
seek(f1,701+(r[1]-48)*94*3+(r[2]-33)*3);
```

```

    read(f1,a1,a2);d[n,1]:=a1 shr 3; d[n,2]:=a1 and 7 * 4+a2
shr 4 and 3;
    seek(f1,701+r[3]=¥b0) * 94 * 3+(r[4]-¥a1) * 3);
    read(f1,a1,a2);d[n,3]:=a1 shr 3; d[n,4]:=a1 and 7 * 4+a2
shr 4 and 3;
    end;
    if j = 6 then
    begin
        seek(f1,701+(r[1]-48) * 94 * 3+(r[2]-33) * 3);
        read(f1,a1);d[n,1]:=a1 shr 3;
        seek(f1,701+(r[3]-48) * 94 * 3+(r[4]-33) * 3);
        read(f1,a1);d[n,2]:=a1 shr 3;
        seek(f1,701+(r[5]-¥b0) * 94 * 3+(r[6]-¥a1) * 3);
        read(f1,a1,a2);d[n,3]:=a1 shr 3;d[n,4]:=a1 and 7 * 4+a2
shr 4 and 3
        end;
    if j = 8 then
    begin
        seek(f1,701+(r[1]-48) * 94 * 3+(r[2]-33) * 3);
        read(f1,a1);d[n,1]:=a1 shr 3;
        seek(f1,701+(r[3]-48) * 94 * 3+(r[4]-33) * 3);
        read(f1,a1);d[n,2]:=a1 shr 3;
        seek(f1,701+(r[5]-48) * 94 * 3+(r[6]-33) * 3);
        read(f1,a1);d[n,3]:=a1 shr 3;
        seek(f1,701+(r[7]-¥b0) * 94 * 3+(r[8]-¥a1) * 3);
        read(f1,a1);d[n,4]:=a1 shr 3;
    end;
    if j > 8 then
    begin
        seek(f1,701+(r[1]-48) * 94 * 3+(r[2]-33) * 3);
        read(f1,a1);d[n,1]:=a1 shr 3;
        seek(f1,701+(r[3]-48) * 94 * 3+(r[4]-33) * 3);
        read(f1,a1);d[n,2]:=a1 shr 3;
        seek(f1,701+(r[5]-48) * 94 * 3+(r[6]-33) * 3);
        read(f1,a1);d[n,3]:=a1 shr 3;
        seek(f1,701+(r[j]-1]-¥b0) * 94 * 3+(r[j]-¥a1) * 3);
        read(f1,a1);d[n,4]:=a1 shr 3;
    end;
    until n = 3113; m[aa+1]:=3114;
    seek(f1,¥5126); i := 0;
    for j := 1 to 498 do
    begin read (f1,a1,a2,a3);if a3 and 31 > 0 then
        begin i := i+1;
            rc[i,1]:=a1 shr 3;
            rc[i,2]:=a1 and 7 * 4+a2 shr 4 and 3;
            rc[i,3]:=a2 and 15 * 2+a3 shr 5 and 1;
            rc[i,4]:=a3 and 31;

```

```

        end;
        end;
        writeln('正在扩充二字词,请等。');
        assign (f,'d:wchrin.wm5');reset(f);
        assign(f2,'d:wbxch.dat');rewrite(f2);seek(f2,¥10);
        assign(f3,'d:char');rewrite(f3);
        seek(f,¥07fa);
        read(f,a,b,c);n2:=0;
        repeat
            i:=0;
            repeat
                read(f,x);if x > 160 then begin i:=i+1;r[i]:=x end;
            until (x < 160)or (x = ¥ff);
            seek(f1,701+(a-¥b0) * 94 * 3+(b-¥a1) * 3);
            read(f1,a1,a2);k1:=a1 shr 3; k2:=a1 and 7 * 4+a2 shr 4 and
3;
            j:=1;
            repeat
                seek(f1,701+(r[i]-¥b0) * 94 * 3+(r[j+1]-¥a1) * 3);
                read(f1,b1,b2);k3:=b1 shr 3; k4:=b1 and 7 * 4+b2 shr 4
and 3;
                for k := m[k1] to m[k1+1]-1 do
                    if (k1 = d[k,1])and (k2 = d[k,2]) and (k3 = d[k,3]) and
(k4 = d[k,4])
                        then goto 10;
                for k := 1 to 443 do
                    if (k1 = rc[k,1]) and (k2 = rc[k,2]) and (k3 = rc[k,3]) and
(k4 = d[k,4])
                        then goto 10;
                c1:=(k2 and 3 * 16)+k3 div 2; c2:=k4+(k3 and 1)shl 5;
                n2:=n2+1; r[j]:=4[j]-¥80;r[j+1]:=r[j+1]-¥80;
                write (f2,a1,c1,c2);write(f3,a,b,r[j],r[j+1]);
                10:j:=j+2;
            until(j=i-1) or (x = ¥ff);
            a:=r[i-1];b:=4[i];
            until x = ¥ff;
        seek(f,¥596a);n3:=0; writeln('正在扩充三字词,请等。');
        repeat s1:= '';
            for j:=1 to 6 do begin read(f,r[j]);if r[j]>160 then
s1:=s1+ch4(r[j])end;
            for j:=1 to 12 do if s1 = s3[j] then goto 20;
            if r[1]=¥ff then goto 20;
            j:=0;
            repeat
                j:=j+2;
                seek(f1,701+(r[j]-1]-¥b0) * 94 * 3+(r[j]-¥a1) * 3);read(f
1,a1,a2);

```

```

y[j div 2]:= a1 shr 3;
if j=6 then y[4]:= a1 and 7 * 4+a2 shr 4 and 3;
until j=6;
c1:= y[1] sh1 3+y[2] shr 2;
c2:=(y[2] and 3 * 16)+y[3] div 2; c3:= y[4]+(y[3]= d[k,3]) and
(y[4]= d[k,4])
for k:= m[y[1]] to m[y[1]+1]-1 do
if (y[1]= d[k,1]) and (y[2]= d[k,2]) and (y[3]= d[k,3]) and
(y[4]= d[k,4])
then goto 20;
for k:= 1 to 443 do
if (y[1]= re[k,1]) and (y[2]= rc[k,2]) and (y[3]= rc[k,3]) and
(y[4]= re[k,4])
then goto 20;
n3:= n3+1;4[5]:= r[5]- ¥ 80;r[6]:= r[6]- ¥ 80;
write(f2,c1,c2,c3);write(f3,r[1],r[2],r[3],r[4],r[5],r[6]);
20:until r[1]= ¥ ff;
seek(f, ¥ 7cc0);n4:= 0;; writeln('正在扩充四字词, 请等。');
repeat se:= '';
for j:= 1 to 8 do begin read(f,r[j]);if r[j]>160 then
s2:= s2+chr(r[j]);end;
for j:= 1 to 6 do if s2= s4[j] then goto 30;
if r[1]= ¥ ff then goto 30;
j:= 0;
repeat
j:= j+2;
seek(f1,701+(r[j]-1)¥ b0) * 94 * 3+(r[j]- ¥ a1) * 3);read(f
1,a1);
y[j div 2]:= a1 shr 3;
until j= 8;
c1:= y[1] sh1 3+y[2] shr 2;
c2:(y[2] and 3 * 16)+y[3] div 2; c3:= y[4]+(y[3] and 1) sh1
5;
for k:= m[y[1]] to m[y[1]+1]-1 do
if (y[1]= d[k,1]) and (y[2]= d[k,2]) and (y[3]= d[k,3]) and
(y[4]= d[k,4])
then goto 30;
for k:= 1 to 443 do
if (y[1]= rc[k,1]) and (y[2]= rc[k,2]) and (y[3]= rc[k,3]) and
(y[4]= rc[k,4])
then goto 30;
n4:= n4+1;
write(f2,c1,c2,c3);r[7]:= r[7]- ¥ 80;r[8]:= r[8]- ¥ 80;
write(f3,r[1],r[2],r[3],r[4],r[5],r[6],r[7],r[8]);
30:until r[1]= ¥ ff;
seek(f, ¥ a495);n5:= 0;nn:= 0;; writeln('正在扩充多字词, 请

```

```

等。');
repeat j:= 0;
repeat read(f,a);
if a= ¥ ff then goto 40;
if a > 160 then
begin
j:= j+1;r[j]:= 1;
if j in [2,4,6] then
begin
seek(f1,701+ (r[j]- ¥ a1) * 3);
read(f1,a1);y[j div 2]:= a1 shr 3;
end
end;
until a > 160;
seek(f1,701+(r[j]-1)¥ b0) * 94 * 3+(r[j]- ¥ a1) * 3);
read(f1,a1); y[4]:= a1 shr 3;
c1:= y[1] sh1 3+y[2] shr 2;
c2:=(y[2] and 3 * 16)+y[3] div 2;
c3:= y[4]+(y[3] and 1) sh1 5;
for k:= m[y[1]] to m[y[1]+1]-1 do
if (y[1]= d[k,1]) and (y[2]= d[k,2]) and (y[3]= d[k,3]) and
(y[4]= d[k,4])
then goto 40;
for k:= 1 to 443 do
if (y[1]= re[k,1]) and (y[2]= rc[k,2]) and (y[3]= rc[k,3])
and (y[4]= rc[k,4])
then goto 40;
n5:= n5+1;nn:= nn+1;
write(f2,c1,c2,c3);r[j]-1:= r[j]-1)¥ 80;r[j]:= r[j]- ¥ 80;
for k:= 1 to j do write(f3,r[k]);
40: until a= ¥ ff;
writeln('n2='n2,'n3='n3,'n4='n4,'n5='n5);
n:= n2+n3+n4+n5;nn:= nn div 2+n2 * 2+n3 * 3+n4 * 4;
close(f);close (f1);close(f3);reset(f3);
for i:= 1 to nn do begin read (f3,a,b);write(f2,a,b) end;
writeln('n='n,'nn='nn);
r[1]:= 10(n);r[2]:= hi(n);r[3]:= ¥ 10;r[4]:= ¥ 00;r[5]:= 1o(n *
3)+y10;
r[6]:= hi(n * 3);
r[7]:= lo(nn);r[8]:= hi(nn);r[9]:= lo(¥ 10+3 * n+nn * 2);r[10]:
= hi(¥ 10+3 * n+nn * 2);
r[11]:= ¥ 00;r[12]:= ¥ 00;r[13]:= ¥ 5a;r[14]:= ¥ 44;r[15]:= ¥
5a;r[16]:= y20;
seek(f2,0);for k:= 1 to 16 do write(f2,r[k]);
close(f2);
writeln('2.13H 汉字系统五笔词汇自动扩充完毕。');
end.

```