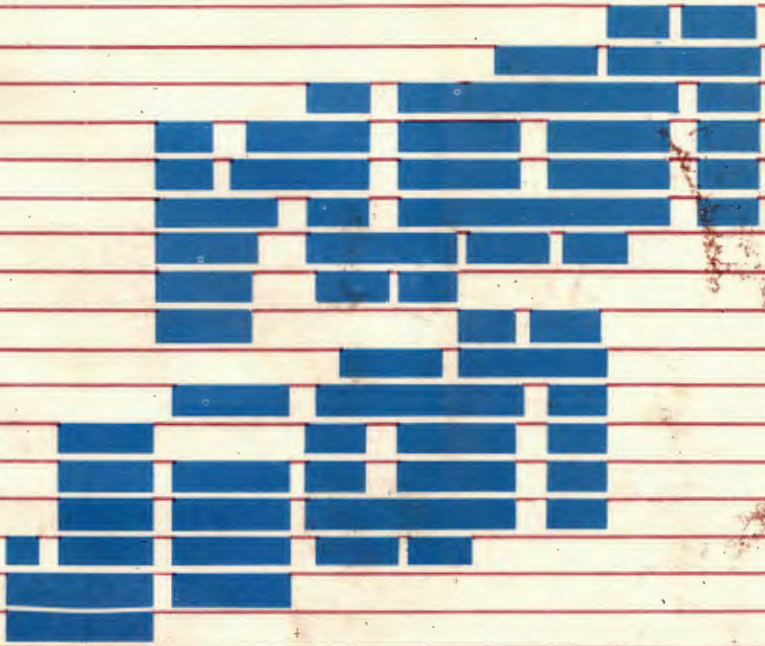
中文软件与软件汉化

郑茂松 著



電子工業出版社

封面设计：阎欢玲

ISBN7-5053-1515-3 / TP · 278 定价：12.50 元

中文软件与软件汉化

郑茂松　著

电子工业出版社

(京)新登字 055 号

内容提要

本书总结和概括中文软件设计与应用的一般原则，强调中文软件与西文软件的区别，讨论软件汉化的原理和方法。

本书为中文软件的开发者提供了软件汉化的基本概念以及各层次软件汉化的基本方法；同时也为中文软件的使用者介绍了各层次中文软件的特点以及一般使用方法。

本书适于计算机应用人员学习提高用，亦可供计算机软件开发与研究人员参考，还可用作计算机专业的教材。

中文软件与软件汉化

郑茂松 著

责任编辑 王惠民

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经售

北京顺义李史山胶印厂制版印刷

开本：850×1168 毫米 1/32 印张：15.25 字数：400 千字

1992 年 2 月第 1 版 1992 年 2 月第 1 次印刷

印数：1～8,000 册 定价：12.50 元

ISBN7—5053—1515—3/TP・278

前　言

随着计算机在我国的推广使用，对中文软件的需求亦日益增多。读者迫切需要有关中文软件与软件汉化方面的专著。本书试图为弥补这方面的空白起到抛砖引玉的作用。

本书力图总结和概括中文软件设计与应用的一般原则，强调中文软件与西文软件的区别，讨论软件汉化的原理和方法。

本书为中文软件的使用者介绍了各层次中文软件的特点以及一般使用方法，同时也为中文软件的开发者提供了软件汉化的基本概念以及各层次软件汉化的基本方法。

全书共有十一章。第一章简述软件的分类、中文软件的开发途径、中文软件与西文软件的主要区别、中文系统的软件层次与汉字处理功能的传递性、多用户系统与网络系统的软件层次。第二章从中文软件设计与应用的角度出发，介绍汉字属性、汉字编码字符集、汉字代码、汉字输入输出方式和设备等中文信息处理的有关概念。第三章从汉字输入、汉字输出、汉字造字、汉卡、中文操作系统支持的中文软件几方面讨论中文操作系统的特点。第四章简述繁体字中文系统与简体字中文系统之间的主要区别、繁体字与简体字兼容的软件包和中文系统。第五章介绍单用户系统、多用户系统和网络系统的主要区别，讨论中文多用户系统和中文网络系统的特点。第六章介绍程序语言与数据库管理系统的基本常识，讨论中文程序语言和中文数据库管理系统的特点。第七章分别介绍字处理软件、数据表软件、CAD／CAM／CAI 与图形软件、组合软件与软件族、统计软件、工具软件的基本常识，讨论中文通用应用软件的特点。第八章概述管理信息系统以及应用系统的开发方法，介绍几种典型的中文应用系统。第九章

总结和概括软件汉化的任务、层次、方式、原理、工具等基本概念。第十章分别讨论微型计算机上和大、中、小型计算机上操作系统汉化的基本方法，从汉字输入处理、汉字输出处理、汉字字库管理三方面介绍汉字处理程序的基本设计原理，并以汉字输入方案的自动生成和汉字打印驱动程序的自动生成为例说明中文操作系统的自动生成方法。第十一章以高位均为 1 的双字节汉字内码为例，从增加汉字处理功能、考虑中文环境下的特殊性问题、提示信息汉化三方面讨论基于中文操作系统的软件汉化的基本方法。

1989 年 8 月，作者曾应新加坡中文与东方语言信息处理学会邀请，在新加坡国立大学讲授了《中文软件设计》课程。本书是在原讲稿基础上整理而成。借此机会，作者向新加坡的朋友们表示衷心的感谢！向支持和关心我的同志们致敬！

由于作者水平有限，难免有错误和不足之处，敬请读者批评指正。

作者

1990 年 5 月　于北京

目　录

第一章　中文电脑的软件层次

电脑(Electronic Brain)和计算机(Computer)是同一事物的两个不同的名称。追溯电脑的发展历史，在电脑发明初期，由于它具有记忆和计算的功能，确实是前所未有的突破，因此称为Electronic Brain，译为电脑。后来人们逐渐发现，它缺乏思维和判断能力，却能迅速、正确、大量地承担计算工作，故又称它为Computer，译为计算机。随着电脑软硬件技术的发展，它的功能已远非仅仅用于计算，特别是随着人工智能、专家系统、知识库的发展，计算机一词已无法准确地描述它的本质特征。况且，人类所追求的目标是：使它具有思维、判断、甚至发明创造的能力，尽可能多地模拟人脑的思维，储存人类智慧和知识，从而越来越多地代替人类的脑力劳动。因此，又觉得把它称为电脑更为贴切。鉴于上述原因，本书将电脑和计算机混为一谈。

中文电脑是能够处理汉字的电脑。中文系统（汉字系统）是能够处理汉字的计算机软硬件系统。中文软件（汉字软件）是能够处理汉字的软件。

电脑中的软件本来就是分层次的，了解软件层次对于软件开发和使用，尤其是对于中文软件的设计与应用都是必要的。

本章总结和概括中文软件的开发途径以及汉化软件与西文软件的主要区别，以中文系统、汉字打印、多用户系统和网络系统为例，阐述中文电脑的软件层次，说明软件功能的传递性。讨论软件层次与汉字处理功能的传递性对中文软件开发和使用的影响。

1.1 软件的分类

软件大致可分为两类：系统软件和应用软件。下面，简要介绍系统软件和应用软件的分类以及它们之间的界限。

1.1.1 系统软件

系统软件一般包括：

1. 操作系统

操作系统是电脑程序系统或软件的核心，相当于乐队的指挥。其主要功能是：管理中央处理机、内存和外部设备，调度作业，处理中断，控制程序的执行等。

操作系统可分为单用户单任务操作系统（例如，DOS）、多任务操作系统（例如，OS／2）、多用户操作系统（例如，UNIX）。

2. 网络软件

由于网络中计算机型号、操作系统、程序语言等各种资源千差万别，要实现不同计算机之间的通信和资源共享，就必须从软件着手制定一套全网共同遵守的约定，这种约定通常称为协议。实现协议的软件就是网络软件。

由于网络软件通常是附加在操作系统之上的，在操作系统基础上扩充了网络功能，因此常常把它称为网络外壳(Shell)。

3. 程序设计语言及其翻译程序

程序设计语言按其语言成分距离机器表示的远近，可分为高级语言和低级语言。

低级语言是面向机器的程序设计语言，包括机器语言、汇编语言和宏汇编语言。低级语言通常是为特定计算机或计算机系列专门设计的。

机器语言是计算机直接使用的程序语言或指令系统，它由一

组可被机器直接识别和执行的机器指令组成。机器指令一般由操作码和操作数地址构成。机器语言程序亦称为机器语言代码，简称为机器代码，这些代码无需翻译便可被机器所接受。

汇编语言是机器语言的符号表示形式，亦称为符号机器语言。它用符号形式表示机器指令，用助记符代替操作码，用标识符代替操作数地址。汇编指令与机器指令基本上是一一对应的。

宏汇编语言是在汇编语言基础上增加宏指令。宏指令是由一组汇编指令定义的，往往可由用户按一定规则自行定义。宏汇编语言与汇编语言有时统称为汇编语言。

汇编语言程序或宏汇编语言程序需要通过汇编程序转换为机器语言程序才能执行。

高级语言是面向算法的程序设计语言。它包括科学计算语言、商用语言、系统程序设计语言、数据库语言、绘图语言等。高级语言程序需要通过编译程序、解释程序、伪编译程序转换为机器语言代码才能执行。

汇编程序、编译程序、解释程序、伪编译程序统称为翻译程序。

值得注意的是，近年来出现的非过程语言，尤其是超高级语言的发展。超高级语言是指函数式语言、逻辑语言、面向对象的语言等非过程语言。这些语言无需规定问题的解法，也用不着描述计算过程，用户只要给出要解决的问题，并提供输入数据和输出形式，便可得到问题的解答。也就是说，它们着重提供关于“做什么”的描述能力，而不着重描述“如何做”的细节。这就使程序设计摆脱了过程式的程序设计概念，例如，地址、内容等等，彻底从冯·诺依曼型(Von Neumann)计算机体系结构解放出来。

4. 数据库和数据库管理系统

数据库是存放数据的“仓库”。直观上说，计算机上使用的“仓库”就是磁盘（软盘和硬盘）、磁鼓、磁带或其它外存储媒

介。

数据库管理系统是操作和管理数据库的软件。它通常包括三个部分:

(1) 数据描述语言及其翻译程序;

(2) 数据操作 (或查询) 语言及其编译程序或解释程序;

(3) 数据管理子程序。

上述三个部分往往是用单一的数据库语言及其翻译程序来实现的。

数据库管理系统适用于数据处理、信息管理、情报检索等领域。

5. 窗口管理软件

窗口管理软件是近年来兴起的一种软件。目前,新开发的一些程序语言都具有窗口设计功能,而且一些系统软件和应用软件均具有窗口提示功能。设置窗口的目的是为了建立友好的用户界面。

独立的窗口管理软件一般是作为外壳(Shell)建立在软件的外层上。例如,Microsoft WINDOWS 就是建立在 DOS 的外层上的窗口管理软件。值得注意的动向是,中文 WINDOWS 的开发。Microsoft WINDOWS 以图形方式显示和打印文字,从而为软件提供了图文并茂的窗口操作环境。它的基本设计思想是:当软件要求显示或打印文字时,都必须由 GDI(Graphics Device Interface)通过显示驱动程序 (Display Driver) 或打印驱动程序(Printer Driver)输出文字。若要输出汉字只要改造这两个驱动程序,在驱动程序中增加查找和识别汉字字形的功能,便可显示和打印出汉字来。关键在于因汉字字量大,如何提高查找和识别汉字的效率,例如,使汉字字库重新排序,并按折半查找法查找汉字。输入汉字仍要承袭原来中文系统的观念。中文 WINDOWS 的开发可能会使中文系统由文本(text)方式转入图形(graphic)方式,从而为程序语言、数据库管理系统、应用软件提供了良好的

中文操作环境，现存的中文操作系统有可能会逐步被淘汰。

6. 公用程序

公用程序有时亦称为实用程序、服务程序。公用软件等。例如，排序程序、合并程序、编辑程序、拷贝程序、打印程序等。

7. 计算机辅助软件工程(CASE: Computer-Aided Software Engineering)

由于计算机应用广泛深入，计算机软件开发的工作量日益增大，致使产生了“软件危机”(Software Crisis)。CASE 提出了解决这种低产量低质量的软件危机的方法，为软件开发者提供了良好的软件开发工具和软件开发环境，从而使软件工作方法更为有效。尤其是大型软件工程，复杂性呈现了非线性增长，就更加迫切需要控制这种复杂性的软件开发工具和软件开发环境。近年来发展的各种应用程序生成器就是一类典型的软件开发工具。

1.1.2 应用软件

应用软件可分为通用应用软件和专用应用软件两大类。

1. 通用应用软件

通用应用软件不是针对某一应用领域的某一具体问题编写的，而是适用于若干应用领域的一类问题的应用软件。通用应用软件通用性很强，具有普遍的使用价值。

目前，微型计算机上常用的通用应用软件包括：字处理软件、数据表软件、组合软件、计算机辅助设计软件、图形软件、统计软件、工具软件等。

2. 专用应用软件

专用应用软件是用来解决专门应用领域中的具体问题而编写的应用软件。例如，工资管理软件、财务管理软件、人事管理软件等。

1.1.3 系统软件与应用软件之间的界限

系统软件与应用软件之间没有一条严格的界限。甚至可以这样说，一切由计算机生产厂家提供的具有普遍使用价值的软件都可算作系统软件。用户为解决特定问题而编写的软件一般不属于系统软件。但是，一旦用户编写了一个通用性很强的软件，具有普遍使用价值，可以提供给其他用户，这样的软件就可以算是系统软件了。例如，有些通用应用软件（字处理软件或数据表软件）亦可称为系统软件，或处于向系统软件过渡的阶段，甚至有些专用应用软件（例如，IUS）也可能会逐渐演变成系统软件。

1.1.4 智能软件

尤其值得一提的是近年来发展起来的智能软件。智能软件应用于知识工程、专家系统、自然语言理解、机器翻译、模式识别、定理证明、计算机视觉、机器人等人工智能领域中。

智能软件与传统程序设计之间的区别如下：

序号	智能软件	传统程序设计
1	以符号处理为主	常常以数值处理为主
2	启发式查找（解答的步骤是隐含的）	算法（解答的步骤是显式的）
3	控制结构通常与定义域的知识分离	控制结构与数据信息结合在一起
4	容易修改和扩充	难于修改
5	常常允许一些不正确的回答	要求正确的回答
6	通常接受满意的解答	通常寻找最佳的解答

1.2 中文软件及其开发

中文软件就是能够处理汉字的软件。在中国推广使用计算机，离不开中文软件。因此，中文软件的开发是至关重要的。

1.2.1 中文软件的开发途径

中文软件的开发有两条途径:

1. 重新设计和开发完全中文化的中文软件

有人主张从操作系统开始，设计出把各种汉字设备等硬件资源都考虑在内的中文操作系统；并主张程序设计语言完全中文化，符合汉语规则，定义汉字数据类型，设置汉字输入输出格式，甚至语句或命令中的关键字或保留字也用汉字书写，例如，如果……，那么……等等，因此要重新设计中文程序设计语言的翻译程序；同样，要重新设计出其它中文系统软件和中文应用软件。这种完全中文化的想法是好的，但却要花费大量人力、物力和时间。更为重要的是，这样的中文软件开发方法同世界上迅速发展的软件技术不相适应，不能充分利用国际上通用的丰富的软件资源，赶不上世界上软件发展的速度，其应用也有很大的局限性。因此，这条途径看来很难走通。

2. 对西文软件实行汉化

另一条途径是目前大多数人广泛采用的，就是对原有西文软件实行汉化。

所谓汉化，就是汉字化，就是使西文软件增加汉字处理功能，同时保持中西文兼容性，保持西文软件的原有功能。经过汉化的软件，称为汉化软件，泛称为汉字软件或中文软件。

1.2.2 汉化软件与西文软件的主要区别

汉化软件与西文软件的主要区别在于：汉化软件对西文软件增加了汉字处理功能。

汉字处理功能包括:

(1) 汉字输入

通过汉字输入设备输入汉字，并把它转换为汉字内部码;

(2) 汉字信息的加工处理

各类软件对汉字内部码进行各种加工处理，例如，并置、截取、插入、删除、替换、传送、比较、排序、查找等操作；

(3) 汉字输出

把加工处理后的汉字内部码转换为汉字字形信息，通过输出设备输出汉字。

为了保持西文软件的原有功能，达到中西文兼容，同时为了使软件汉化工作量达到最低限度，对西文软件尽量少做修改甚至不做修改便可处理汉字，常常遵循下列两项原则：

(1) 选择汉字内码尽量保持与西文字符编码在信息加工处理上的一致性，因此，可使汉字在计算机内部的表示、在磁盘上的表示、在通信媒介上的表示，与西文字符没有本质的差别；

(2) 在程序语言等软件中不另外设置单独的汉字数据类型，而把汉字纳入程序语言等软件中常有的字符类型或字符串类型，把汉字与西文字母一样看待，只不过在计算字符个数时，一个汉字视为两个字符，因此，可使汉字与西文字母的处理原则上没有本质的差别。

基于上述两项原则，汉字与西文字符在信息加工处理方面就没有本质的区别，因而汉化软件与西文软件的主要区别是，汉字的输入输出（例如，汉字的键盘输入、显示和打印等）与西文字符的输入输出不同。因此，汉化软件对西文软件增加汉字处理功能，主要是增加汉字输入输出功能。

1.3 中文系统的软件层次

1.3.1 软件层次与软件功能的传递性

电脑中的软件是分层次的。了解软件层次，对于使用和开发软件都是有好处的。

单用户系统的软件层次由低到高大致可以分为以下几层：

(1) 操作系统;

(2) 程序语言、数据库管理系统、通用应用软件的翻译程序（以下简称为翻译程序);

(3) 应用程序。

如图 1.1 所示。

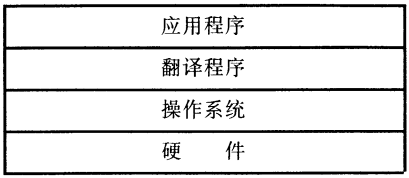


图 1.1　单用户系统的软件层次

就象建筑楼房一样，高层软件（外层软件）建立在低层软件(内层软件）基础上，低层软件支持高层软件。

低层软件的功能能够传递给高层软件。也就是说，高层软件依赖于低层软件的功能，受低层软件功能的影响和制约。例如，在 DOS 和 UNIX 上开发的同一程序语言的编译程序也不相同。我们把这种性质称为软件功能的传递性。

1.3.2　中文软件层次与汉字处理功能的传递性

中文系统与西文系统的软件层次是一致的。在中文系统中，不同层次的中文软件有着迥然不同的使用和开发方法。

根据软件功能的传递性，在中文系统中，低层软件的汉字处理功能能够传递给高层软件。也就是说，高层软件依赖于低层软件的汉字处理功能，受低层软件汉字处理功能的影响和制约。因此，一方面，高层软件的开发和使用要考虑低层软件汉字处理功能的传递性；另一方面，低层软件的开发和使用，也要考虑它本身的汉字处理功能对高层软件的传递性。由此可见，了解中文系统的软件层次与汉字处理功能的传递性，对于开发和使用中文软件是必要的。

如果低层软件是中文软件，那么，一方面，高层软件可以继承它的汉字处理功能；另一方面，高层软件可以通过自身的修改或扩充汉字处理模块来补充增加无法从低层软件获得的汉字处理功能。如果低层软件是西文软件，那么，高层软件只能通过自身

的修改或扩充汉字处理模块来重新建立汉字处理功能。本节将分别简要说明如何在中文系统的三个软件层次上建立汉字处理功能。

操作系统的使用界面是系统功能调用界面；程序语言、数据库管理系统、通用应用软件的使用界面是命令、语句、函数，应用程序的使用界面是它所形成的最终用户界面。在各个软件层次上建立汉字处理功能的目的是，使各个层次软件的使用界面具有汉字处理功能。

1. 中文操作系统

操作系统通常是通过扩充汉字处理模块来建立汉字处理功能的，主要是通过扩充汉字设备驱动模块来建立汉字输入输出功能的，例如，汉字键盘输入功能、汉字显示功能、汉字打印功能等。由于操作系统是计算机系统的软件核心，因此，整个计算机系统中的各层次软件都依赖于中文操作系统的汉字处理功能，受中文操作系统汉字处理功能的影响和制约。例如，如果中文操作系统提供了某一种汉字输入方法，那么它所支持的程序语言、数据库管理系统、通用应用软件，甚至应用程序界面，均可使用这种汉字输入方法输入汉字；否则，它所支持的高层软件就很难使用这种汉字输入方法来输入汉字。

2. 中文程序语言、中文数据库管理系统和中文通用应用软件

如果程序语言、数据库管理系统、通用应用软件在中文操作系统支持下开发和使用，那么，一方面，它们能够利用原有的系统功能调用来继承中文操作系统的汉字处理功能；另一方面，它们应当通过对翻译程序进行修改或扩充汉字处理模块来补充增加无法从中文操作系统获得的汉字处理功能。例如，在中文操作系统支持下，尽管在大多数场合下高层软件能够输入输出汉字；但在某些场合下，比如，有的西文软件禁止非可打印字符的输入输出，从而也阻碍了汉字的输入输出，为此，必须通过软件汉化把

输入输出字符的允许范围扩大为包括汉字，使软件在这些场合下也能够输入输出汉字。又例如，针对西文软件未考虑汉字绕回问题的缺陷，通过软件汉化避免末端出现半个汉字的禁则现象，用以增加汉字的绕回功能。

如果程序语言、数据库管理系统、通用应用软件在西文操作系统支持下开发和使用，那么，由于西文操作系统无汉字处理功能，它们根本无法从西文操作系统获得汉字处理功能，因此，必须在它们的翻译程序中重新建立汉字处理功能，通常是通过扩充汉字处理模块来实现的。

3. 中文应用程序

如果用中文程序语言，中文数据库管理系统、中文通用应用软件编写应用程序或处理数据，那么，应用程序及数据是通过使用它们提供的命令、语句、函数来继承中文程序语言、中文数据库管理系统、中文通用应用软件甚至中文操作系统的汉字处理功能的。例如，在 BASIC 程序中使用 LPRINT 语句和 CHR $ ( ) 函数可以继承中文程序语言和中文操作系统的汉字打印功能，在 dBASE 程序中使用? 命令、@命令和 CHR( )函数也可以继承中文数据库管理系统和中文操作系统的汉字打印功能（见 1.3.3 节 1.)。又例如，在 Wordstar 或 PE 处理的文本文件中插入 Esc 序列命令可以继承中文字处理软件和中文操作系统的汉字打印功能(见 1.3.3 节 2.)。(广义地说，各种文件都可以看成是数据文件，即使是程序文件，也可看成是数据文件。程序文件是由西文字符和汉字组成的。对程序文件施加的操作是建立、修改和执行。从这个意义上说，文本文件、图形文件、图象文件等也均可认为是数据文件)。

应用程序及数据除了通过使用程序语言、数据库管理系统、通用应用软件提供的有关汉字处理的命令、语句、函数继承低层中文软件的汉字处理功能外，还可以通过使用它们提供的外部接口命令、语句或函数来调用汉字处理模块，用以在应用程序及数

据这一层上补充增加无法从低层中文软件获得的汉字处理功能。例如，在 dBASE 中可用 RUN 命令执行操作系统命令和外部程序，用 CALL 命令调用汇编语言程序，用以在应用程序这一层上调用汉字排序程序来扩充汉字排序功能（见 6.5.1 节 6.和 11.1.5 节），或调用中文打印公用程序来扩充汉字多字体打印功能（见 1.3.3 节 2.）。特别是，通过调用汇编语言程序，可使对外部设备的输入输出控制、对字节和位的处理、对一些逻辑运算获得较高的效率和速度。

如果用西文程序语言、西文数据库管理系统、西文通用应用软件编写应用程序或处理数据，那么，由于西文程序语言、西文数据库管理系统、西文通用应用软件中无汉字处理功能，应用程序和数据对它们的汉字处理功能的依赖性也就无从谈起。因此，必须在应用程序这一层上重新建立汉字处理功能，通常是通过使用程序语言、数据库管理系统、通用应用软件提供的外部接口命令、语句或函数调用汉字处理模块来实现的。例如，在 BASIC 程序中通过使用 CALL 语句直接调用汉字处理模块，用以在应用程序这一层上获得汉字输入输出功能及其它汉字处理功能（见第十章）。

1.3.3 汉字打印的软件层次

汉字打印功能是汉字处理功能的一部分。由于汉字打印功能是在中文系统的不同层次上建立的，因此汉字打印功能也具有不同的软件层次。

本节通过讨论汉字打印的软件层次，作为补充说明中文系统的软件层次的一个例子。

汉字打印的软件层次大致可分为以下几个层次：

(1) 中文操作系统的汉字打印功能；

(2) 中文程序语言、中文数据库管理系统、中文通用应用软件的汉字打印功能；

(3) 中文打印公用程序;

(4) 中文计算机排版系统。

下面，逐一介绍汉字打印的这几个软件层次上的汉字打印功能。

1. 中文操作系统的汉字打印功能与中文程序语言、中文数据库管理系统、中文通用应用软件的汉字打印功能

中文操作系统的汉字打印功能是由它的汉字打印驱动程序实现的。这些汉字打印功能可以传递到它所支持的程序语言、数据库管理系统、通用应用软件，甚至应用程序。为了适应不同型号打印机，需要配备不同的汉字打印驱动程序，因而中文操作系统提供的汉字打印功能可能会随之有所不同。

中文程序语言、中文数据库管理系统、中文通用应用软件的汉字打印功能，可由下列两种途径来实现:

(1) 依赖中文操作系统的汉字打印功能，并对程序语言、数据库管理系统、通用应用软件补充增加无法从中文操作系统获得的汉字打印功能。

(2) 在西文操作系统支持下，对程序语言、数据库管理系统、通用应用软件重新建立汉字打印功能。

在应用程序中可以利用程序语言、数据库管理系统、通用应用软件、甚至操作系统原有的命令、语句、函数，来使用中文程序语言、中文数据库管理系统、中文通用应用软件、甚至中文操作系统的汉字打印功能。

例如，在联想式汉字系统支持下的 BASIC 程序可通过 LPRINT 语句和 CHR $( )函数来使用中文程序语言和中文操作系统的汉字打印功能。BASIC 程序如下:

程序 A

```
10 LPRINT CHR $ (27);"@"
20 LPRINT CHR $ (27);"F";CHR $ (2);CHR $ (2);CHR $ (2);
30 LPRINT "中文与东方语言信息处理学会"
```

```
40 LPRINT CHR$(27);"F";CHR$(4);CHR$(4);CHR$(3);
50 LPRINT "中文电脑讲座"
```

程序 B

```
10 LPRINT CHR$(27);"A"
20 LPRINT CHR$(27);"F2 2 2";"中文与东方语言信息处理学会"
30 LPRINT CHR$(27);"F4 4 3";"中文电脑讲座"
```

联想式汉字系统的汉字打印驱动程序提供了两种形式的中文打印命令：Esc+"@"置第一种形式，在这种形式下，打印命令中的参数都要按二进制字节串方式输入；Esc+"A"置第二种形式，在这种形式下，打印命令中的参数都用十进制数字串方式输入，而且每个参数后都必须有一个空格。程序 A 和程序 B 分别使用了第一种形式和第二种形式的字形设置命令。字形设置命令以 F 开头，后面的三个参数分别表示水平放大系数、垂直放大系数和字间距。程序 A 和程序 B 的打印结果相同，即"中文与东方语言信息处理学会"的字形横向和纵向各放大 2 倍，字间距为 2；"中文电脑讲座"的字形横向和纵向各放大 4 倍，字间距为 3。如下：

中文与东方语言信息处理学会

中文电脑讲座

同样，在联想式汉字系统支持下的 dBASE 程序亦可通过?命令，@命令和 CHR( )函数来使用中文数据库管理系统和中文操作系统的汉字打印功能。dBASE 程序如下：

程序 C

```
SET PRINT on
? CHR(27)+"A"+CHR(27)+"F2 2 2 中文与东方语言信息处理学会"
```

```
? CHR(27)+"A"+CHR(27)+"F3 3 2 中文电脑讲座"
? " "
SET PRINT off
```

程序 C 在? 命令中使用第二种形式的中文打印命令，打印结果是："中文与东方语言信息处理学会"的字形横向和纵向各放大 2 倍，字间距为 2；"中文电脑讲座"的字形横向和纵向各放大 3 倍，字间距为 2。如下：

中文与东方语言信息处理学会
中文电脑讲座

程序 D

```
SET DEVICE TO print
@ PROW( ), 1 SAY CHR(27)+'A'+CHR(27)+'F3 3 2 中文与东方语言信息处理学会'
@ PROW( )+1, 1 SAY ' '
SET DEVICE TO screen
```

程序 D 在@命令中使用第二种形式的中文打印命令，打印结果是："中文与东方语言信息处理学会"的字形横向和纵向各放大 3 倍，字间距为 2。如下：

中文与东方语言信息处理学会

又例如，在 CCDOS 4.0 中也有类似的字形变换命令，它采用向打印机发送序列 Esc+"I"+"X"来实现字形变换，其中，X 为大写字母 A 到 N 表示 14 种不同的字形，X 为小写字母 a 到 n 分别表示 A 到 N 的加重字。在 CCDOS 支持的程序语言或数据库管理系统中亦可通过 CHR $( )或 CHR( )这样的函数来调用

字形变换命令，实现打印字形的变换。

前面，我们已经说明，在应用程序中如何使用中文程序语言、中文数据库管理系统和中文操作系统的汉字打印功能。下面，我们以字处理软件为例，来说明在数据文件中如何使用中文通用应用软件和中文操作系统的汉字打印功能。具体地说，在字处理软件处理的文本文件中如何引入中文操作系统的打印命令，从而在打印文本文件时按照中文操作系统的汉字打印命令去格式化打印文本文件。

例如，在 CCDOS 4.0 支持下的中文 WORDSTAR 中打印汉字。

CCDOS 4.0 采用向打印机发送序列 Esc+″I″+″X″来实现字形变换，其中，X 为大写字母 A–N 表示 14 种不同的字形，X 为小写字母 a–n 表示 A–N 的加重字。由于西文 WORDSTAR 中不接受 Esc+″I″+小写字母，因此要在 WORDSTAR 中打印加重字有以下两种解决办法:

(1) 汉化 WORDSTAR，使它接受序列 Esc+″I″+小写字母;

(2) 在打印驱动程序中采取变通方法：在 WORDSTAR 中，所有 14 种未加重字形(A–N)仍采用 Esc+″I″+″X″进行字形变换，而用序列 Esc+″I″+″X″+ESC+″I″+″O″表示 X 字形的加重。

在 WORDSTAR 中编辑文本文件，键入:

″Ctrl″+″P″+″E″+″中文与东方语言信息处理学会″
″Ctrl″+″P″+″E″+″Ctrl″+″P″+″X″+″中文电脑讲座″

则屏幕显示:

^E 中文与东方语言信息处理学会

由于在 WORDSTAR 中，E 对应 DOS 中的 B，X 对应 DOS 中的 O，因此相当于向打印机发送下列序列：

Esc+″I″+″B″+″中文与东方语言信息处理学会″
Esc+″I″+″B″+Esc+″I″+″O″+″中文电脑讲座″

从而在 WORDSTAR 中打印该文本文件时，打印出轻重两行汉字，第一行是 B 形汉字，第二行是 B 形对应的加重字。

又例如，在联想式汉字系统支持下的 PE 中打印汉字。

若要把文本文件中的汉字横向放大 3 倍，纵向放大 2 倍，字间距为 4，则可用 PE 在汉字左边插入：

＜ESC＞A＜ESC＞F3 2 4

其中，＜ESC＞是一个控制字符，其值为十进制 27，它必须这样键入：按 Alt−x 键，在屏幕第 25 行出现提示：Type a character，接着用右边的小键盘依次键入 Alt−0, Alt−2, Alt−7。显然，这样键入＜ESC＞不仅麻烦，而且受限颇多，比如 LOTUS 1−2−3 和行编辑程序都不能这样键入＜ESC＞。联想式汉字系统允许用户定义＜ESC＞的等效字符。定义方法是：在 LXPC.PRO 文件中加入一条

def ＜键码＞＝prt−ctrl

其中，＜键码＞就是等效字符的扫描码，即每个键按其在键盘上所处的位置规定的序号。应尽量选择不常用的键，比如，选“、”代替＜ESC＞，该键的扫描码为 2960，在 LXPC.PRO 文件中加入一条

def 2960＝prt−ctrl

之后，凡是要键入＜ESC＞的地方均可用键入“、”来代替。

2. 中文打印公用程序

中文打印公用程序多半用于多字体打印或汉字报表打印，本身带有多字体字库，用于扩充操作系统打印功能，提供高质量的汉字打印输出。用户需要事先在要打印的汉字文本文件中插入格式命令，中文打印公用程序解释执行文件中提供的格式命令，用以在打印过程中进行字体变换和格式控制。

然而，中文打印公用程序与中文操作系统的汉字打印驱动程序不一样，它的打印功能无法传递到程序语言、数据库管理系统和通用应用软件以及应用程序，因此在应用程序中无法利用程序语言、数据库管理系统、通用应用软件原有的有关汉字处理的命令、语句、函数来使用它，而只能利用外部接口命令、语句或函数来调用它。例如，可在 dBASE 程序中用 RUN 命令调用执行中文打印公用程序。

下面，以我们开发的 dBASE 通用多字体报表系统 dGMRS 为例，来说明在 dBASE 程序中如何用 RUN 命令调用多字体打印程序（中文打印公用程序）来打印多字体报表。

dBASE 提供了很强的报表格式设计和报表输出功能，但是由于它所产生的报表不合乎中国人的习惯，因此在国内的使用率很低。其原因是：一则，dBASE 提供的报表格式不带表格线，而中国人习惯于表格处理；二则，一般的中文 dBASE 只能打印一种汉字字体。鉴于上述原因，国内目前开发了很多通用报表系统。这些通用报表系统大多首先将数据库文件转换为报表系统的表格形式，然后再打印出报表来。然而，这些通用报表系统是独立于数据库管理系统，尚需对数据库文件进行转换，不能在 dBASE 中直接打印报表，使用不大方便；另外，大多数通用报表系统都未采用多字体打印。

dBASE 通用多字体报表系统 dGMRS（如图 1.2 所示）旨在解决上述问题。这个系统的特点是：

(1) 它是 dBASE 的子系统，建立报表格式文件和打印报表的全部过程均在 dBASE 内直接完成；

(2) 以多字体的形式打印报表。

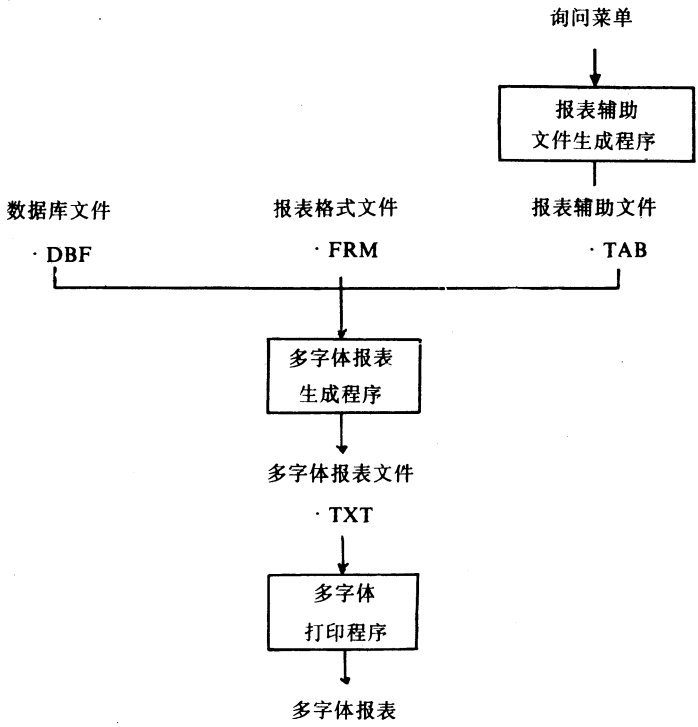


图 1.2 dBASE 通用多字体报表系统

dGMRS 充分利用 dBASE 原有的 CREATE / MODIFY REPORT 和 REPORT FORM 命令，在原报表格式文件的基础上增加了一个报表辅助文件。通过询问菜单，由用户提供以下信息：段页格式控制，字体变换，设计字距，打印页标志，字形自由变倍打印，制表（表格线不占有效行），对表格的标题进行字体的改变，对表格内字体进行改变等等。该系统根据这些信息生成报表辅助文件，然后把数据库文件、原报表格式文件、报表辅

助文件三者结合形成多字体报表文件，最后在 dBASE 程序中通过 RUN 命令调用多字体打印程序，便可打印出既具有表格线又含有多字体的中国式报表来。

3. 中文计算机排版系统

计算机排版系统是由计算机控制的自动排版系统。编辑人员将排版内容及组版格式输入计算机，计算机通过计算机排版软件自动安排版面，然后用激光印刷机印刷，或用照排机自动制成版面，供印刷使用。

计算机排版系统可分为两大类：专业性计算机排版系统和非专业性计算机排版系统。专业性计算机排版系统早在 1965 年就出现了。近年来，非专业性计算机排版系统迅速发展，常称为桌上排版系统（DTP：Desk Top Publishing System）。

中文计算机排版系统自带汉字字库，自含汉字印刷功能，因此可独成系统，并可基于西文操作系统。当然，亦可在中文操作系统上使用，充分利用中文操作系统所支持的字处理软件或编辑程序以及原有汉字字库。由于中文计算机排版系统需要昂贵的印刷设备，而且所占内存和磁盘空间很大，执行速度也较低，因此不适于用作中文打印公用程序，更不适于用作中文操作系统的打印驱动程序。

1.4 多用户系统与网络系统的软件层次

多用户系统与网络系统的软件层次，主要是指多用户功能或网络功能的层次。在多用户系统和网络系统中，不同的软件层次，有着不同的多用户功能或网络功能的使用和处理方法。因此，了解多用户系统和网络系统的软件层次，对于在多用户系统和网络系统环境中使用和开发软件是必要的。

多用户系统与网络系统的软件层次由低到高大致可以分为以下几层：

(1) 多用户操作系统或网络软件;

(2) 程序语言、数据库管理系统或通用应用软件的翻译程序;

(3) 应用程序。

在多用户系统和网络系统中，不同软件层次上的软件均应设置各自的多用户功能或网络功能。假设只在多用户操作系统中设置多用户功能，或只在网络软件中设置网络功能，而不在它所支持的程序语言、数据库管理系统或通用应用软件中设置多用户功能或网络功能，或者尽管在程序语言、数据库管理系统或通用应用软件中设置了多用户功能或网络功能，但在编写应用程序或处理数据的过程中未使用相应的命令、语句和函数，则应用程序或数据无法实现多用户功能或网络功能，无法有效地处理硬件资源、软件资源和数据资源的共享，无法有效解决多用户环境和网络环境中的数据保护、安全保密、出错处理等问题。

多用户操作系统本身具有多用户功能。网络软件中含有网络功能，它以加外壳的方式扩充操作系统的功能。实现程序语言、数据库管理系统或通用应用软件的多用户功能或网络功能有两种方法：一是直接改造原单用户软件，例如，网络 dBASE Ⅳ，网络 LOTUS 1-2-3 等；二是在原单用户软件基础上加一层网络外壳，例如，网络 IUS 等。应用程序的多用户功能和网络功能是通过使用程序语言、数据库管理系统、通用应用软件中的多用户或网络命令、语句和函数编写程序来实现的。

第二章　中文信息处理的基本概念

中文信息处理是一种文字信息处理。概括起来，可把文字信息处理过程分为三个阶段:

(1) 信息输入：通过输入设备把文字信息输入计算机，并转换为代码;

(2) 信息加工处理:根据各类不同的应用，借助预先设计好的软件，对代码进行加工处理;

(3) 信息输出：把加工处理后的代码转换为文字信息，通过输出设备输出文字。

中文信息处理与西文信息处理相比较，一方面，从信息处理的角度看，中文信息处理与西文信息处理没有本质上区别，原则上现有的西文信息处理系统都可用来处理中文信息；另一方面，由于汉字属性（汉字字量、字形、字音、字义等）的特殊性，导致了中文信息处理（主要是汉字的输入输出处理）的复杂性，因此，西文信息处理系统必须经过适当的改造才能处理中文信息。

图 2.1 给出了中文信息处理过程的示意图。

根据图 2.1 中文信息处理流程中涉及到的基本内容，本章首先介绍汉字属性，然后介绍汉字编码字符集，再后介绍各种汉字代码，最后分别介绍汉字输入方式和汉字输出方式。本章仅介绍中文软件设计与应用所涉及到的中文信息处理的基本概念，而不是专门讨论中文信息处理系统。

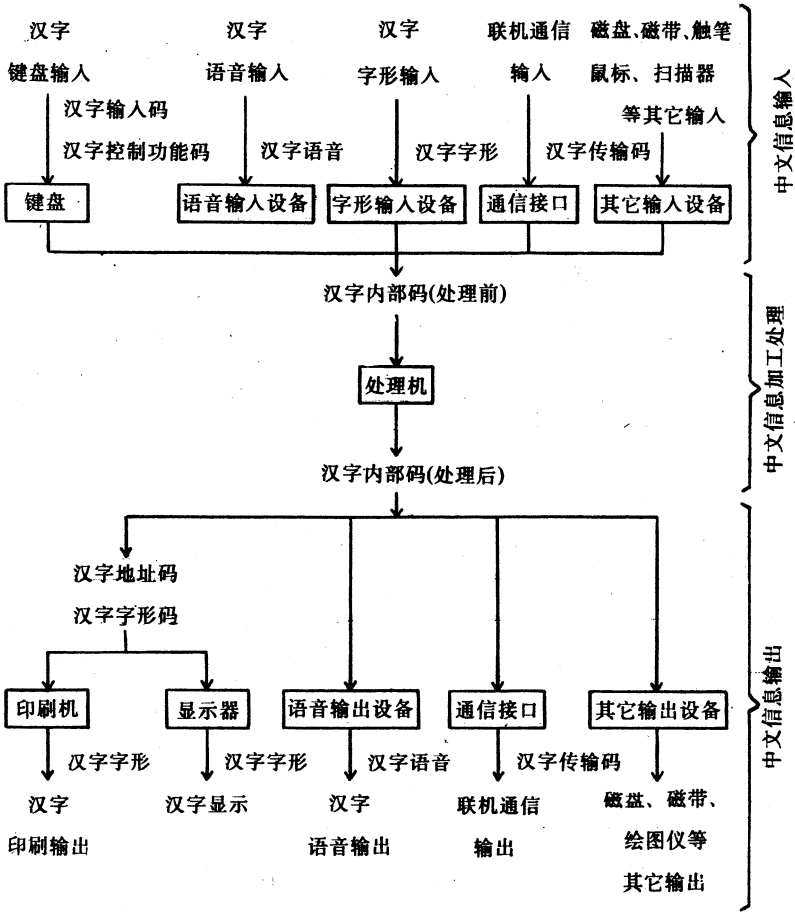


图 2.1　中文信息处理流程图

2.1 汉字属性

中文电脑的主要处理对象是汉字。中文软件与汉字属性有着密切的关系。因此，要研究中文电脑和中文软件，首先要了解汉字属性。

汉字属性是指汉字的一些基本特性，例如，汉字字量、字形、字音、字义等。本节将对这些基本概念分别予以简要介绍。

2.1.1 汉字字量

在中文电脑中，选用多少个汉字，是中文信息处理的首要问题。

汉字是表意文字，即象形文字，它的每个字母有其特有的形状和构造，这是与各种拼音文字（例如，英文）的主要区别。最早的汉字是甲骨文字，约有3000多个。东汉的《说文解字》收字10516个。直到1916年清代《康熙字典》收字42176个。据估计，累计到现在，古今汉字字量多达六万多个。然而，目前实际使用的汉字，只不过六、七千字，其它五万多个生僻字已被淘汰。在当代口语化的书面语言中，日益新生滋长的已不是单字，尤其不是从前那种以生僻自炫的单字，而是能用少数单字灵活组成的新词。目前，汉语的发展趋势是：汉字造字奄奄一息，汉字造词势在必行。

在中文电脑中，究竟选用多少个汉字，选用哪些汉字，最终体现在汉字编码字符集上。我国颁布的汉字编码字符集的基本集共收录汉字6763个，包括常用汉字3755个，次常用汉字3008个。除此之外，汉字编码字符集的两个辅助集共包括稀用字和罕用字一万六千余个。台湾省的通用汉字标准交换码共收录汉字13051个，包括常用字5401个，次常用字7650个。

汉字的选用范围是根据汉字的使用频度来决定的。按照汉字的使用频度，可把汉字分为常用字、次常用字、稀用字、罕用字等几个等级。中国的汉字编码字符集的基本集收录的一级汉字就是常用字，二级汉字就是次常用字，两个辅助集分别是稀用字和罕用字。台湾省的通用汉字的标准交换码收录了常用字和次常字两级汉字。

汉字使用频度的统计结果，能够正确反映大多数汉字的使用情况，但也有下述特殊情况：

(1) 由于汉字使用频度的统计结果受空间和时间的限制，致使有些统计结果是不准确的。例如，有的汉字在不同的使用环境中的使用频度差异很大；同时，汉字的使用频度也受时代背景的影响而使统计资料受到干扰。

(2) 同一汉字在不同专业领域中使用时，其频度也有差异。有些汉字在某一专业中常用，而在其它专业中却不常用，甚至不用。因此，对不同专业的汉字使用频度要分别进行统计。在若干有代表性的专业领域内统计出各自所用汉字的频度后，求取平均值，从而得到综合频度。

2.1.2 汉字字形

汉字字形是汉字的一个重要属性。研究汉字字形是为了找出汉字的结构规律，以便为中文电脑在字形信息处理方面提供依据。例如，

(1) 在汉字输入方法中，有很多键盘输入编码方案是基于字形特性的，比如，五笔字型、仓颉码等。尤其是，汉字字形输入方式，更是与汉字字形密切相关。

(2) 汉字字符集中的汉字基本上是按字形规律排列的。比如，我国大陆和台湾公布的汉字编码字符集基本上是按部首和笔画排序的。

(3) 在建立汉字字库时，特别是在建立用字根合成的汉字字

库时（又称部件组字法），通过对字形的分析研究，可以选择最少数量的字根，合理地组合所需的汉字，从而减少存储空间，加快存取速度。

汉字字形是汉字形体结构的图象。汉字是由字根构成的，字根是由笔画构成的，笔画又是由位点构成的。本节将对这些基本概念予以简单介绍。

1. 位点

位点是构成汉字的最小单位。例如，在中文电脑中，构成汉字的点阵数有 16×16，24×24，32×32，48×48，64×64，96×96，128×128，256×256，等等。在一个 x–y 坐标图中，每一位点都是 x–y 坐标的一个交点。

2. 笔画

笔画是由位点构成的。笔画包括两个方面：一是笔形，即笔画形体；二是画数，即笔画数量，亦称笔数。在形成字体结构时，二者有如下不同的表现：

(1) 单笔结构

笔数为 1，笔形有：

横一　竖丨　撇丿　捺㇏　弯㇆　拐㇄　六种。按方向性(笔向) 区别，可分为以下两类：

①单向笔画：一丨丿㇏

②复向笔画：㇆㇄

(2) 复笔结构：

笔数为 2、3 或多笔，笔形按位置关系和接触关系分类如下：

①独立结构

i。 连接结构

2 笔：丁厂冂匚亠卜卩冂𠂉亻人勹冖讠一𠂇工𠂆刀凵𠃌厶

3 笔：工兀千上下万口个夕久𠂉彐阝尸巳弓山𠂤

多笔：扌卜凸凹臣丐勿日月欠爿占

ii. 交叉结构

2笔：十ナ乂七弋犭乂 力九匕乜

3笔：艹乇扌乇大丈扌女犭

多笔：丰耒车韦事秉中聿

②离散结构

i. 左右离散结构

2笔：儿八丷

3笔：川⺌巛

多笔：𠆢 州

ii. 上下离散结构

2笔：二冫

3笔：三彡氵习

多笔：兰

3. 部首

部首是组成汉字的不宜分拆的基本结构单位。所谓部，就是按字形把汉字分类为若干个部类，每一部类中的汉字都含有各字所共有的一个表意结构。所谓首，就是每个部类各立一个类首，排列在该部类各字的前头。我们称这个带队的表意汉字叫部首。

部首的历史很长。作为部首的某些汉字后来逐步演变成全无字义的纯粹记号（例如，手部的"扌"，水部的"氵"等）。尽管如此，部首的意义及部首在汉字分类应用上的功能一直保持不变。

由于汉字有形、音、义属性，因此历史上曾出现按形、音、义分部首的标准。直到1963年《辞海》在"部首查字法查字说明"中才有比较明确的两条规定，如下：

(1) 依据字形定部，一般采用字的上、下、左、右、外等部位作部首，其次是中坐和左上角，按照以上七种部位都无法确定部首的，列入余类；

(2) 部首共250个，按笔画排列，同笔画的按一丨丿㇏㇇五种笔形顺序排列，另有余类，排在最后。

虽然这种依据字形定部是对汉字分类归部标准的一大改进，但由于汉字结构复杂多变，因此仍不免出现歧义情况，例如，对于“二元未”三字，可归入“一”部，又可归入“二”部，其中“元”字又可分解为“一二兀儿”这四种结构，因此可归选的部首有“一”部、“二”部、“兀”部和“儿”部。

鉴于上述原因，从实用的角度出发，一个切实可行的方案是：把几百个部首简化为大约 100 个字首（即字体起笔处的那个通用字根）作为汉字分类检索的新标准。实质上，字首就是部首。

4. 字根

字根是由笔画组成的，是在通用的部首基础上，根据这些部首在汉字选字范围内出现的频度而选择出来的。

根据对 4356 个常用汉字分解统计，字根的出现频度依次如下：

口 732，木 496，氵 433，人 430，扌 360，土 339，日 317，冂 312，十 298，艹 284，亻 278，丷 229，勹 220，八 217，又 209，大 201，忄 182，乂 168，夕 167，⺕ 165，阝 160，月 159，小 155，厶 152，刀 149，二 145，贝 141，讠 137，女 131，纟 130，宀 129，二 127，田 121，⺈ 112，卜 105，冖 105，丬 103，山 102，彳 101，弋 99，金 98，匕 97，三 93，廿 93，匚 90，王 88，火 87，文 85，禾 80，尸 75，虫 75，目 73，厂 72，广 68，礻 66，灬 63，子 62，罒 58，白 56，车 55，犭 55，米 54，疒 53，石 53，饣 52，巾 51，口 51，工 50，足 47，止 47，⺮ 44，疒 43，弓 42，习 41，马 41，牛 41，衤 40，门 39，羊 38，酉 37，幺 35，臼 34，雨 34，刂 32，方 30，户 29，穴 28，矢 28，……(以下从略)

中文电脑中选用的字根个数，以控制在 100 个左右为宜。这样，一方面，易学，易记，易用；另一方面，可减少击键次数。

按照字根结构的形态特征和笔画特征，即字根内笔画与笔画之间的关系，可把字根分为以下八类：

(1) 单笔字根:

只一笔就形成一个独立结构的字根。例如，一丨丿㇏㇇㇄

(2) 散笔字根

只有散笔关系的字根。例如，

二三八𠂊丷⺌冫氵彐

(3) 连笔字根

只有连笔关系的字根。例如，

工厂匚歹卜止冂月口足日月四𠂉亻白冖夕饣勹工广宀尸巳弓刀阝匕凵山厶

(4) 交笔字根

只有交笔关系的字根。例如，

十艹扌七车女牛乂犭力九又

(5) 散连笔字根

具有散、连两种关系的字根。例如，

贝彳𠂉疒门穴讠礻衤之纟

(6) 散交笔字根

具有散、交两种关系的字根。例如，

忄小米

(7) 连交笔字根

具有连、交两种关系的字根。例如，

耳王土西廿木大田巾虫角毛禾火子

(8) 散连交笔字根

具有散、连、交三种关系的字根。例如，

雨酉舟鱼

字根原来都是汉字，发展到现在，有些字根已不是汉字，而成为一种纯粹的记号。例如，“㇆𠂊冫氵讠礻衤冖扌犭忄”等字根不能把它们原样算作汉字，必须把它们改写为“乙爪冰水言示衣幂手犬心”才是汉字。又例如“丨丿㇏㇇㇄丷匚冂口𠂉亻亠⺈勹龴阝凵𠂇⺂乂”等字根已失去意义，纯粹是一种记号而已。

5. 汉字

汉字（单字）是由字根构成的。汉字字体内各字根相互间的表现形式，是汉字结构的最重要特征。它包含两个方面：一是字式，即字体结构内各字根间体现在接触关系上的一种结构方式；二是字型，即字体结构内各字根相互间的一种结构类型。

(1) 字式

汉字的字式有如下四种：

①单式　字体内部构件不可分拆，只是一个单独存在的原始字根。例如，日月木火鱼雨

②散式　字体内字根与字根间只有离散关系。例如，“盟”字的“日月皿”三者间是离散关系。

③连式　字体内一单笔字根与一复笔字根相连。例如，

i. 单复相连：天（一大）正下千禾疋耳(一耳)

ii. 复单相连：丕（不一）韭业少尺（尸丶）久（夕丶）

④交式　字体内字根笔画互相交叉。例如，未（二小）末(一木）兆（𠂉儿）林（林乂）臾(臼人)

(2) 字型

汉字的字型有如下四种：

①独体型

独体型汉字就是独体字。图式为：□。例如，

i. 单式独立体型：三石鱼米山（单根结构）

ii. 连式独体型：天下千少尺（复根连笔结构）

iii. 交式独体型：夫丈事秉半坐（复根交笔结构）

②左右型　字体内的左字根与右字根间有一定间隙的散式结构。图式为：□□。例如，相鸠邢炳锉

③上下型　字体内的上字根与下字根间有一定间隙的散式构型。图式为：□
□。例如，杏英蚕杂岩

④内外型

字体内一个内字根被一个外字根包围着的散式构型。例如，

囚困国　　图式为： 

网凶区　　图式分别为： 

这历司　　图式分别为： 

由此可见，独体型包括单式、连式、交式三种，而左右型、上下型、内外型则全属散式一种。

图 2.2 概括了汉字的各种类型。

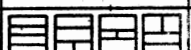
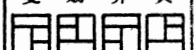
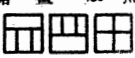
	一根字	二根字	三根字	四根字	合计
独体型	 田月由中天母 等				1 种
左右型		 佃促肌	 湘 沼 邵 概 海 动  娜 墉 都	 州 褥 湖 鹕  碗 鹋 溜 劉  漫 鹋 疑	15 种
上下型（内外型）		 宙男卷笑贯	 曼 茄 盟 霉 藕 墅 粪 宛	 蔓 荔 葬 巽  露 鹭 霰 照  樊 琵	15 种
合计	1 种	2 种	6 种	22 种	31 种

图 2.2　汉字字型表

其中，由于内外型的汉字构型为数较少，故作上下型处理较为简便适宜。

2.1.3　汉字字音

汉字字音是汉字的又一重要属性。研究汉字字音，有助于为

中文电脑在字音信息处理方面提供依据。例如，

(1) 在汉字输入方法中，有很多键盘输入编码方案是基于字音特性的，比如，拼音输入法、注音输入法等。尤其是，汉字语音输入方式，更是与汉字字音密切相关。

(2) 汉字编码字符集中有些汉字是按字音规律排列的。比如，基本集中的第一级汉字是按汉语拼音字母顺序排列的。

汉字的字音，源自汉语的语音。语音来自人的发音器官。发音的部位叫做音位。来自各个不同音位的语音最小单位叫做音素。

我国的汉语拼音方案和台湾省的汉语注音方案是汉字字音和汉语语音的研究成果。

值得注意的是，汉字同音现象十分严重。在常用和次常用汉字中，共有400多种单音，一音多达一百二、三十个字。虽然在社会实际应用中，由于语言环境和上下文等多种客观因素，同音现象所产生的矛盾并不突出，但在中文电脑中，却是一个困难的问题。例如，汉语拼音输入方法需要在一音多字的情况下选择汉字，致使汉字输入速度很难提高。此外，对于那些不知其读音的汉字是无法用汉语拼音输入汉字的；而且，各种不同方言地区的人也常常会拼错字音。

2.1.4 汉字字义

语言的表现形式是“音”，其潜在内容是“义”。文字的表现形式是“形”，其潜在内容是“音”和“义”。

每个汉字都有几个意义，一般常有2–5个意义，有的多达6–9个意义，甚至有少数汉字多达十几个意义。汉字在不同的专业领域中有不同的意义，差异很大。

中文语言自动处理、机器翻译、自然语言理解等方面的研究，要涉及到汉字字义，尤其是解决语义歧义性问题。

2.2 汉字编码字符集

在中文电脑中，究竟选用多少个汉字，选用哪些汉字，最终体现在汉字编码字符集上。本节将首先介绍计算机的各种编码字符集，然后分别介绍我国和台湾省公布的汉字编码字符集。

2.2.1 计算机系统的编码字符集

为了确保中文信息处理与西文信息处理的一致性和兼容性，中文信息系统采用的汉字编码字符集，一般都是在信息处理交换用的字符集基础上扩充而来的。

计算机系统使用的信息处理交换用编码的字符集共有下列三大代码体系：

1. 七位编码字符集

ISO 646 代码、ASCII 码、GB1988 代码均属于七位编码字符集代码体系。

ISO 646 是国际标准化组织(International Standards Organization)制定的《信息处理交换用的七位编码字符集》国际标准。计算机系统上使用的七位编码字符集都源于国际标准 ISO 646。该字符集规定了信息处理交换用的 128 个字符，每个字符都是七位编码，其中包括 94 个图形字符、32 个控制字符、1 个间隔字符（SP）和 1 个抹消字符（DEL）。

ASCII 码是美国信息交换标准码(American Standard Code for Information Interchange)，它与国际标准 ISO 646 兼容。ASCII 字符集广泛应用于微型计算机系统。ASCII 字符表如下：

字符	十进制	十六进制	注　解
NUL	0	00	Null
SOH	1	01	Start of Heading
STX	2	02	Start of Text
ETX	3	03	End of Text
EOT	4	04	End of Transmission
ENQ	5	05	Enquiry
ACK	6	06	Acknowledge
BEL	7	07	Bell
BS	8	08	Backspace
SH	9	09	Horizontal Tabulation
LF	10	0A	Line Feed
VT	11	0B	Vertical Tabulation
FF	12	0C	Form Feed
CR	13	0D	Carriage Return
SO	14	0E	Shift Out
SI	15	0F	Shift In
DLE	16	10	Data Link Escape
DC1	17	11	Device Control 1
DC2	18	12	Device Control 2
DC3	19	13	Device Control 3
DC4	20	14	Device Control 4
NAK	21	15	Negative Acknowledge
SYN	22	16	Synchronous Idle
ETB	23	17	End of Transmission Block
CAN	24	18	Cancel
EM	25	19	End of Medium

续

字符	十进制	十六进制	注 解
SUB	26	1A	Substitute
ESC	27	1B	Escape
FS	28	1C	File Separator
GS	29	1D	Group Separator
RS	30	1E	Record Separator
US	31	1F	Unit Separator
SP	32	20	Space
!	33	21	
"	34	22	
#	35	23	
$	36	24	
%	37	25	
&	38	26	
'	39	27	
(	40	28	
)	41	29	
*	42	2A	
+	43	2B	
,	44	2C	
−	45	2D	
.	46	2E	
/	47	2F	
0	48	30	
1	49	31	
2	50	32	
3	51	33	
4	52	34	

续

字符	十进制	十六进制	注　解
5	53	35	
6	54	36	
7	55	37	
8	56	38	
9	57	39	
:	58	3A	
;	59	3B	
<	60	3C	
=	61	3D	
>	62	3E	
?	63	3F	
@	64	40	
A	65	41	
B	66	42	
C	67	43	
D	68	44	
E	69	45	
F	70	46	
G	71	47	
H	72	48	
I	73	49	
J	74	4A	
K	75	4B	
L	76	4C	
M	77	4D	
N	78	4E	
O	79	4F	

续

字符	十进制	十六进制	注　解
P	80	50	
Q	81	51	
R	82	52	
S	83	53	
T	84	54	
U	85	55	
V	86	56	
W	87	57	
X	88	58	
Y	89	59	
Z	90	5A	
[	91	5B	
\	92	5C	
]	93	5D	
^	94	5E	
—	95	5F	
`	96	60	
a	97	61	
b	98	62	
c	99	63	
d	100	64	
e	101	65	
f	102	66	
g	103	67	
h	104	68	
i	105	69	
j	106	6A	

续

字符	十进制	十六进制	注 解
k	107	6B	
l	108	6C	
m	109	6D	
n	110	6E	
o	111	6F	
p	112	70	
q	113	71	
r	114	72	
s	115	73	
t	116	74	
u	117	75	
v	118	76	
w	119	77	
x	120	78	
y	121	79	
z	122	7A	
{	123	7B	
|	124	7C	
}	125	7D	
~	126	7E	
DEL	127	7F	Delete

其中，码值为十进制 32–126 的 ASCII 字符为图形字符，亦称为可打印字符或键盘字符；其余 ASCII 字符为非键盘字符，它们有两种含义，有时当控制字符用，有时当图形字符用，取决于它们所处的环境。例如，码值为十进制 24 的 ASCII 字符 CAN 有时可显示一个向上箭头，码值为十进制 27 的 ASCII 字符 ESC 有时可显示一个向左箭头。

GB1988《信息处理交换用的七位编码字符集》是我国根据国际标准 ISO 646 制定的国家标准。

2. 八位编码字符集

EBCDIC 码，ISO 4873 代码、ASCII 扩充字符集代码均属于八位编码字符集代码体系。

EBCDIC 是扩充的二进制编码的十进制交换码(Extended Binary-Coded Decimal Interchange Code)，简称为扩充的二-十进制交换码，是一种主要用在 IBM 公司设备上的八位字符代码。计算系统上使用的八位编码字符集大多源于 EBCDIC。该代码提供了 256 种不同的位组合格式，其中包括 191 个图形字符、64 个控制字符和 1 个间隔字符（SP）例如:

EBCDIC 字符	EBCDIC 码
1	11110001
2	11110010
3	11110011
A	11000001
:	:

ISO 4873 是国际标准化组织在 ISO 646 七位编码字符集基础上扩充而成，码值为十六进制 00-FF。

ASCII 扩充字符集是在 ASCII 字符集基础上扩充而成的。它的十进制码值为 0-255，十六进制码值为 00-FF，是一种八位代码。

3. 多八位编码字符集

显然，七位编码字符集只规定了 128 个字符及其编码表示，八位编码字符集只规定了 256 个字符及其编码表示。它们只能满足西文信息处理的需要，远远不能满足中文信息处理的需要。为了满足中文信息处理的需要，并保证汉字编码字符集与七位编码字符集或八位编码字符集兼容，常采用双字节七位编码或双字节八位编码来对汉字编码。为此，国际标准化组织制定了国际标准

ISO 2022《七位与八位编码字符集的扩充方法》。

我国根据 ISO 2022 制定了国家标准 GB 2311《信息交换用七位编码字符集的扩充方法》。它是以七位编码字符集为基础进行代码扩充的。国家标准 GB 2312《信息交换用汉字编码字符集——基本集》是根据 2311 的代码扩充方法制定的汉字交换码标准。由于它与 GB1988 兼容，因此使计算机系统能方便地增加汉字处理功能和进行中文信息交换。由于单字节七位编码字符集只有 94 个图形字符，因此双字节七位编码字符集可容纳 94×94＝8836 个汉字及其它图形字符（见图 2.3）。

同样，亦可用双字节八位编码来对汉字编码。若用双字节 EBCDIC 码来表示汉字编码，则由于单字节 EBCDIC 码只有 190 个图形字符，因此双字节 EBCDIC 码可表示 190×190＝36100 个汉字及其它图形字符。若用双字节 ISO 4873 代码来表示汉字编码，则可表示 35344–36484 个汉字及它图形字符。

这种扩充方法，固然可以解决一部分问题，但造成系统开销大、效率低，尤其是对东方语言，例如，中文、日文、朝鲜文等。为了在计算机系统中使世界各种语言文字信息处理兼容。国际标准化组织正在积极组织制定多八位编码字符集（ISO 10646）。该标准有可能为多文种、大字符集信息处理技术建立新的平面，这一战略地位受到各国家、各大公司的高度重视，争论也异常激烈。中文、日文、朝鲜文在多八位编码字符集(Multi–Octet Coded Character Set)中的安排问题是研究的重点，而象形表意文字（汉字）在基本多文种平面的编码则是争论的焦点（详见 4.3 节）。

2.2.2 汉字编码字符集——基本集

GB2312–80《信息交换用汉字编码字符集——基本集》共收录汉字等图形字符 7445 个，其中包括汉字 6763 个及其它图形字

符 682 个。

汉字分为两级：第一级汉字 3755 个，由于第一级汉字是常用字，一般都知道读音，故按汉语拼音字母顺序排列（多音字取它的常用字），同音字再以起笔笔画顺序为序，笔画顺序为：横（一）、竖（丨）、撇（丿）、点（丶）、折（乙、𠃌、𠃋、亅等），若起笔笔画相同，则按第二笔画排序，依次类推；第二级汉字 3008 个，由于第二级汉字大部分较生僻，不易掌握读音，所以按部首顺序排列，部首次序及同部首字按笔画数排列，笔画数相同的部首和字以笔画顺序（同第一级汉字中规定的笔画顺序）为序。所含部首及其排列顺序如下：

一丨丿丶𠃌厂匚卜刂冂亻人入八勹亠冫冖讠卩阝刍夕力又口厶土扌艹 口巾山彳彡犭夕女饣广忄夭门氵片礻宀辶彐尸弓女子马幺巛王玉韦木扌歹车戈瓦日贝见牜气攵月欠殳文立方火灬户礻心钅石目田四皿钅矢禾白乌疒穴礻来耳页虍虫缶灿舟衣羊米艮羽糸麦走赤豆酉豕足龰角言雨齿电隹金鱼革骨鬼食髟麻鹿黑鼠鼻

其它图形字符包括：

(1) 一般符号 202 个，包括间隔符、标点符号、运算符、单位符号和制表符；

(2) 序号 60 个，其中 1–20 二十个，(1) – (20) 二十个，①–⑩十个，(一) – (十) 十个；

(3) 数字 22 个，其中 0–9 十个，罗马数字十二个；

(4) 英文字母 52 个，其中大小写各 26 个；

(5) 日文假名 169 个，其中平假名 83，片假名 86 个；

(6) 希腊字母 48 个，其中大小写各 24 个；

(7) 俄文字母 66 个，其中大小写各 33 个；

(8) 汉语拼音字母 26 个；

(9) 汉语注音符号 37 个。

基本集中汉字的选择是按使用频度来决定的。根据我国 1975 年的查频统计，基本集中的 6763 个汉字的使用覆盖率可达

99.9%左右，但是它们的使用频度却相差很大。根据20年代到70年代的各种汉字使用频度统计，3000–4000个常用字覆盖率达99.9%左右。因此，实际上只要具备这三、四千个字就能大体上满足一般应用的需要。因此，为了便于使用，便于设备的分档制造，有必要把这三、四千字作为一级常用字区分开。在综合考虑了汉字使用频度的高低、构词能力强弱、实际用途的大小等情况后，选择了一些具有代表性的3755个汉字作为一级常用字，其余3008个汉字作为二级次常用字。

GB2312图形字符代码表规定了汉字交换码标准（见图2.3）。每个图形字符的汉字交换码均用两个字节表示，每个字节为七位二进制码。为了保持同现行计算机系统所采用的标准信息处理交换码的兼容性，每个字节的七位码在0100001–1111110之间，这正是ASCII字符中可打印字符（字母、数字、特殊字符）的编码（21–7E）。GB2312规定的汉字交换码亦称为国标码，通常用十六进制数表示。代码表纵向分为94个区，由第一字节标识，横向将每个区分成94个位置，由第二字节标识。因此，代码表中的汉字或非汉字图形字符亦可用它在代码表中所在位置的区号和位号来标识，我们称这个区号和位号的编码为区位码。实际上，区位码是与国标码一一对应的，区位码就是十进数表示的国标码。显然，代码表最多可收8836个（即94×94个）图形字符，国标码为2121–7E7E，区位码为0101–9494；其中，第16–55区是一级汉字，国标码为3021–577E，区位码为1601–5594；第56–87区是二级汉字，国标码为5821–777E，区位码为5601–8794；第1–9区是其它图形字符，国标码为2121–297F，区位码为0101–09FE；第10–15区和第88–94区是空白位置，留作备用。区位码转换为国标码的方法是：把区码与位码分别转换为十六进制码，合并后加2020。

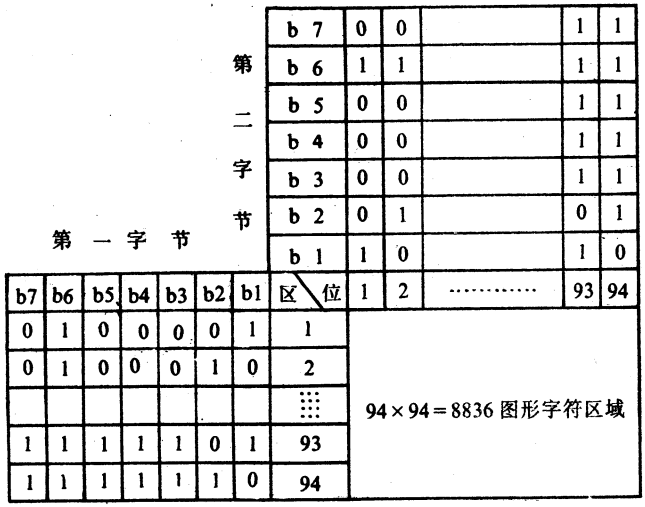


图 2.3 GB2312 图形字符代码表示意图

自从公布了这一汉字交换码标准之后，它便成为汉字的一项重要属性。GB2312 目前已广泛用于中文电脑软硬件的设计与应中。例如，汉字字库的设计目前均以 GB2312 为标准。汉字整字输入键盘盘面文字的选择、汉字输入码的转换、汉字输出设备的汉字地址码，也都遵照 GB2312 来设计。还有，中文操作系统、中文多用户系统、中文网络系统、中文程序语言、中文数据库管理系统、中文通用应用软件、中文应用系统的设计，也都是以 GB2312 为基础的。此外，由于 GB2312 是中文信息处理技术的基础标准，因此许多标准都与它密切相关，例如，汉字点阵字形标准、磁盘和磁带等格式标准、各种汉字输入输出设备标准的制定，都贯彻执行 GB2312 标准。

2.2.3 汉字编码字符——辅助集

从 GB2312 的使用情况来看，基本集已基本满足绝大部分用

户的使用要求，但对于某些特殊用户，例如，香港等目前尚未推广简化字的地区、古汉语研究、古籍翻印、图书管理、大城市户籍管理等，则感到只有基本集的六千多个汉字还不够用。因此，又在基本集的基础上公布了 GB7589–87《信息交换用汉字编码字符集第二辅助集》和 GB7590–87《信息交换用汉字编码字符集第四辅助集》，从五万余字中又筛选了一万六千余字，作为基本集的扩充。

辅助集的编码结构与基本集相同（仍是 94×94），即辅助集中每个汉字也是用两个七位编码表示。第 1–4 区为保留区，汉字排列在第 5–94 区。如图 2.4 所示。

区号 \ 位号	1 2 3 4 5 …… 93 94
1 2 3 4	保留标准区
5 6 : : 93 94	图形字符区

图 2.4　辅助集的编码结构

根据汉字的查频统计，基本集以外汉字的使用频度都很低（万分之一以下），因此，辅助集中汉字的选择不宜采用阶梯式查频统计的方法，而是根据一些实际使用的典型辞典，参照各汉字在这些辞典中的字义项数（构词能力）和实际用处等进行选择。

总之，辅助集的编制要综合考虑汉字的使用频度、构词能力和实际用途。

汉字有简体字、繁体字、异体字之分。可按简体字、繁体字和异体字的顺序来确定同一汉字在辅助集中的位置。基本集 G_0、第二辅助集 G_2、第四辅助集 G_4 等偶数编号的字符集为规范字集（即简体字集），而 G_1、G_3、G_5 等奇数编号的字符集为对应的繁体字集（包括异体字）。例如，G_1 集中的每个繁体字或异体字都与 G_0 集中的简体字一一对应。也就是说，同一汉字的简体字和繁体字的区位号是相同的，只是集合号不同。比如，“体”字在 G_0 集中区位号为 4469，则它所对应的繁体字“體”在 G_1 集中的区位号也为 4469；“异”字在 G_0 集中区位号为 5076，则它所对应的异体字“異”在 G_1 集中在区位号也为 5076，如果 G_0 集中的汉字无对应的繁体字和异体字，则它在 G_1 集中也无对应的汉字。

由于辅助集中的汉字绝大部分都是生僻字。为了便于检索。汉字按部首和笔画数排序，其方法同基本集。

2.2.4 扩展集

由于基本集无法满足某些场合下的特殊需要,因此有些中文系统在基本集的基础上扩充了若干扩展集。例如，

(1) 汉字辅助集；

(2) 繁体字；

(3) 少数民族文字；

(4) 多国文字(日文、俄文、德文、法文、葡萄牙文等)；

(5) 古文字；

(6) 一些有用的符号，例如，上标符，下标符等，这对于显示和打印数学公式、化学结构式是很有用的，比如，2^{63}，H_2O 等。

(7) 基本集中的汉字不是以覆盖全部四码电报所表示的汉

字，这样，就难以满足通信的需要，有的中文系统用 88—94 区支持通信子集，增补了约七百个汉字；

(8) 满足联机仿真中与某种主机联机时特殊字符的需要，比如，与 HP 主机联机就有这种要求。

2.2.5 通用汉字标准交换码

台湾省公布的《通用汉字标准交换码 CNS11643》字集包括常用汉字 5401 个，次常用汉字 7650 个，共计 13051 个汉字，字序按部首和笔画数排列。

2.3 汉字代码

汉字代码包括汉字输入码，汉字内部码、汉字地址码、汉字字形码、汉字交换码、汉字控制功能码等。

各种汉字代码之间的关系，以及它们在中文信息处理过程中的位置，已在前面的图 2.1 中描述。由图可知，从汉字代码的角度看，中文信息处理系统就是一个进行各种汉字代码转换的系统。

由于各个中文系统的结构和功能不同，因此各个中文系统不都同时具有上述的各种汉字代码，而且汉字代码在中文系统中的位置也是因系统而异的。

本节将逐一介绍各种汉字代码。

2.3.1 汉字交换码

汉字交换码是一种用于中文信息处理系统之间或者通信系统之间进行信息交换的汉字代码。我国的 GB2312《信息交换用汉字编码字符——基本集》制定了汉字交换码的标准。台湾省的《通用汉字标准交换码》亦是汉字交换码。

汉字交换码的制定，应主要考虑以下两个问题：

(1) 中西文信息处理交换码的兼容性;

(2) 汉字选择的实用性，通用性和编码效率。

2.3.2 汉字输入码

汉字输入码主要用于通过键盘输入汉字。用来代表某一汉字的一组键盘符号，称为这个汉字的输入码。为了建立友好的用户界面，输入码规则必须简单清晰、直观易学、容易记忆、操作方便、码位短、输入速度快，重码少，符合人体工学，既适合初学者，又能满足专业输入者的要求，便于盲打。为达到这一目标，人们根据汉字的各种属性提出了上千种汉字输入编码方案。究竟哪种编码方案好，还有待实践检验。一般说来，由于用户不同，用途有别，故对采用什么编码方案不能强求统一，不过，应当制定一个评测汉字输入编码的规则，对各种编码方案测试，并公布各项测试结果，以供用户和研制单位选择汉字输入方法作为参考依据。如果能通过对各种汉字输入编码方案的评测，优选出几种最佳编码方案，这对中文电脑的推广应用无疑是有积极意义的。

2.3.3 汉字内部码

汉字内部码亦称为汉字内码或汉字机内码。

计算机处理汉字，实际上处理的是汉字代码。供计算机内部加工处理用的汉字代码称为汉字内码。汉字内码可分为存储码，运算码和传输码三种。存储码用于存储信息，既可用作汉字的机内表示，又可用作在磁记录媒体（磁盘、磁带、磁鼓）中的表示。运算码用于参与各种操作运算。传输码用于计算之间的通信(亦称通信码)，或用于在计算机内部各部件之间传送汉字信息(比如，传送给显示器或打印机的内部传输码)。最好能把这三种汉字内码统一起来，以减少代码转换，提高汉字处理效率。

1. 汉字内码的设计原理

在计算机内部如何表示内码是中文信息处理的重要问题。汉

字内码的设计往往与具体的系统及使用要求密切相关，没有统一的格式。因此，目前汉字内码的形式有很多种：二字节汉字内码，三字节汉字内码，四字节汉字内码，带引导符的汉字内码，带括号的汉字内码，等等。汉字内码应采用哪种形式，取决于具体的计算机环境。一般应考虑以下几点原则：

(1) 与西文字符编码(ASCII 码、EBCDIC 码等)不发生冲突，容易区分汉字与西文字符；

(2) 用尽可能少的字节数表示尽可能多的汉字；

(3) 与标准交换码兼容，即与交换码有尽可能简单明确的对应关系；

(4) 便于操作运算，而且当操作运算时不容易产生二义性和不确定性；

(5) 便于纳入各种程序语言的字符类型或字符串类型。

2. 汉字内码的表示形式

下面，介绍目前常用的几种汉字内码形式。

(1) 二字节汉字内码

①带标识位的二字节汉字内码

这种汉字内码用两个八位字节表示。它在双字节七位编码基础上，第一个字节和第二个字节高位别为 1 和 1，或 1 和 0。或 0 和 1。

对于两个字节高位均为 1 的情况，容易区分汉字和西文字符；但是汉字代码易产生二义性，这是因为任何两个高位为 1 的字节均可以组成一个汉字内码，在对汉字进行编辑时极易出错。例如，用替换命令将汉字串“交换”中的“换”字改为“还”字，结果有时是“交够”而不是“交还”。原因何在？

汉字串“交换”基于 GB2312 的汉字内码为 BDBB 和 BBBB，汉字“还”基于 GB2312 的汉字内码为 BBB9。当使用批量替换命令时，编辑程序先对汉字串“交换”的内码进行扫描。当扫描到第一个 BB 时，便开始替换操作，即用 BBB9 替换了 BD

BB BB BB 中前两个 BB BB，结果是 BD BB B9 BB，这正是汉字串“交够”基于 GB2312 的汉字内码。

对于第一个字节高位为 1 第二字节高位为 0 的情况，系统将高位为 1 的字节与紧随其后的那个高位为 0 的字节合起来解释为汉字内码。这种格式可以避免汉字代码的二义性，但对单个西文字符进行编辑时，也会出现二义性错误。

对于第一字节高位为 0 第二个字节高位为 1 的情况，系统将高位为 1 的字节与紧挨着前一个高位为 0 的字节合起来解释为汉字内码。这种格式需要逆扫描，虽然可以避免汉字代码的二义性，但对单个西文字符进行编辑时容易产生二义性。

这种汉字内码仅用两个字节表示一个汉字，汉字处理和传输速度快；而且容易纳入程序语言的字符类型或字符串类型，可直接参与字符串的操作。尤其是，汉字显示或打印时通常一个汉字恰好占据两个西文字符的位置，从而保持了汉字存储与输出格式的一致性，并使汉字输出与西文字符输出有简单的对应关系，因而汉字处理较为容易，而且效率高。但是，这种汉字内码容易产生上述的二义性，增加了软件汉化工作量；此外，在计算机之间通信或在计算机内部各部件之间传送汉字信息时，往往不允许高位 1 通过，例如，有些终端只接受七位码，而高位 1 用作奇偶校验位，在这种情况下，这种汉字内码就不适用了。

目前国内绝大部分微型计算机上的中文系统都采用以基本集汉字交换码（即国标码）两个字节高位均加 1 形成的汉字内码，因此汉字内码标准较为统一。由于基本集的国标码为 2121−7E7E，因此相应的汉字内码为 A1A1−FEFE。同样，，由于一级汉字的国标码为 3021−5779，二级汉字的国标码为 5821−777E，其它图形字符的国标码为 2121−296F，因此相应一级汉字的内码为 B0A1− D7F9，二级汉字的内码为 D8A1−F7FE，其它图形字符的内码为 A1A1−A9EF。国标码转换为内码的方法是：国标码加 8080。

然而，这种汉字内码有着致命的缺点：由于现有的中文系统过分强调汉字内码码值的连续性，致使汉字内码的码值(A1-FE 占据了 IBM 扩充字符集中的制表符、背景符等图形符号的码值位置，因此很多西文软件在中文系统中无法使用有关的命令和语句，而且出现许多奇怪的汉字，例如，——变为哪字(哪字的内码为 C4C4)，＝＝变为屯字（屯字的内码为 CDCD)，  变为鞍字(鞍字的内码为 B0B0),  变为  馁(馁字的内码为 C4D9)，等等。特别是，在当前应用软件窗口化的趋势下，西文软件的窗口汉化是相当困难的。因此，建议将这种汉字内码改造为越过有关的制表符和背景符的码值位置。为此，基于基本集的汉字内码不必强求两个字节的高位均为 1，可设置一级汉字的两个字节的高位仍均为 1，而二级汉字的第一字节高位为 1，第二字节高位为 0。这样，既可以解决越过英文制表符和背景符的位置的问题，又可以使代码区足以容纳基本集 7445 个图形字符，从而提高中西文兼容性，使应用软件汉化工作量大大减少。

②双字节 EBCDIC 码

用双字节 EBCDIC 码表示汉字内码，要避开 EBCDIC 码中的图形字符。也就是说，利用 EBCDIC 码中空闲的码位来编制汉字内码。在实际使用中，由于 EBCDIC 码中空闲码位常用作特殊定义的字符，因此一般不要轻易占用。

(2) 三字节汉字内码

①带标识码的三字节汉字内码

这种汉字内码的形式如图 2.5 所示。

标识码	第一字节	第二字节

图 2.5 带标识码的三字节汉字内码

当中文系统扫描汉字与西文字符混合的字符串时，遇到标识

码后，就把紧随它的两个字节解释为一个汉字，否则仍把单个字节作为西文字符处理。

这种汉字内码既适用于ASCⅡ码等七位编码体系，又适用EBCDIC码等八位编码体系。

这种汉字内码在程序语言语法的影响方面和在汉字输出转换时，对汉字和西文字符之间的识别能力要优于带引导码的汉字内码，并且给实现变量名、文件名的汉化的带来方便。但一个汉字用三字节表示，占存储空间大，输入输出码制的转换开销也较大，而且输出格式编排中西文对应关系不方便。此外在汉字检索识别上，需要解决标识码与ASCⅡ或EBCDIC码等西文字符编码的冲突问题，例如，检索一个或两个字节的字符信息无法区别它是一个汉字还是两个西文字符。更为严重的是，标识码选择本身就是一件难事，该标识码至少在原西文系统中现在和将来都不会用到，而且在其它系统中也不会用到，否则很难最大限度地实现中西文系统兼容。

②三字节字母数字码

这种汉字内码利用大写字母A–Z，小写字母a–z和数字0–9共62个字符来给汉字编码。它有许多种排列组合编码格式，下面列举几种；

小写字母、大写字母、数字三字节码

大写字母、小写字母、数字三字节码

小写字母、小写字母、数字三字节码

大写字母、大写字母、数字三字节码

大写字母、数字、大写字母数字三字节码

三字节大写字母码

由于有的小型机甚至有些大型机在连接终端设备时，系统只允许从终端进入字母和数字，因此，三字节字母数字码是有实际意义的。但这种内码用三个字节才能表示一个汉字，开销很大。

③王安电脑三字节汉字内码

王安电脑三字节汉字内码仅适用于王安系统。它的各字节内容定义如下:

第一字节，选用下列七个图形字符之一:

十六进制代码	5B	5D	60	7B	3A	3D	7E
图形字符	[	]	·	{	:	}	~

第二字节和第三字节，选用大写字母 A–Z。小写字母 a–z。数字 0–9 或第一字节定义的七个图形字符（共计 69 个图形字符）之一。

(3) 四字节汉字内码

① 带标识码的四字节汉字内码

这种汉字内码的形式是一个标识码后跟三个字节的图形字符。例如，CCDOS 的通信管理模块用这种汉字内码作传输码，它的编码格式是：标识码为 7EH（图形字符～），第二字节为数字 0–9 或大写字母 A–F，第三字节和第四字节为数字 0–9 或大写字母 A–V。又例如，天龙四字节码，标识码为 3AH（图形字符:）。

这种汉字内码由于只用于七位码，故在一般的微型计算机多用户系统中用作传输码，可顺利地通过终端驱动模块。但这种内码用四个字节才能表示一个汉字，开销太大。

② 四字节字母数字码

这种汉字内码利用大写字母 A–Z，小写字母 a–z 和数字 0–9 共 62 个字符来给汉字编码。例如，各个计算机系统对 ASCII 字符集中 94 个图形字符的约束情况不同，有的系统，例如 Cyber 机，就连小写字符都不受理，但对于大写字母 A–Z 和数字 0–9，各种计算机系统均采用。因此，在某种场合下，由大写字母和数字组成的四字节汉字内码是有实际意义的。

四字节字母数字码的优缺点与三字节字母数字码大致相同。

③四字节字母数字及其它字符码

这种汉字内码利用大写字母 A–Z，小写字母 a–z、数字 0–9 及其它图形字符来给汉字编码。它有许多种排列组合编码格式，下面列举几种：

数字、字母、其它字符、数字四字节码

数字、字母、其它字符、字母四字节码

数字、字母、其它字符、其它字符四字节码

在上述四字节编码格式中，对字节中的字母及其它字符的选择，可根据具体计算机系统环境而定。不一定选取 A–Z 或 a–z 中的全部字母，32 个其它字符也可只选其中的一部分，还可全部用数字码来表示。

某些对图形字符有特殊要求的系统，可采用此类汉字内码结构。这种结构存在字节编码字符选择难、存储空间大等问题，一般不宜采用。

(4) 带引导码的汉字内码

这种汉字内码分别用汉字引导码（例如，移出字符 SO）和西文引导码（例如，移入字符 SI）来标识汉字串和西文字符串的开始，用以在汉字和西文字符混合的字符串中区分汉字和西文字符。如图 2.6 所示。

汉字引导码	汉字串	西文引导码	西文字符串

图 2.6　带引导码的汉字内码

这种汉字内码既适用于 ASCII 码等七位编码体系，又适用于 EBCDIC 码等八位编码体系。

这种汉字内码可直接采用汉字交换码（例如，国标码或区位码）来表示汉字，而且用引导码区分汉字和西文字符，使得汉字内码与 ASCII 或 EBCDIC 码等西文字符编码不会发生冲突。但由于采用了控制字符或特殊字符区分汉字与西文字符，同一汉字

数据在计算机内部的表示也不唯一，例如，一个汉字引导码后跟若干个汉字也可以表示为若干个汉字引导码后跟一个汉字，因此对程序语言有关字符类型或字符串类型语法的影响较大，对字符串的并置、分解、插入、删除、修改等操作都很不方便，使程序语言的汉化困难较大。而且，一个汉字要用两个以上的字节（把引导符也计算在内）才能表示，开销较大。此外，引导码的选择也较为困难。

(5) 带括号的汉字内码

这种汉字内码用一对控制字符或一对图形字符作为括号把汉字串括在里面，而 ASCII 码或 EBCDIC 码等西文字符不用任何控制控制字符或图形字符括起来。

这种汉字内码与带引导码的汉字内码的优缺点相似。尤其是当两个汉字串连接时，第一个汉字串的闭括号和第二个汉字串的开括号成为冗余码，显得有些累赘，而要去掉它们，则系统需要进行额外处理。用作括号的控制字符或图形字符不能派作其它用途，否则会出现二义性。特别是，在不同的操作系统中，各种终端驱动模块对于控制字符都有各自的解释和用法，因此把控制字符用作括号时，要格外小心，以免混淆。

3. 万"码"奔腾

下面，介绍台湾省采用汉字内码的情况。

台湾省采用的汉字内码目前正处于万"码"奔腾的局面。由于早期发展中文系统时，汉字内码无标准可循，致使台湾各家电脑厂商推出的中文系统采用不同的汉字内码。例如，市面上最常见的倚天、国乔、零壹这三个中文系统的汉字内码就各不同。目前，较为普遍使用的汉字内码有 BIG-5 码、5550 码、通用码、TCA 码等。下面逐一介绍这四种汉字内码。

(1) BIG-5 码

BIG-5 码是根据台湾《通用汉字标准交换码》发展而成的汉字内码。它含有常用字 5401 个，码值为 A440-C67E；次常

用字 7652 个，码值为 C940–F9D5；其它图形字符 441 个，码值为 A140–A3E0。它是按连续编码方式编码的，码值有序而且连续。

(2) IBM–5550 码

IBM–5550 码是 IBM5550 机器中使用的汉字内码。它含有常用字 5401 个，码值为 8C40–A8C9；次常用字 7652 个，码值为 A940–D1C4；其它图形字符 685 个，码值为 8940–8B4A。

(3) 通用码

通用码采用《通用汉字标准交换码》的字集和字序，并把标准交换码的第一字面的第一个字节加 80，第二字节加 80，第二字面仅第一个字节加 80。它含有常用字 5401 个，码值为 C4A1–FDCB；次常用字 7650 个，码值为 A121–F244；其它图形字符 234 个，码值为 A1A1–C2C1。通用码采用了灵活的留空编码方式。

(4) 统一内码势在必行

由于各种中文系统的汉字内码不同，给用户带来诸多不便，然而最头痛的还是电脑厂商、软件公司和软件设计者。一方面，他们要在不同的中文系统上重复开发功能相同的中文软件，浪费了人力和时间，增加了成本。另一方面，他们需要给用户提供内码转换程序（内码转换的关键是建立内码对照表），以便用户把他们在一种中文系统下开发的应用程序和数据移植到另一种中文系统上。然而，由于各种中文系统所提供的汉字及其它图形字符的个数和内容不一致，使得转换后的应用程序和数据未必会与转换前的应用程序和数据完全一致，这是因为有时两种中文系统的汉字内码根本对应不上，而且有些图形字符会因另一种中文系统未提供相应的内码而找不到码值。

中文软件环境的标准化势在必行，这是台湾电脑界的一致呼声。为了改善中文软件环境，增加中文电脑间的互通能力，提高用户的方便性，大同、宏碁、神通、旭青、诠脑、Microsoft、

HP 七家公司于 1988 年 12 月 6 日成立中文微电脑推广基金会(Chinese Microcmoputer Extended Foundation; CMEX), 简称中推会。中推会的三项目标是:

① 推广合法的中文微电脑磁碟作业系统 (磁盘操作系统);

② 推广统一的中文内码 (TCA 码);

③ 推广一致的中文微电脑应用软件界面标准, 简称 CSI(Chinese System Input-Output Interface)。

(5) TCA 码

TCA 码是台北市电脑公会(Taipei Computer Assciation)制定的汉字内码标准, 因此称为公会推荐中文内码, 简称为公会码。TCA 码有以下几个优点:

① 字符集根据《通用汉字标准交换码》排序, 内码与交换码用简单公式便可转换。

② 内码以二字节编码, 含有常用字 5401 个, 码值为 92A0-AF67; 次常用字 7650 个, 码值为 B030～D8C3; 其它图形字符 234 个, 码值为 8130～91C0。

③ 对随时增加新字提供了灵活的增加办法, 并预留给用户的造字区较大, 而且连续。

④ 为配合各类电脑的使用, 避开符号区及控制码区, 因此与程序语言、数据库管理系统及通用应用软件的冲突机会少, 符合 DOS、Unix 及其应用软件的环境。

2.3.4 汉字字形码

汉字字形是指汉字字库中存储的汉字字形信息。目前汉字字形的产生方式大多是数字式的, 即以点阵方式形成汉字。因此, 汉字字形码主要是指汉字字形点阵的代码。

汉字字形点阵有 16×16 点阵、24×24 点阵、32×32 点阵、48×48 点阵、64×64 点阵、96×96 点阵、128×128 点阵、256×256 点阵等等。

汉字字形按点阵的多少可分为通用型和精密型两类。通用型用于一般用途的电脑打印，由于信息量少，通常存储整字信息。精密型用于电脑排版，由于信息量较大，通常采用信息压缩存储技术，只存储经压缩后的字形信息。

下面，以“一”字的 16×16 点阵字形为例来说明汉字字形码(见图 2.7)。

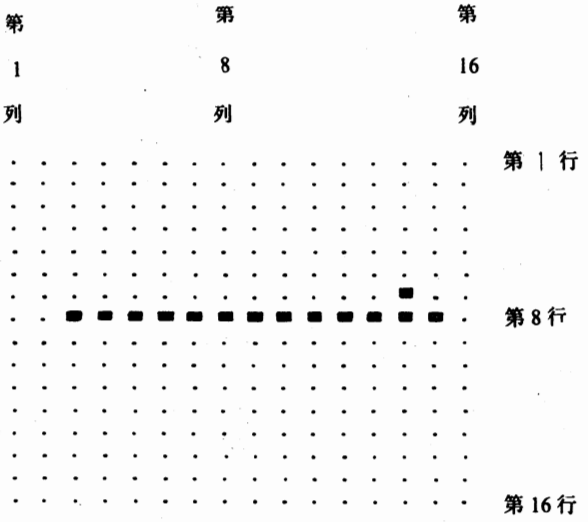


图 2.7 “一”字的 16×16 点阵字形

假设存取字形的单位为字节（通常为 8 位），则按行取点的存取次序是：先按 1 行逐个取点，然后按第 2 行，直至第 16 行；同一行左边 8 位为第一字节，右边 8 位为第二字节。由此可见，对于 16×16 点阵的汉字字形码，需要占 (16×16) ÷8＝32 个字节。例如，“一”字的按行取点字形码如图 2.8 所示。

行序号	十六进制表示
1	00 00
2	00 00
3	00 00
4	00 00
5	00 00
6	00 00
7	00 04
8	3F FE
9	00 00
10	00 00
11	00 00
12	00 00
13	00 00
14	00 00
15	00 00
16	00 00

图 2.8 “一”字的按行取点的字形码

上述字形的存取顺序是按行取点，如果是按列取点，则字形码与上述字形不同。例如，“一”字形的按列取点的字形如图 2.9 所示。

行序号	十六进制表示
1	00 00
2	00 00
3	01 00
4	01 00
5	01 00

行序号	十六进制表示
6	01 00
7	01 00
8	01 00
9	01 00
10	01 00
11	01 00
12	01 00
13	01 00
14	03 00
15	01 00
16	00 00

图 2.9 “一”字的按列取点的字形码

此外，即使对于相同点阵而且存取顺序相同的同一汉字，由于不同的字形设计，其字形码也不相同。

2.3.5 汉字地址码

汉字地址码是指汉字字库（这里主要指整字形的点阵式字库）中存储汉字字形信息的逻辑地址码（相对地址码）。目前，容纳汉字字库的存储器主要有以下两类：

(1) 半导体存储器，例如，只读存储器 ROM，可编程只读存储器 PROM，随机存取器 RAM 等；

(2) 磁性存储器，例如，软磁盘、硬磁盘、磁带、磁鼓等。

由于容纳汉字字库的存储器不同，则汉字地址码的物理表示方法也不一样。例如，半导体存储器汉字字库的汉字地址码通常由插件地址、存储模块地址、存储片地址、数据地址等组成；软磁盘字库的汉字地址码通常由盘片地址、区段地址、磁道地址、数据地址等组成；磁带字库的汉字地址码则由带地址（磁带机

号)、记录块地址、数据字段地址等组成。

然而，不论哪种汉字字库，汉字字形信息都是按一定顺序(大多按标准汉字交换码中汉字的排列顺序) 连续存放在存储器上。因此，汉字地址码大多也是连续有序的，而且与汉字内码间有着一定的对应关系。为了简化汉字内码到汉字地址码的转换，这种对应关系应尽量简单，以致有的中文系统的汉字地址码的形式甚至与汉字内码完全一致。

2.3.6 汉字控制功能码

汉字交换码、汉字输入码、汉字内码、汉字字形码、汉字地址码都是与一个具体的汉字或其它图形字符相联系的，即它们都表示一个汉字数据，称之为“图形字符码”或“汉字数据码”。在中文电脑中，除了上述表示汉字数据的图形字符码外，还有一种并不表示具体的汉字数据的代码，它们用于汉字数据的处理、传送或解释执行，例如，引导符、格式控制符、编辑功能符等，这些就是汉字控制功能码。

汉字控制功能与汉字数据码共同组成汉字数据流，贯穿于汉字输入、机内处理、汉字输出等整个汉字处理过程。因此，汉字控制功能符的设计将直接影响整个中文系统的效率和性能。

汉字功能控制码的设计包括下列两个方面:

1. 控制功能编码表示的确定

控制功能编码表示的确定要考虑以下两点:

(1) 应保持中西文控制功能编码表示的兼容性，既要适用于汉字数据处理，又要适用于西文数据处理，特别是对于通用型汉字处理系统，汉字控制功能码必须按照有关国家标准和国际标准设计;

(2) 编码要简单明确，效率要高，即应以尽可能短的码位表示尽可能多的控制功能。

2. 控制功能含义的确定

汉字控制功能码的控制功能含义的确定取决于中文系统本身的用途。对于不同的中文系统可选择不同的功能码。例如，对于通用型汉字处理系统，应该具备中文信息处理和交换的基本控制功能；对于专用型汉字处理系统（例如，中文编辑系统，中文排版系统），可以在基本功能码的基础上再扩充一些专用的控制功能码，以满足专用领域中汉字处理的需要。

2.4 汉字输入方式

在中文电脑中，首先碰到的是如何输入汉字的问题。然而，由于汉字字量多，不能象英文等拼音文字那样，仅仅用普通的西文键盘便可容易的解决输入问题。此外，由于用户的需求不同，使用的文字范围也不同。因此，较难设计出一种标准的汉字输入设备。

目前使用的汉字输入方式大致可分为以下两类:

(1) 用人工操作键盘、字盘等物理输入设备来输入汉字。键盘和字盘是目前使用最广泛的汉字输入设备。虽然这种汉字输入方式仍处在不断发展和完善的阶段，但已在各种中文电脑中成功地解决了汉字输入问题。

(2) 用人工智能方式直接对文字或语音进行识别输入。人工智能输入设备主要是汉字识别和语音识别系统或装置。汉字识别技术对印刷体和手写体汉字的识别已取得一定进展，某些系统的识准率达 98%以上，拒识率为 1.3%。误识率为 0.3%。语音识别目前仅限于特定的发音者，只能识别单个汉字、字组或词，尚不能识别连续的汉字语音。总之，汉字字模和汉字语音的识别目前尚处于研制和实验阶段，虽也有一些产品开发出来，还有待进一步提高。尽管如此，它们对于实现汉字输入方式多样化，并从根本上摆脱手工操作的汉字输入方式和提高汉字输入的效率有着深远的意义。

本节主要介绍汉字键盘输入方式，并简要介绍汉字语音输入方式和汉字字形输入方式。

2.4.1 汉字键盘输入

目前中文电脑使用的汉字键盘是多种多样的。按照操作方式，基本上可分为以下两种输入方式：

(1) 直接输入方式是在汉字键盘上选择所需汉字或接触汉字字盘直接输入汉字。它所采用的汉字输入设备是：汉字整字键盘(全键式汉字整字键盘或多段移位式汉字整字键盘)、笔触式汉字字盘、中文打字机式汉字键盘等等。

(2) 间接输入方式是利用汉字输入编码来输入汉字。它所采用的汉字输入设备是：汉字字根式键盘、标准西文键盘等等。

下面，逐一介绍上述这些汉字输入设备及其输入方式。

1. 汉字整字键盘

汉字整字键盘主要有以下两种：

(1) 全键式汉字整字键盘

全键式汉字整字键盘采用一字一键的方式，键盘上每个键都与一特定的汉字相对应。它的基本操作就是选字和击键，一次输入一个汉字。

(2) 多段移位式汉字整字键盘

在全键式汉字整字键盘的基础上，多段移位式汉字整字键盘采取了类似于字母数字键盘用移位键（Shift）来区分大小写字母的方法，即在一个文字键上定义多个汉字，用相应数目的移位键来区别文字键上的各个汉字。由于它采用了文字键与移位键配合的方法输入汉字，因此亦可称之为主辅键式汉字整字键盘。如图 2.10 所示。

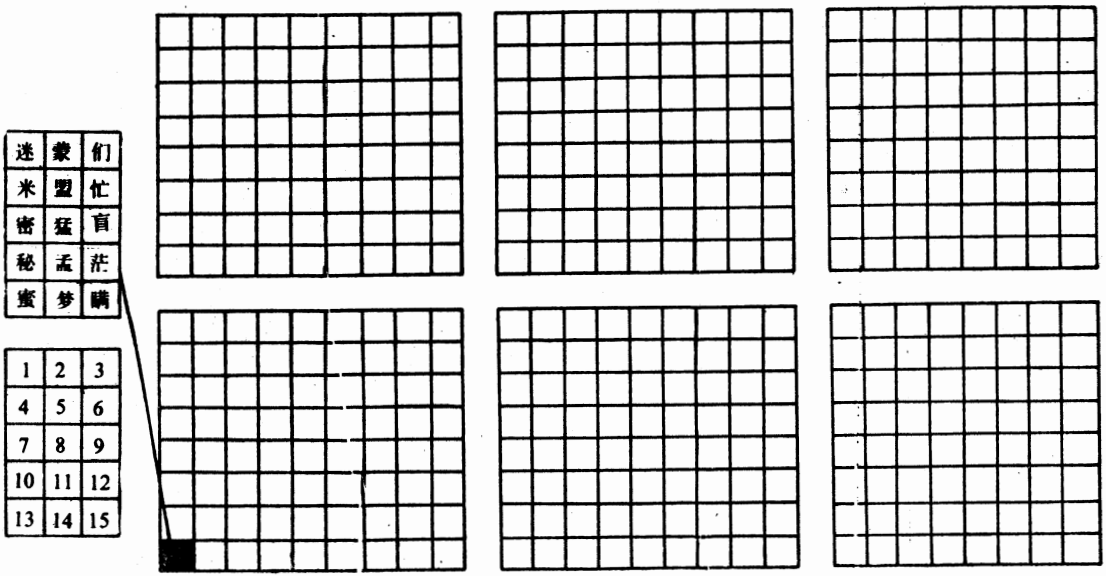


图 2.10　多段移位式汉字整字键盘示意图

图 2.10 是 15 段移位式汉字整字键盘，它有 15 个移位键，一个文字键对应 15 个汉字。由此可见，多段移位式汉字整字键盘的移位键个数越多，文字键个数就可以越少。

汉字整字键盘输入方式的优点是：直观易学，操作简便，无需记忆汉字的编码，没有重码问题，适用于输入量大的报刊编辑出版部门。它的主要缺点是：不但需要制作专门设备，增加系统成本；而且，盘面字数很多，几千个字摆在面前，找字困难，效率低；此外，收容的汉字字量受键盘尺寸的限制，尚需解决盘外字输入问题，因此推广使用比较困难。

2. 笔触式汉字字盘

笔触式汉字字盘由坐标盘、文字盘（简称字盘）、接触笔(简称触笔)、控制器四个部分组成。如图 2.11 所示。

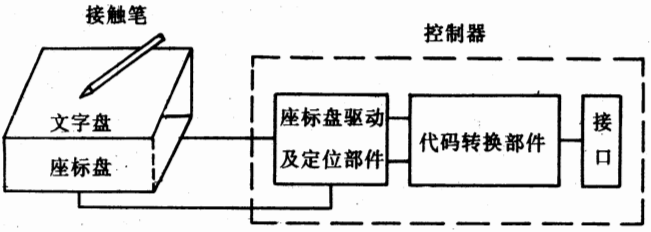


图 2.11　笔触式汉字字盘示意图

笔触式汉字字盘的工作原理是：字盘上印有按矩阵排列的汉字，将字盘盖在坐标盘上，使字盘上的汉字与坐标盘的 X，Y 扫描线的交叉点一一对应。这些交叉点就是文字位置检测点。当用触笔触及字盘上的某一汉字时，坐标盘就输出它所对应的 X，Y 坐标值。控制器的作用是对坐标盘进行驱动，检测出 X，Y 坐标值，并把它转换成对应的编码，通过接口部分传送到电脑或终端设备。

笔触式汉字字盘按坐标盘的工作原理可分为静电耦合式字盘、电磁感应式字盘、光电式字盘、磁致伸缩式字盘、压感式字

盘等。触笔一般分为有线式和无线式两种。某些压感式字盘不用触笔，而是直接用手指触摸输入。

笔触式汉字字盘也是采用整字排列，在字盘上找字与查字典一样，操作直观方便，容易被人接受。但是，由于必须用触笔在数千个汉字中选择，逐字输入，则速度不会很高；而且，用触笔在很小的格子里点字，时间长了容易疲劳；此外，由于汉字字盘尺寸的限制，盘面收容字数有限，因此需要解决盘外字输入问题。

3. 中文打字机式汉字键盘

在中文打字机上加装代码发生器（发生代码信号的机构），便成为中文打字机式汉字键盘。

按所加装的代码发生器种类可把中文打字机式汉字键盘分为电磁感应板式、全息照相编码式、铅字代码式、条形码式、坐标输入式等汉字输入装置。

中文打字机式汉字键盘的最大优点是：原来熟习中文打字机的打字员可以不经训练成为键盘操作员。但是，中文打字机的机械活动部分多，因此可靠性差，而且输入速度不高。

4. 汉字字根键盘

用以汉字字根组合方式输入汉字的键盘，叫做汉字字根键盘。使用这种键盘，最常用字一次输入，其它字用字根组合方式输入。

字根键的数量随汉字构成方法的不同而不同。如果一个汉字由较多的字根组成，则字根键较少。但是，很多汉字不仅笔画数多，而且纵横交错，错综复杂，很难分辨。因此，对每个汉字分解的部分不宜太多。否则，一方面会使汉字的分辨组合发生困难，另一方面会使编码过长，从而增加了按键次数，使输入速度降低。

现有的汉字字根键盘，有的字根个数 1000 多个，这种键盘没有脱离整字键盘的直观概念，也保留了编码比较短的特点；有

的字根个数 600 多个，每个汉字用四个字根来组合；也有的把现行的汉字通用部首 214 个当作字根；还有的把 214 个部首归并为 64 个字根，平均约三个部首共用一个字根键；更有甚者，利用标准西文键盘，标上相应的字根，例如，五笔字型字根键盘（见图 2.15）和仓颉码字根键盘（见图 2.16）。

5. 标准西文键盘

标准西文键盘是指国际上通用的键盘，亦称为字母数字键盘或 ASCII 键盘。国内提出的各种汉字输入编码方案绝大多数都是针对这种键盘的。这种键盘具有通用性，不需另外设计专门的汉字输入设备，只要利用计算机本身配备的键盘就可以解决汉字输入问题。此外，标准西文键盘还具有轻便、键位少、适于盲打，输入速度快等优点。

为了方便汉字输入，有的标准西文键盘还在按键上另外标识上辅助汉字输入的记号，例如，汉语拼音键盘、汉语注音键盘、汉字字根或笔形编码键盘等。

汉语拼音键盘是在标准西文键盘的按键上标识上汉语拼音符号形成的，如图 2.12 所示。

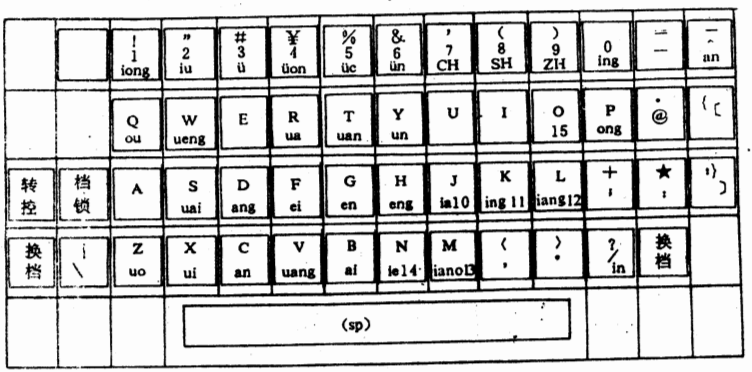


图 2.12　汉语拼音键盘

汉语注音键盘是在标准西文键盘的按键上标识上汉语注音符号形成的。图 2.13 给出国乔。零壹中文系统的注音键盘示意图；图 2.14 给出倚天中文系统的注音键盘示意图。

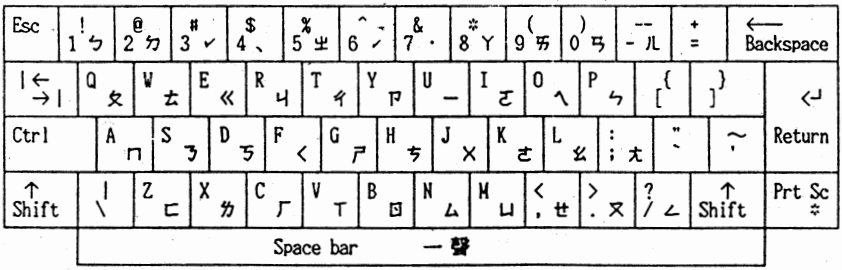


图 2.13　国乔、零壹中文系统的注音键盘

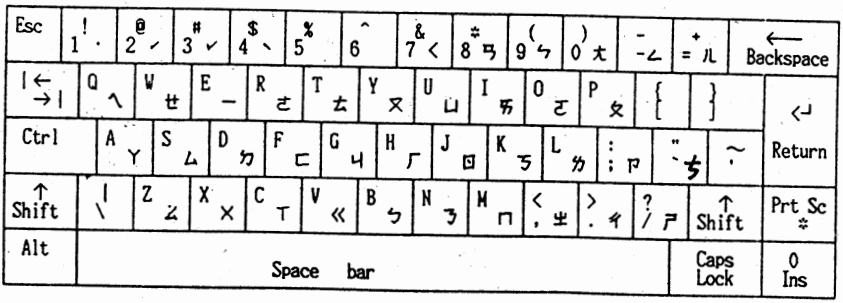


图 2.14　倚天中文系统的注音键盘

汉字字根或笔画编码键盘是在标准西文键盘的按键上标识字

根或笔画编码形成的。在汉字字形输入编码方案中，如果能把构成汉字的字根或笔画压缩到能用 48 个字母数字键表示时（一个键可以表示若干个字根或笔画），就可以用这种键盘输入汉字。图 2.15 给出五笔字型输入法的字根键盘示意图；图 2.16 给出仓颉输入法的字根键盘示意图。

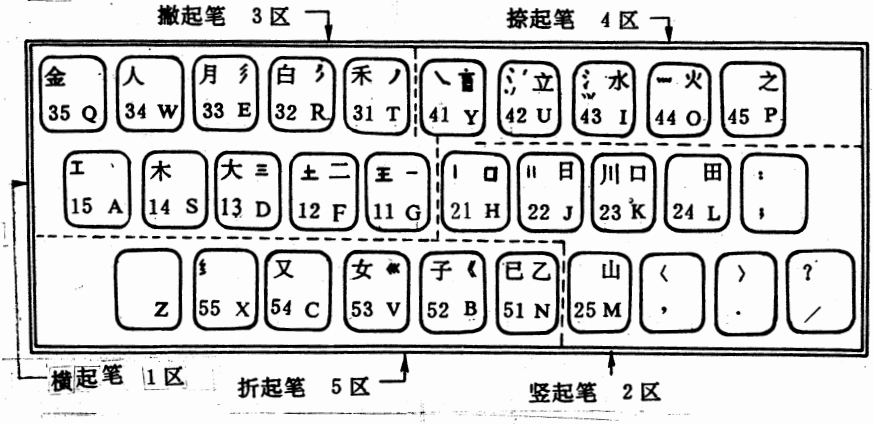


图 2.15 五笔字型键盘

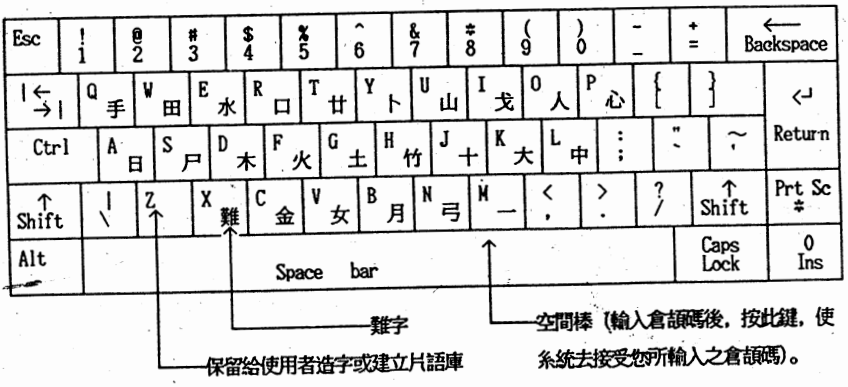


图 2.16 仓颉键盘

2.4.2 汉字语音输入

汉字语音输入，是利用产生声音的物理模型，通过语音分析手段，预先将一些语音的特征参数提取出来，并存储在处理系统中。当语音信号输入时，处理系统根据对该信号所提取的特征参数和所存储的参考特征参数进行比较，通过逻辑判断方法或“距离”测量方法，对语音进行识别辨认。

目前的汉字语音输入技术仍处于实验阶段，只能在对词汇量、读音方式等作一定限制的条件下识别语音，这样的语音识别系统大致有以下几类:

1. 单字识别系统

这类系统对使用对象不加限制，不需专人发音，可在具有噪声的机房内使用，但识别字数较少，例如，0～9 十个数字。

2. 词汇量较少的系统

这类系统采用单字输入格式，连续字之间发音要求有停顿。词汇量为 100～500 单字。对讲话环境没有太多限制，可采用电话传输系统。对讲话者性别年龄没有限制，但必须固定专人发音，若要更换发音者，必须经过训练，对词汇表中每个字发音一次或多次。

3. 通用实时语音识别系统

这类系统适用于由任意字组成的汉字短语，但组成短语的字的个数是确定的。

由此可见，目前的汉字语音输入系统仅限于特定的发音者，要求发音者间断地发每一个字的音，这就影响了声音的自然速度和人机对话的效率，有待进一步进行连续语音识别的研究。近年来，已有试验性的连续语音识别系统问世，这种系统除了需要语音的特征参数之外，还需考虑句法、语法等条件。

2.4.3 汉字字形输入

汉字字形输入方法，就是汉字字形识别方法，即用模式识别方法识别汉字，并以给定的代码存入电脑。

汉字字形识别过程简述如下:

1. 输入汉字

用光学的方法（例如，阴极射线飞点扫描方法、激光扫描方法、光敏元件方法、光导摄象管方法等），对输入原稿上的印刷体或手写体汉字字形进行光学扫描，以一定的时间间隔，取出扫描范围内的信息。

2. 变换成二值图形

把汉字字形的取样信息用 1 或 0 表示，假定汉字笔画经过的位点为 1，背景点为 0，或者相反，这样，便将输入的汉字变换成二值化的图式。

3. 输入字形的预处理

由于印刷质量和纸张的差别，通过光学方法对纸面进行扫描而得到的点阵式字形信息往往会带有各种不同程度的污染；此外，由于光学系统的转换或机械运动的误差，字形信息中也会出现杂音。由于污染和杂音的干扰，在检测的字形点阵图式中，不该有黑点的地方有了黑点，或者该有黑点的地方却窜进了白点。预处理的任务就是要除去污染和杂音，并使文字规格化。

文字规格化，是对输入文字的大小、位置等施以特定的标准化处理。对于输入的文字，如果不符合系统对文字大小的规定，则可把输入文字作为平面上的二维图形，按纵横方向分别乘以适当的比例因子，便于进行放大或缩小；如果不符合系统对文字位置的规定，则把文字区域的中心对应到特定的位置上，以校正文字的偏移，必要时可围绕中心施以适当角度的旋转，以校正输入文字的倾斜。

4. 特征抽取

对经过预处理的汉字图形抽出它们的图式特征，例如，笔画的长度、角度、端点、交叉点、笔画分布、四周特征、背景特征

等，然后把这些特征作为识别标准的学习图形，以多维向量的形式存放在分类辞书和判定辞书中。

5. 汉字字形的匹配判定

对输入汉字的图式和辞书中的标准图式进行比较，与输入图式最接近的标准图式相匹配，作为判定结果输出，送到处理系统中。由于汉字字量大，字形复杂，因此对汉字字形的匹配处理要分两步进行：首先要根据分类辞书对输入字形进行粗略的分类，给出输入文字所属的类别；然后，再将输入文字与判定辞书中所在类别的预选字进行匹配，并最后判定出结果。

汉字字形输入方式目前已趋向于实用化和商品化。典型汉字识别装置是光学字符阅读机(OCR：Optical Character Reader)，它利用汉字识别技术直接阅读汉字。近年来，国际上印刷体汉字识别技术基本上达到了实用水平，手写体汉字识别技术也已达到或接近实用水平。

2.5 汉字输出方式

汉字输出对于在中文电脑中建立友好的用户界面起着重要的作用。

汉字输入方式主要有以下几种：汉字印刷输出、汉字显示输出、汉字语音输出。本节将简要介绍这些汉字输出方式及其有关的汉字输出设备。

2.5.1 汉字印刷输出

汉字印刷输出必须考虑汉字的固有特征：汉字字量大，字形复杂，组成汉字的点阵位点多，通常还要求不同尺寸不同字体的汉字混合输出，要求既能横排也能竖排，汉字中可夹杂西文字符，此外，带汉字的表格也较为复杂，例如，标题、表头、表体、表尾、表格线的设计等。由此可见，汉字印刷技术的难度

大，设备成本也高。

汉字印刷输出是中文电脑的一种主要的汉字输出方式。用于中文电脑的汉字印刷机，按其工作原理分类，可分为击打式和非击打式印刷机。击打式汉字印刷机主要有针式点阵打印机。非击打式汉字印刷机包括激光印刷机、喷墨式印刷机、热感式印刷机、静电式印刷机、光纤管转印印刷机。上述这些用于电脑的汉字印刷机均属于非字型式印刷机，汉字字形是以点阵的形式存储在汉字字形库中。尽管目前这类汉字印刷机的印字质量尚不如采用活字的字型式印刷机，但排版自动化已势在必行，随着印刷质量的提高，电脑排版必将取代手工排版。下面，逐一介绍上述各种汉字印刷机。

1. 针式打印机

针式打印机以点阵方式输出汉字。它通常由打印机械、驱动电路和打印机控制器三个部分组成，各部分协调动作，驱动打印头上的打印针击打色带，从而把从主机送来的汉字信息打印到打印纸上。

按打印头上打印针的多少，可把用于汉字打印的针式打印机分为 9 针打印机、16 针或 18 针打印机、24 针打印机等。9 针打印机（例如，FX−100，CP−80，IBM 图形打印机等）通常用于打印 16×16 点阵汉字。由于一次纵向只能打印 9 个点，因此一行汉字必须分两次打印，第一次打印一行汉字的上半部分 8 个点，第二次打印下半部分的 8 个点。由于汉字是拼接打印而成的，因此这种打印机打印出的汉字呈矩齿状，有明显的分离感觉，字形不大美观，而且打印速度也慢，但它的造价较低，又是西文系统的标准设备，因此在要求不高的场合下还是可以使用的。16 针打印机（例如，SM−16P 等）和 18 针打印机（例如，FT−8000 等）适合于打印 16×16 点阵汉字（18 针打印机还可打印底线），一行汉字一次打印，字形质量尚可，造价也不高，对于一般用户还是可以接受的。24 针打印机（M−2024，

TH3070，LQ-1500，M-1570 等）适合于打印 24×24 点阵或更高点阵的汉字，虽然价格较高，但打印质量较好，功能较强，打印速度也快，特别适用于要求打印多字体汉字的场合。

针式打印机按本身是否带汉字字库，可分为带汉字字库的针式打印机和不带汉字字库的针式打印机。不带汉字字库的针式打印机接受从主机传来的汉字字形点阵信息，并送到打印机控制器的打印缓冲区（或称缓冲存储器）中。由于汉字字形点阵信息量大，传送汉字字形点阵信息需要大量占用主机时间，尤其是，若是汉字字形点阵信息存放在磁盘上，则尚需频繁访问磁盘，从而使系统效率降低。而且，对于不同的中文系统和不同型号的针式打印机，要配备不同的汉字打印驱动程序，致使软件开发工作量增大。带汉字字库的针式打印机内装有由 ROM 芯片构成的汉字字模库，只需从主机接收汉字内码，便可由打印机控制器在打印机内转换成对应的汉字字形点阵信息，并送至打印机缓冲区。而且，主机只要把打印机规定的控制码（例如，ESC 控制序列）送给打印机控制器，就可以完成各种变倍、横向、竖向打印以及加重打印等操作要求，充分发挥打印机本身的功能。一方面，由于主机与打印机之间只需要传送汉字内码，而无需传送汉字字形点阵信息，所以大大节省了打印汉字的传输时间，提高了主机运行效率；另一方面，无需配备汉字打印驱动程序，便可用于各种中文系统，因此汉字打印过程变得极为简单。随着 ROM 价格不断下降，带汉字字库的针式打印机还必将逐渐增多。目前，带汉字字库的针式打印机一般只有一种字模，不便于字形变换和多字体打印，而且无法适应造字的需要，有待进一步扩大功能。

2. 激光印刷机

激光印刷机是一种结构精密、功能完备的汉字印刷机。它由印刷机构和控制电路两部分组成。印刷机构综合了激光、电子照相、机电控制等多方面技术，用于输出汉字。控制电路用来控制

印刷机构的动作，同时进行印刷机与主机的信息交换。

激光印刷机印刷质量高，速度快，利用了强功率激光可直接制作胶版，出版效率高。然而，激光印刷机是一种复杂的设备，成本很高，只宜在规模较大的计算机系统中使用。近年来，已研制和生产了各种简易型激光印刷机。由于它的价格低，性能优良，在小型计算机系统中可普遍应用。近年来微型计算机系统也开始使用这种激光印刷机，特别是桌上排版系统。

3. 喷墨、热感、静电、光纤管等式样的印刷机

针式打印机和激光印刷机是当前汉字印刷机的主流，但对于其它类型的印刷机，由于各自的特点，也在某些场合被用作汉字印刷机。下面，简要介绍这些汉字印刷机。

1. 喷墨式印刷机

喷墨式印刷机是从喷咀喷射出的墨滴发生偏转而形成字形的印字记录设备。

喷墨式印刷机有以下特点:

(1) 可使用普通纸，成本低;

(2) 具有较高的印字质量;

(3) 印字速度快;

(4) 噪声小;

(5) 易实现彩色印字（在喷射机构中安置三个喷射头，分有三种颜色的墨水，即可实现彩色文字和图形的印刷)。

2. 热感式印刷机

热感式印刷机将热感印字头同热感纸相接触，计算机传送来的印字信息控制印字头瞬间发热，使热感纸受热而改变颜色，形成所需的点阵汉字字形。

热感式印刷机有以下特点:

(1) 机械传动结构简单，价格低，可靠性高;

(2) 低噪声，易小型化，可做成便携式;

(3) 可实现高分辨率;

(4) 不易实现高速度;

(5) 需使用特殊的热感纸，不宜长期保存。

3. 静电式印刷机

静电式印刷机工作原理与静电复印机很相似。首先在作为记录纸的电介质材料上直接加高压，获得汉字的静电潜象，通过静电力吸附显色剂形成可见图象（即显象），然后再经过定影，从而得到汉字。

静电式印刷机有以下特点:

(1) 可实现高速行式印字;

(2) 结构简单，价格低，工作可靠;

(3) 印字质量优良;

(4) 噪声小;

(5) 需要采用特殊的静电记录纸。

4. 光纤管转印印刷机:

光纤管转印印刷机利用光纤管直接在记录纸上曝光成象，从而得到汉字。

光纤管转印印刷机有以下特点:

(1) 印字质量高;

(2) 印字速度高;

(3) 驱动系统复杂，价格高;

(4) 必须用特殊的氧化锌纸作记录纸，造价高。

2.5.2 汉字显示输出

汉字显示也是一种汉字输出方式。

电脑上使用的显示器有阴极射线管（CRT: Cathode Ray Tube）显示、液晶显示、等离子显示、发光电子管显示等几种。但最广泛使用的是用 CRT 做成的显示器，又称作监视器(Monitor)。显示器屏幕上的图象是由许多亮度不同或色彩不同的点阵组成的。屏幕点阵密度越高，显示的图象就越清楚。人们

常用分辨率的高低来衡量 CRT 显示器的性能。

能显示汉字的显示器，称为汉字显示器。同西文字符的显示一样，汉字显示亦分为图形显示方式和字符显示方式两种。下面分别介绍这两种显示方式。

1. 汉字的图形显示方式

图形显示方式采用整屏幕点阵信息缓冲存储刷新的方法。它主要依靠位于彩色／图形适配器上的缓冲存储区，称为显示缓冲区，或显示存储器，或刷新存储器。显示缓冲区用于映射屏幕图象，通过扫描显示缓冲区的信息对屏幕进行定时刷新。对于单色显示，屏幕上的每一点与显示缓冲区中每一位存储信息一一对应。存储信息为 1 代表屏幕上的一个亮点，存储信息为 0 代表屏幕上的一个暗点；反之亦然。对于彩色显示，一般采用 RGB 三色控制法，用红、绿、蓝三种基色组成多种颜色。这时，屏幕上的一点对应显示缓冲区的若干位信息。若屏幕上的一点对应三位存储信息，就能获得 8 种颜色。若一点对应四位存储信息，则可组合出 16 种颜色。

图形显示方式既能显示字符（西文字符或汉字），又能显示图形，容易达到图文并茂的效果，在对显示速度要求不高而且显示器分辨率较高的情况下，采用这种显示方式来显示汉字是适宜的。

然而，图形显示方式需要容量较大的显示缓冲区。另外，当显示字符时，如果屏幕要有些改动，例如，增加一个字符，删除一个字符，字符行的滚动等，往往要使显示缓冲区的内容“移位”，这种“移位”传输的信息量有时是很大的；而在大多数情况下只要求显示字符（特别是西文字符只有几十种），若把屏幕上的每个字符的点阵信息都存储在显示缓冲区中，就会有相当的重复，而且势必会影响显示速度。此外，采用图形显示方式来显示汉字，对显示器的分辨率要求较高。为了保持中西文显示环境的一致性，汉字显示器每屏应能显示 25 行，每行应能显示 40 个汉

字。假设汉字的点阵是 16×16，则必须保证显示器的屏幕点阵至少为 640×400；否则，只能根据显示器的分辨率减少汉字显示行数和每行显示的个数，或压缩汉字显示的点阵数(即抽点显示)。

2. 汉字的字符显示方式

字符显示方式也有一个显示缓冲区，亦称为字符信息存储器，其中存储字符的内部码或地址码。字形发生器把显示缓冲区中的地址码转换为字形，在屏幕上显示出来。

由于用于字符显示方式的显示缓冲区中存储的是字符的地址码，而不是字符的点阵信息，因此节省存储器，同时易于修改屏幕信息，可获得较高的汉字显示效率。

用以驱动显示器的有两类适配器：单色适配器和彩色／图形适配器。这两类适配器上都备有显示缓冲区。单色适配器只能驱动显示器以字符显示方式显示西文字符；彩色／图形适配器既能驱动显示器以字符显示方式显示西文字符，又能驱动显示器以图形显示方式显示汉字和图形。但是，这两类适配器均无法驱动显示器以字符显示方式显示汉字，因此只能另加硬件来实现汉字的字符显示方式，例如，采用汉卡来构造汉字字形发生器和显示缓冲区。由于汉字显示是完全用硬件按字符显示方式实现的，因此没有额外的开销，速度快，工作效率高；而且不必修改 DOS 的 INT10 显示模块，使程序语言和应用软件显示环境不变，达到中西文软件最大限度的兼容。

2.5.3 汉字语音输出

汉字语音输出，严格地说，应当是汉语语音输出。

汉语语音输出是中文电脑的一种输出手段，有着重要的使用价值。

汉语语音输出是指利用语音的数字信息，采用语音合成(Speech Synthesis)的方法由数字信息还原成模拟量而输出人耳能

听到的语音，这种输出方法又称为语音合成。

. 汉语语音输出可以采用以音节为单位的输出方法，因为汉语的特点是一个字一个音节，字是独立的发音单位，这比一般采用拼音文字的语言（例如，英语、法语、俄语等）要简单得多。汉语的音节共有一千二百多种，比较单纯。而采用拼音文字的语言虽然音素不多，但是音节种类繁多，而且多音节的词很难完全按单个音节的发音来组合。正由于汉语发音的特点。汉语语音全部控制参数的存储量较少，而且发音的控制也较简单。

第三章　中文操作系统

操作系统是计算机的软件核心。中文操作系统是具有汉字处理功能的操作系统。目前，大多数中文软件是基于中文操作系统开发和使用的。因此，了解中文操作系统与西文操作系统的区别，对于中文软件的设计与应用是十分必要的。

本章从汉字输入、汉字输出、汉字造字、汉卡、中文操作系统支持的中文软件等几个方面来讨论中文操作系统。

3.1　汉字输入

汉字输入是中文电脑推广使用的首要问题，是中文信息处理的瓶颈。

目前，汉字输入有键盘输入、语音输入和字形输入等输入方式。键盘输入是当前汉字输入的主要手段。为了提高汉字的输入效率，充分利用计算机的原有设备，目前大多使用标准西文键盘，用键盘符号来表示汉字，通过键入汉字输入码来输入汉字。因此，本节主要讨论汉字输入编码。在讨论汉字输入编码之前，首先介绍如何通过键盘输入汉字及其所涉及到的基本概念。最后，介绍智能化汉字输入方法。

3.1.1　如何输入汉字

为了保证中西文兼容，在汉字输入环境下既能输入西文字符，又能输入汉字，中文操作系统一般提供两种输入方式：西文输入方式和汉字输入方式。在西文输入方式下，通过键入西文字符来输入这一西文字符。在汉字输入方式下，无论采用哪种汉字

输入方案，都是通过键入汉字输入码来输入汉字的。

1. 汉字输入提示区

中文电脑的显示屏幕分为正文区和提示区两部分。正文区用于显示正文。提示区主要用于辅助汉字输入，此外还提供一些必要的系统状态信息，是中文操作系统所特有的。屏幕上有两个光标：一个光标在正文区，用于指出下一个字符在屏幕上显示的位置；另一个光标在提示区，用于指出当前汉字输入码下一个码元显示的位置。

汉字输入提示区有两种显示方式：一种是提示行，另一种是提示窗口。提示行一般位于屏幕的最下面一行、两行或三行。提示窗口采用屏幕窗口方式，这样就能提高屏幕的有效利用面积，窗口的位置可由用户动态地在屏幕中上下左右移动。

汉字输入提示区包括下列几部分：

(1) 输入方案名是当前所选择的输入方案的名称。例如，在汉字输入方式下，对于拼音输入方案，显示“拼音”，对于五笔字型输入方案，显示“五笔”等等；在西文输入方式下，对于ASCII字符输入方式，显示“ASC”、“ASCII”、“英数”等等。

(2) 汉字输入码是用来代表汉字或词组的一组键盘符号。对于同一个汉字，在不同的输入方案中，对应不同的汉字输入码。例如，“啊”字在拼音输入方案中对应的输入码为a，而在区位输入方案中对应的输入码为1601。

构成汉字输入码的键盘符号称为码元。不同的输入方案可能采用不同的码元类型，例如，有的输入方案采用字母，有的采用数字，有的则兼用字母和数字，有的还用小键盘，有的甚至使用键盘的全部符号。一个输入方案中最长一个汉字输入码的码元个数称为最大码长。

(3) 待选重码汉字区：若干个不同的汉字或词对应于同一个汉字输入码，称这些汉字或词为重码汉字或重码词。待选重码汉

字区用于显示当前汉字输入码所对应的重码汉字或重码词（例如，同音、同形、同义汉字或词）。每个重码汉字或词前面设置一个代表字符，供查找和选择汉字或词时用。例如，CCDOS 最多可同时显示 10 个重码汉字，每个汉字前面各有一个数字编号，供挑选汉字时使用。又例如，联想式汉字系统的提示区信息安排如图 3.1 所示。

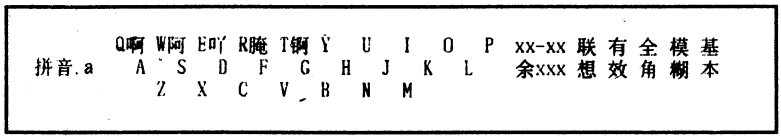


图 3.1 联想式汉字系统的提示区

其中，待选重码汉字区有按键盘形状排列的 26 个字母，每个字母后面跟有汉字或词组，可以从中选择和输入需要的汉字和词。

当一个汉字输入码所对应的重码汉字或词较多时，待选重码汉字区无法将它们一次显示出来，则需要把它们分页显示。当键入汉字输入码后，第一次显示的重码汉字或词为第一页。如果没有所需的汉字，则可通过翻页的方法来显示其它页。按规定的控制键可显示下一页或上一页，也可循环显示各页的重码字或词，以供查找和选择所需要的汉字或词。

(4) 标志区用于显示是否还有待显示的重码字或词，并提供一些必要的系统状态信息。例如，CCDOS 用→表示翻页方向为正，即往后翻页，且还有下一页重码字或词未显示；用←表示翻页方向为负，即往前翻页，且还有上一页重码字或词未显示；用 00 表示在该方向上已是最后一页了，没有待显示的重码字或词了。第一次显示首页重码字或词时，假定翻页方式为正。又例如，联想式汉字系统用“余 xxx”表示尚有 xxx 个重码字或词未显

示完。

2. 汉字输入步骤

(1) 切换输入方案：汉字输入方式与西文输入方式（一般指ASCII字符输入方式）之间的切换，各种汉字输入方案之间的切换，均是用控制键来实现的，例如，Alt-F1到Alt-F10，Alt-Ctrl-0到Alt-Ctrl-9，Shift-F1到Shift-F10等。各种中文操作系统规定使用的控制键不一样。当按动规定的控制键后，提示区出现所选择的输入方案的名称。

(2) 键入汉字输入码：在选定了汉字输入方案后，键入相应的汉字输入码，提示区显示键入的输入码。

(3) 选择汉字：在键入汉字输入码后，提示区的待选重码汉字区上出现若干个重码汉字或重码词，通过键入重码字或词前面的代表字符，来选择需要的汉字。对于无重码类编码，无需经过选择汉字这一步骤；对于重码类编码，若某一特定的汉字输入码只有一个对应的汉字，则也无需选择汉字。如果重码字或词太多，而在当前待选重码汉字区中又找不到所需要的汉字，则可通过翻页到下一页或上一页寻找所需要的汉字，直至把所选择的汉字输入到计算机中，并在正文区的当前光标所在处显示出来。

3. 全形与半形

在中文操作系统中，无论是输入、显示还是打印，对英文字母、数字及其它特殊字符等西文字符都设置了两种宽度的字形：全形和半形。全形字符就是与汉字一样大的西文字符，半形字符就是半个汉字宽的西文字符。广义地说，汉字也是全形字符。全形与半形亦称为全角与半角。一般说来，字母和数字都有相应的全形字符，在汉字字符集中都提供了它们的图形字符。除此之外，其它键盘符号也有对应的全形字符，不同的中文操作系统有着不同的规定。

例如，IBM5550 中文操作系统规定的键盘符号及其对应的全形字符如下：

键盘符号	<	>	?	−	,	.	/	\
全形字符	《	》	？	”	，	。	、	“

又例如，联想式汉字系统规定的键盘符号及其对应的全形字符如下：

键盘符号	~	!	@	#	$	%	∧	&	*	(	)
全形字符	～	！	@	#	¥	%	∧	&	*	（	）
键盘符号	−	+	¦	、	−	=	\	{	}		
全形字符	−	+	|	”	−	=	、	{	}		
键盘符号	[	]	:	″	;	′	<	>	?	,	.
全形字符	[	]	：	”	；	“	<	>	？	’	。

对于全形字符的输入方法，不同的中文操作系统有着不同的规定。但是，不外乎设置两种输入环境，用以区别全形字符和半形字符，在一种输入环境中可以输入半形字符，而在另一种输入环境中可以输入全形字符。下面，举几例说明之。

(1) 在某些汉字输入方案中，可以通过键入汉字输入码来输入全形字符。例如，在区位输入方案中，可用区位码 0333 输入大写字母 A 的全形字符，亦可用区位码 0101 输入全形字符“。”。

(2) 设置常用输入方案，按规定键进入这一输入方案后，提示区中分页显示常用汉字及其它图形字符，其中包括字母、数字、标点符号等全形字符，通过键入它们前面的代表字符，便可输入所需要的全形字符。

(3) 在西文输入方式和汉字输入方式之外，另设全形输入方式。西文输入方式与全形输入方式的区别在于：在西文输入方式下（例如，ASCII 字符输入方式），通过键入西文字符来输入它的半形字符；在全形输入方式下，通过键入西文字符来输入它所

对应的全形字符。

(4) 在西文输入方式下设置两种输入状态：半形输入状态和全形输入状态，通过规定控制键来切换状态。在半形输入状态下，通过键入西文字符来输入它的半形字符；在全形输入状态下，通过键入西文字符来输入它所对应的全形字符。

(5) 在汉字输入方式下，通过未被用作汉字输入码的码元和选择汉字用的代表字符的其它键盘符号，来输入全形字形。例如，如果中文操作系统用小写字母和数字来作为汉字输入码的码元，并用大写字母作为重码字或词前面的代表字符，则可规定通过键入除字母和数字之外的其它特殊字符来输入它们所对应的全形字符。此外，也可在某一汉字输入方案中，通过键入这一输入方案的非输入码码元和非代表字符，来输入全形字符。例如，在拼音输入方案中，可通过键入数字来输入它的全形字符；在区位输入方案中，可通过键入字母来输入它的全形字符。当然，如果在中文操作系统中未规定上述的全形字符输入方法，那么在汉字输入方式下，可通过未被用作汉字输入码的码元和选择汉字用的代表字符的其它键盘符号，来输入半形字符。

3.1.2 汉字输入编码

汉字输入编码与汉字字典的查字法相似。任何一种汉字字典都有一种或几种对汉字进行分类排序的查字法。就其实质而言，查字法也可以说是一种汉字编码法，只不过用于电脑的汉字输入编码方法与用于汉字字典的查字法在要求上有所差异而已。汉字字典的查字法常常采用部首查字法、拼音查字法、四角号码查字法等等。虽然这些查字法适用于汉字字典，但是，如果用作电脑的汉字输入编码，则会出现大量汉字对应于一个编码的严重的重码现象，这对于电脑的汉字输入是不适用的。

汉字输入码按汉字属性分类，可分为汉字形输入码、汉字字音输入码、汉字字义输入码、汉字字形字音字义混合输入码、其

它汉字输入码。

汉字输入码按字和词分类，可分为汉字单字输入码和汉字词组输入码。各种汉字输入方案一般都有相应的词组输入码。

汉字输入码按有无重码分类，可分为重码类编码和无重码类编码。无重码类编码是指一个汉字的编码是唯一的。重码类编码是指一个编码对应若干个汉字。对于无重码类输入方案，每汉字的编码都是唯一的，当键完一个汉字的编码后，该汉字会立即在屏幕光标所在位置上显示出来，适于盲打，对于重码类输入方案，一般采用人机对话式输入方法，当键完一个汉字的编码后，提示区上会把这一编码所对应的所有汉字一次或分页显示出来，用户可进行挑选，通过键入重码汉字或词前面的代表字符，便可输入所需要的汉字。

汉字输入码按检索的方法分类，可分为计算检索编码（例如，内码，国标码、区位码等)和查表检索编码。

汉字输入码按用户类型分类，可分为普及型编码和提高型编码。普及型编码规则简单，容易学习，但输入速度较慢，适于一般人员使用。提高型编码重码率低，输入速度快，但规则复杂，不容易掌握，适于专业操作员使用。

下面，按汉字属性分类和按字和词分类，总结和概括汉字字形输入码、汉字字音输入码、汉字字义输入码、汉字字形字音字义混合输入码、其它汉字输入码及其汉字词组输入码，并简要介绍汉字输入编码的定量和定性评测标准。

1. 汉字字形输入码

汉字是由字根构成的，字根又是由笔画构成的。汉字字形输入码是按组成汉字的字根或笔画编码的。在不同的汉字字形输入方案中，字根或笔画亦称为笔形、部首、偏旁、角形、部件、字元、字首、字型、字式等。

下面，较详细地介绍五笔字形码和仓颉码，扼要地介绍几种

其它的汉字字形输入码。

(1) 五笔字型码：五笔字型编码方案把汉字分解为字根，根据由五种笔画（一，丨，丿，丶，乙)构成的字根和汉字的三种字型(左右型、上下型、杂合型）进行编码。

五笔字型码的编码规则是:

①汉字由字根组成，字根由笔画构成。

②把汉字的笔画分为五种：横（一)、竖（丨)、撇（丿)、捺（丶)、折（乙)，并按使用频度分别赋予代号 1，2，3，4，5。

③根据字根的组字能力和使用频度，优选字根 130 个，按首笔笔画分为五类，各对应标准字母数字键盘的一个区，区号为首笔画的代号，每个区中的字根又按次笔笔画分为五个位，位号为次笔画的代号，从键盘中部向两端放射排列。共计 25 个键位，各键位的代码既可用区位号（11–52）表示，又可用对应的英文字母表示，称为字根码。五笔字型键盘分区及键位安排情况如图 2.15 所示；按助记词顺序排列的字根总表如图 3.2 所示。

区	位	字母	代码	笔画	键名 基本字根
1 横起类	1	G	11	一	王土戈五一
	2	F	12	二	土士二干十寸雨
	3	D	13	三	大犬三王丰开古石厂ナナナ
	4	S	14		木丁西
	5	A	15		工戈艹廿廿匚七弋
2 竖起类	1	H	21	丨	目且上止卜卜虍虍
	2	J	22	丨丨	日曰田早丨刂丿虫
	3	K	23	丨丨丨	口川
	4	L	24	丨丨丨丨	田甲口四皿皿田车力
	5	M	25		山由贝冂冂几
3 撇起类	1	T	31	丿	禾禾竹丿ナイ攵
	2	R	32	丿丿	白手扌手丿ナ亻斤斤
	3	E	33	丿丿丿	月月丹彡氵犭乃用豕豕风
	4	W	34		人亻八犬犬
	5	Q	35		金钅勹鱼犭义儿夕夕夕
4 捺起类	1	Y	41	丶	言讠文方广讠亠主
	2	U	42	丶丶	立辛冫氵丬亠六冫门疒
	3	I	43	丶丶丶	水氵水氵氵灬业小⺍业
	4	O	44	灬灬	火灬灬米
	5	P	45		之辶廴宀冖礻
5 折起类	1	N	51	乙	已巳己⺕尸尸乙心忄小羽
	2	B	52	乙乙	子孑耳阝卩已了也凵
	3	V	53	乙乙乙	女刀九臼彐巛
	4	C	54		又厶マ巴马厶
	5	X	55		纟幺弓匕匕幺

图 3.2 五笔字型字根总表

④根据构成汉字的字根之间的位置关系，可把汉字分为三种字型，如图 3.3 所示。

字型代号	字型	图　示	字　例
1	左 右		汉湘结封
2	上 下		字莫花华
3	杂 合		困凶这司乘本重天且

图 3.3　汉字的三种字型

由于“叭”和“只”、“旮”和“旭”这样的字，单凭字根无法区分，有必要采用上述字型信息，五笔字型的“型”源于此。

由于“沐”、“汀”、“洒”三字中，“木”、“丁”、“西”在同一键上，引起了重码，但这三个字的末笔画不同，故可用最后一笔画加以区分。

将汉字末笔画代号与字型代号两者结合起来，形成一个识别代码，称为末笔字型交叉识别码，简称为识别码，追加在不足四个字根组成的汉字的字根码之后。

⑤键名汉字输入：各键位上标识的字根，称为键名。共 25 个，其中绝大部分是汉字。它们的输入方法是：把所在键连击四下。

例如，

王：	11	11	11	11	大：	13	13	13	13
	G	G	G	G		D	D	D	D
之：	45	45	45	45	言：	41	41	41	41
	P	P	P	P		Y	Y	Y	Y

⑥成字字根输入：在字根总表（见图 3.2)中，除键名外本身

是汉字（包括基本集中所含的“亻”、“氵”等部首在内）的字根，称为成字字根。它们的输入方法是：先按该字根所在键一下，再根据书写顺序依次键入第一、第二及最末一个笔画的代码。不足四笔时，按空格键表示结束。

例如，

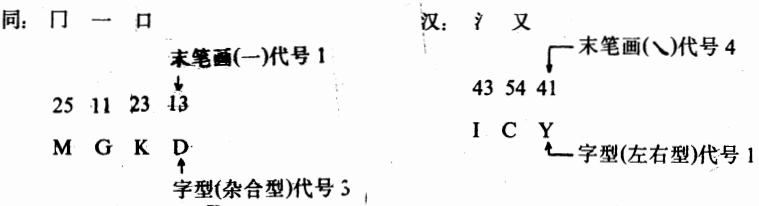
键名	首笔	次笔	末笔	
方：	方	、	一	乙
	41	41	11	51
	Y	Y	G	N
石：	石	一	丿	一
	13	11	31	11
	D	G	T	G
用：	用	丿	乙	丨
	33	31	51	21
	E	T	N	H
干：	干	一	一	丨
	12	11	11	21
	F	G	G	H

键名	首笔	次笔	末笔	
氵：	氵	、	、	一
	43	41	41	11
	I	Y	Y	G
力：	力	丿	乙	
	24	31	51	空格
	L	T	N	空格
厂：	厂	一	丿	
	13	11	31	空格
	D	G	T	空格
十：	十	一	丨	
	12	11	21	空格
	F	G	H	空格

⑦合体字输入，键名和成字字根之外的任何字，称为合体字。合体字的取码规则是：根据书写顺序依次取第一、第二、第三和末一个字根的代码，不足四个根码时，追加识别码。

例如，

给：	纟	人	一	口	逾：	人	一	月	辶
	55	34	11	23		34	11	33	45
	X	W	G	K		W	G	E	P
副：	一	口	田	刂	觸：	丿	目	田	一
	11	23	24	22		31	21	24	11
	G	K	L	J		T	H	L	G



(2) 仓颉码是台湾的一种汉字输入编码方案。

仓颉码的编码规则是：将汉字用 24 个仓颉字母（组成汉字的基本笔画）及其相关字形编码。仓颉字母及其相关字形如图 3.4 所示。仓颉键盘安排情况如图 2.16 所示。

①仓颉字母可分为哲理、笔画、人体、字形四大类。

i. 哲理类：日月金木水火土

此种象形字与人类生活息息相关，具有哲理的味道。

ii. 笔画类：竹弋十大中一弓

斜（丿）、点(丶)、交（+)、叉（乂)、纵（|)、横（–)、钩（亅）这些笔画在汉字中使用频繁，由于它们都不是独立的汉字，所以分别以具有这些笔画特征的竹、戈、十、大、中、一、弓为其代表字母。

iii. 人体类：人心手口

此种象形字代表人体的器官，而且人、手、口常为汉字的偏旁，出现频度高。

iv. 字形类：尸甘山女田卜

此种象形字用字形本身的形状来表现其涵义。由于取字形时，匚、彐、艹、凵、亻、口、亠、辶等字形单独存在并无意义，所以用具有这些字形特征的汉字来代表。

②相关字形：要用 24 个仓颉字母完全组合所有的汉字是不可能的，因此要选取与某一仓颉字母意义相类似的字或它的变形，作为这个仓颉字母的相关字形，亦称为辅助字形。相关字形共有 75 个。

相关字形大抵是按下列四个原则产生的：

组别	名称	英文字键	代表意义	字母	相关字形
1	哲理类	A	日	日	日
		B	月	月	月
		C	金	金	金
		D	木	木	木
		E	水	水	水
		F	火	火	火
		G	土	土	土
2	笔画类	H	斜	竹	竹
		I	点	戈	戈
		J	交	十	十
		K	叉	大	大
		L	纵	中	中
		M	横	一	一
		N	勾	弓	弓
3	人体类	O	人	人	人
		P	心	心	心
		Q	手	手	手
		R	口	口	口
4	字形类	S	侧	尸	尸
		T	并	廿	廿
		U	仰	山	山
		V	纽	女	女
		W	方	田	田
		Y	卜	卜	卜

图 3.4　仓颉字母及其相关字形表

i. 与仓颉字母的形状相似或者按仓颉字母形状变化而得。例如，“夕”是月的相关字形；“⺍”是火的相关字形；“器”是日的相关字形；“士”是土的相关字形。

ii. 字形的特征与仓颉字母的意义相同。例如，弓的代表意义为钩，而亅，乙的字形特征也为钩，故亅，乙为弓的相关字形；中的代表意义为纵，而丨、丰的字形特征亦为纵，故丨、丰为中的相关字形。

iii. 字形的形状是仓颉字母的一部分。例如，“冂”是月的部分字形，故为月的相关字形；“丷”是金的部分字形，故为金的相关字形。

iv. 由某些相关字形变形而来。例如，“夕”是月的相关字形，而“𠂉”又是由“夕”变形而来，故“𠂉”也是月的相关字形。

③仓颉字母及其相关字形的使用：24个仓颉字母各代表一个汉字。按它所对应的英文字键，便可输入相应的字形。

相关字形只不过是汉字组合的字根单元，通常单独存在但不能够表达字的意义（尽管也有少数相关字形本身是有意义的汉字）。若要把相关字形拿来单独使用，则必须把相关字形当作一完整的字，再按仓颉字母分解它的笔画。例如，

匕：	山竹	乙：	弓山
工：	一中一	八：	竹人
又：	弓大	七：	十山
士：	十一	小：	弓金
艹	十十	儿：	中山

凡是不能以一码取足的汉字，均称为组合字。只有在仓颉字母及其相关字形表（见图3.4)中选出恰当的字形，才能正确地组合汉字。

④取码规则

i. 取码顺序，汉字的组合形式可分为并列性、多列性、上下形、外内形和连体形五种。例如，

并列：咽姨促缘郭磋伯

多列：彬潮挪衍论湘僻

上下：昌卡告企芳思咨

外内：囤固因囫回囚困

连体：更两正乘央马乌

取码的顺序是由外而内，由上而下，由左而右。例如，

由外而内：困　田木　　由上而上：昆　日比(由左而右:心心)

周　月土口

用　月手　　全　人一土

巨　尸尸　　杰　木火

由左而右：诱　言秀(卜口竹木尸)

淋　水木木

休　人木

仰　人竹山中

粥　弓火木弓

ii. 连体字取码

凡字体笔画相互交连、无法分离者为连体字。依次取首、次、三、尾四码，不足四码者全取。例如，

妻　十中女

舟　竹月卜戈

商　卜全月口

iii. 分体字取码

分体字是由字首、字身两部分组成。凡字体可作上下、左右或内外分离者，其最上方、最左侧、最外面之字形称为字首，而字首以外的部分称为字身。分体字按字首和字身两部分取码。

(i) 字首取码：限取一至二码，超过时则只取首、尾二码：

a. 一码取足，只取此码。例如，

字首

枝　木

字　宀

厚　厂

b. 凡字首不能一码取足者，均取首、尾二码。例如，

字首

爬　爪　竹人

照　昭　日口

(ii) 字身取码，限取一至二码，不超过三码者，依次全取。例如，

字身

斧　斤　竹中

字　子　弓木

超过三码者分为连体字身与分体字身分别取码：

a. 连体字身取码

依次取首、次、尾三码。

b. 分体字身取码

字身分为次字首、次字身者称为分体字身。

ⓐ 次字首一码取足者，次字身取首、尾二码；

ⓑ 次字首不能一码取足者，次字首取首、尾二码，次字身取首、尾二码。

例如，

字首　字身

建　廴　聿

明　日　月

字首　字身

眸　日　牟

稼　禾　家

字首　字身

爸	父	巴
鸽	合	鸟
疫	疒	殳

(3) 首尾码：只取汉字的字首码（左上角的笔形）和字尾码(右下角的笔形)。取码规则是：先左右后高低，不分笔形顺序；对于内外型汉字，取外形为字首，内形为字尾。

(4) 简易码：它由仓颉码简化而来，只取仓颉码的首尾两码。取码规则是：字形仍以仓颉字母及其相关字形为基础，仍是由上而下、由左而右、由外而内取码，但只取首尾两码。因此，亦称为仓颉首尾简易输入码或速成输入码。

(5) 五笔画：按构成汉字的五种基本笔画，横（一）、竖(丨)、撇（丿）、捺（㇏）、折（乙）依次取码。

(6) 笔形码：按构成汉字的八种基本笔画，横（一）、竖(丨)、撇（丿）、点(㇔)、折（㇕）、弯（㇈）、叉（十）、方（口）依次取码，亦称为八笔画。

(7) 对话：汉字采用25个基本笔形(6个基本笔画一、丨、丶、丿、㇄、㇕，7个常用部首土木氵艹人扌口，12个笔画组合)编码，按从上向下，从左向右、从外向内的顺序分解和取码。

(8) 三角号码：取汉字三个角的字形，字形为300个基本符号，分为99组，赋予01–99的代码。取角原则是：若一个汉字的四个角都是基本符号时，只取三个角。取形原则是：该汉字已使用过的基本符号不再重复使用。

(9) 钱码：优选了若干种使用频度较高的常用笔画结构，作为汉字的基本组成单元，每个汉字只需输入三码。

(10) 字元编码法，选取笔画和常用基本构件（偏旁、部首等）共120个，按字形相似或字义相似分为32组，称为汉字字元，每一组赋予一个代码，用32个代表键输入汉字。

2. 汉字字音输入码

汉字字音输入码是用汉字字音编码的。常见的汉字字音输入码是汉语拼音码和汉语注音码。

汉语拼音和汉语注音把汉字字音看作是由声、韵、调三大要素确定的。每个汉字对应一个音节，每个音节由一个韵母或一个声母与一个韵母组成，并有一个声调。汉语共有声母 21 个，韵母 35 个，声调 4 个：阴平、阳平、上声、去声（亦称为一声、二声、三声、四声），外加一个轻声。声、韵组合起来有 417 个音节形式，声、韵、调组合起来则有 1330 个音节形式。

实质上，以普通话为基础的汉语拼音方案和以国语为基础的汉语注意方案只是声母和韵母的表示形式不同罢了。下面，给出两者声母和韵母的对照形式。

声母表：

b	p	m	f	d	t	n	l
ㄅ	ㄆ	ㄇ	ㄈ	ㄉ	ㄊ	ㄋ	ㄌ
g	k	h		j	q	x	
ㄍ	ㄎ	ㄏ		ㄐ	ㄑ	ㄒ	
zh	ch	sh	r	z	c	s	
ㄓ	ㄔ	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ	

韵母表：

a	o	e	ai	ei	ao	ou	an	en	ang	eng	ong
ㄚ	ㄛ	ㄜ	ㄞ	ㄟ	ㄠ	ㄡ	ㄢ	ㄣ	ㄤ	ㄥ	ㄨㄥ
i	ia	ie	iao	iou	ian	in	iang	ing	iong		
ㄧ	ㄧㄚ	ㄧㄝ	ㄧㄠ	ㄧㄡ	ㄧㄢ	ㄧㄣ	ㄧㄤ	ㄧㄥ	ㄩㄥ		
u	ua	uo	uai	uei	uan	uen	uang	ueng			
ㄨ	ㄨㄚ	ㄨㄛ	ㄨㄞ	ㄨㄟ	ㄨㄢ	ㄨㄣ	ㄨㄤ	ㄨㄥ			

ü üe üan ün er

ㄩ ㄩㄝ ㄩㄢ ㄩㄣ ㄦ

下面，介绍三类汉字字音输入方法：汉语拼音输入法、汉语注音输入法、地方话拼音输入法。

(1) 汉语拼音输入法：由于汉语拼音可采用 26 个英文字母来拼写、因此汉语拼音输入法可用标准西文键盘来输入汉字。

下面，介绍几种汉语拼音输入方案。

①拼音码是把构成字音的拼音字母当作码元来键入的汉字输入码。每个汉字的字音由 1–6 个拼音字母的构成，码长不等。

②双拼码是拼音码的一种简化编码形式，它把每个声母和韵母各用一个字母表示，这样就可用两个字母表示一个汉字的字音，从而减小击键次数，提高输入速度。不过，必须对字母对声母和韵母的替代关系作周密考虑，以避免由此而产生许多新的重码字，也要便于操作人员的记忆。

下面，介绍一种变形的双拼码，它把一部分声母和韵母简化为一个字母键，如下：

拼音	字母键	拼音	字母键
zh	a	ai	l
ch	i	en	f
sh	u	eng	g
an	j	ing	y
ang	h	ong	s
ao	k	ü	v

由于有了 12 个键代替某些声母和韵母，不管一个汉字的拼音字母有多长，都可用三键输入。例如，“状”字的拼音为“zhuang”，拼音字母长达 6 个，仍可用三键“auh”输入“状”字。

③声韵调码是在上述用声母和韵母构成汉字字音的声韵码基础上加上声调，可用 0，1，2，3，4 分别表示轻声、阴平、阳平、上声、去声、亦可用字母或其它字符标调，例如，分别用空白、－、／、V、＼表示轻声、阴平、阳平、上声和去声。

(2) 汉语注音输入法:汉语注音输入法与拼音输入法相似，只是由于注音输入法采用的注音符号与标准西文键盘上的字母相异太大，需在标准西文键盘上标明注音符号，称为注音键盘。台湾各电脑厂商规定的注音键盘配置各不同。图 2.13 和图 2.14 分别给出了国乔、零壹、倚天中文系统的注音键盘配置图。

下面，以倚天注音键盘为例，说明注音键盘的排列原理。

①依照英文字母的发音:

7 C O I E

ㄑ ㄒ ㄛ ㄞ ㄧ

②依照英文音标的发音:

B P M F D T N L K H G J S A R

ㄅ ㄆ ㄇ ㄈ ㄉ ㄊ ㄋ ㄌ ㄎ ㄏ ㄍ ㄐ ㄙ ㄚ ㄜ

③依照英文字母字形相似变形:

V W Z Y X U

ㄍ ㄝ ㄠ ㄡ ㄨ ㄩ

④无规则者:

Q ， ． ／ ； ′ 8 9 0 － ＝

ㄟ ㄓ ㄔ ㄕ ㄖ ㄗ ㄢ ㄣ ㄤ ㄥ ㄦ

⑤声调:

1 2 3 4

· ／ ∨ ＼

(3) 地方话拼音输入法：我国幅原辽阔，方言繁多，对于不会普通话的人，无法正确使用汉语拼音输入法和汉语注音输入法，因此产生了一些地方话拼音输入方案，例如，广东话拼音码，它是按广东话的字音采用 26 个英文字母对汉字编码的。

汉字字音输入码的编码规则简单易学，一般人都可掌握，使用广泛；编码本身需记忆的东西不多，只要懂得汉字的拼写规则，就可以根据字音输入编码；操作也很简单直观。

然而，由于汉字字量很大，致使同音字太多，汉字字音输入码的重码率较高。为了区分同音字，常采用人机对话式输入方法，根据提示区进行二次选择。当键入汉字字音输入码后，屏幕提示区显示同音字，操作人员从这些同音字中选择所需要的汉字。由于需要不断地在提示区中挑选汉字，有时甚至需要翻几页来查找，因此汉字输入速度慢，效率低。为了减少同音字，解决重码问题，提高输入效率，常采用下列办法：

(1) 除汉字的字音属性外，在汉字输入编码中增加字形、字义、字频、笔画数等其它一些属性；

(2) 采用汉字字音词组输入码。由于汉字书面语言是以词汇为单位的，而且同音词比同音字少得多，因此以词汇为单位输入汉语拼音，可大大减少字音输入码的重码率。

3. 汉字字义输入码

汉字字义输入码是用汉字字义编码的。英文输入码就是最常见的一种汉字字义输入码。它以英文单词作为汉字的编码，通过键入英文单词，便可输入中文词。例如，按英文输入码键入

I am Chinese

则可输入下述中文：

我是中国人

由于语言存在着岐义性问题，一个英文单词往往有多个中文解释，因此，当键入一个英文单词后，在提示区中就会出现若干

个与它同义的中文词，这就是所谓的同义词。例如，当键入Chinese后，提示区中会显示以下几个中文词:

中国人　华人　汉语　中文

4. 汉字字形字音字义混合输入码

汉字字形字音字义混合输入码使用汉字的多种属性来对汉字编码，例如，字形、字音、字义，字频等汉字属性。按汉字的多种属性编码，多半是为了克服单纯用字形编码出现的同形字问题，克服单纯用字音编码出现的同音字问题，克服单纯用字义编码出现的同义字问题，降低由此而产生的重码率，提高汉字输入效率。当然，这个问题也不能一概而论，解决同形字、同音字、同义字问题的关键主要取决于编码方案本身。

下面，扼要介绍几种汉字字形字音字义混合输入码。

①首尾首音码：取汉字的字首码（左上角的笔形）、字尾码(右下角的笔形）和首音码（汉语拼音第一个音符所对应的拼音字母，即首字母）来对汉字编码。

②快速输入码：它是用字首码、字尾码、声母、韵母组成的汉字输入码。这是首尾码加拼音码的盲打输入方法。

③拼音首尾码：先键入汉语拼音，如果提示区中仍找不到所需要的汉字，则再键入汉字首尾码的字首码。

④吉页码：取仓颉码的首码和尾码，再加上汉语拼音的首字母。

⑤声韵部形码：取汉字的声母、韵母、部首分类码（自然类15个，生物类20个，生理类25个，生活类38个，余类70个）和起笔码（汉字的部首以外部分的起笔形)。

⑥易码：每个汉字均对应三个码元，首码取汉字的汉语拼音的首字母，次码取汉字字形的首笔，尾码取汉字字形的末笔。首笔和末笔按横（一)、竖（|)、撇（丿)、点(丶)、折（乛）五种笔画编码。

⑦见字识码：这种编码是建立在汉字字形和字音基础上，以字形为主，汉字的偏旁、部首、单字一律做为字元看待。把汉字分解成字元串，每类字元赋予一个关系字，用关系字的汉语拼音的首字母来构成汉字的编码。例如，“亻”的关系为人，“丶”的关系字为捺。又例如，“韶”字由立、日、刀、口四个字元组成，而这四个字元恰恰均为关系字，这些关系字的语音分别是 Li, Ri, Dao, Kou，因此“韶”字的汉字输入码为 LRDK。

5. 其它汉字输入码

除上述按汉字字形、字音、字义编码的汉字输入码外，其它常见的汉字输入码还有：内码、国标码、区位码、电报码、电信码等。这些汉字输入码大多是用数字（十进制或十六进制）编码，而且是无重码类编码，因此可以盲打，适用于专业操作员。此外，对于用户新造的汉字，在未加入到其它汉字输入方案之前，一般均采用内码、国标码、区位码来输入它们。下面，扼要介绍这些汉字输入码。

(1) 内码：无论什么汉字输入码，输入电脑后都要转换成内码，以便电脑内部加工处理。内码一般采用十六进制数表示。

由于各种中文系统所采用的内码不同，因此没有统一的编码。我国多半采用带标识位的二字节内码，即国标码两个字节的高位均加 1（国标码加十六进制数 8080），如图 3.5 所示。台湾省采用的内码目前仍处在万“码”奔腾的局面，例如，“零”字的 BIG−15 码为 B973，IBM5550 码为 9DD9，通用码为 675B。为了提高中文软件的兼容性和可移植性，内码统一势在必行。

(2) 国标码是指 GB2312 规定的汉字交换码，均用两个字节表示，每个字节为七位二进制码，通常用十六进制数表示。如图 3.5 所示。

(3) 区位码是十进制数表示的国标码，因此亦称为国标区位码。GB2312 图形字形代码表纵向分为 94 个区，由第一字节标

识，横向将每个区又分为 94 个位置，由第二字节标识。因此，区位码表示汉字及其它图形字符在代码表中的位置。区位码转换为国标码的方法是：把区码与位码分别转换为十六进制数，合并后加十六进制数 2020。区位码、国标码、内码对照表如图 3.5 所示。

编码区域	区位码	国标码	内码
基本集	0101–9494	2121–7E7E	A1A1–FEFE
一级汉字	1601–5594	3021–577E	B0A1–D7FE
二级汉字	5601–8794	5821–777E	D8A1–F7FE
其它字符	0101–0994	2121–297E	A1A1–A9FE
保留区	1001–1594	2A21–2F7E	AAA1–AFFE
	8801–9494	7821–7E7E	F8A1–FEFE

图 3.5　区位码、国标码、内码对照表

(4) 电报码是我国邮电电报业务用的一种汉字编码方法。一个汉字用四个十进制数字编码，作为明码通信用。由于它是一国际标准，亦称为国际电报码。电报码基本上按汉字的部首顺序排列，每个汉字的编码从 0001 起比前面的汉字递增 1。

(5) 电信码是台湾省的电信明码，用四个十进制数字表示一个汉字，编码从 0001 到 9999，对应 9999 个汉字。

6. 汉字词组输入码

目前，几乎所有的汉字单字输入方案都发展了相应的词组输入方案，有些还把汉字单词和词组混合编码。汉字输入编码的研究正朝着以词组为单位输入汉字的方向发展。这是因为:

(1) 在当代以口语化的书面语言中，日益新滋长的不是单字，而是能用少数单字灵活组成的新词;

(2) 在字编码方案中，一个编码只能代表一个汉字，输入时只可逐字输入；而在词编码方案中，一个编码可以代表一个词组，即可以代表多个汉字，输入时既可逐字输入，又可逐词输入，因此，汉字词组输入方案可用较少的码元输入较多的汉字，从而大大地加快汉字输入速度，提高汉字输入效率；

(3) 由于同形词比同形字少得多，同音词比同音字少得多，同义词比同义字少得多，因此采用汉字词组输入码，可大大减少汉字输入码的重码率。

下面，扼要介绍三类分别按音、形、义编码的汉字词组输入码。

(1) 拼音词组输入码

下面例举几种拼音词组输入方案。

①二字词组的拼音码，采用两个字的全部拼音，例如，“中国”的拼音码为 Zhongguo。三字词组和四字词组的拼音码，取每个汉字的汉语拼音的首字母，拼成词组的拼音码，例如，“科学院”的拼音码为 kxy，“新加坡”的拼音码为 xjp。

②每条词组由三个代码组成，这三个代码分别是这条词组的第一、第二和最后一个汉字的汉语拼音的首字母。若词组只有两个汉字时，则三个代码由第一个汉字的拼音首字母和两个相同的第二个汉字的拼音首字母组成。例如，“中国科学院”的拼音码为 zgy，“新加坡”的拼音码为 xjp，“国家”的拼音码为 gjj。

③若词组小于或等于四个汉字，则取词组中的每个汉字的拼音首字母，即词组的代码最多四个；若词组大于四个汉字，则取词组中第一、第二、末前和末字的拼音首字母。例如，“新加坡”的拼音码为 xjp，“中国科学院”的拼音码为 zgxy。

④成语输入方案

成语大多数由四个汉字组成，取各个汉字的拼音首字母来输入成语。例如，“抛砖引玉”的拼音码为 pzyy。

⑤为了加快输入速度，常用字用一个或两个拼音字母输入。

例如，

q去　w为　r人　y要　u有　d的　g个　h和　j就

k可　l了　z在　x小　c出　b不　n能　m没

ba把　be被　da到　de得　di地　la来　li里　hu化　zh着

(2) 五笔字型词组输入码，其编码规则是：

①二字词：每字各取前两码，共四码。例如，“操作”的五笔字形代码为

扌　口　亻　𠂉

32　23　34　31

R　K　W　T

②三字词：前两字各取一码，最后一字取两码，共四码。例如“操作员”的五笔字形代码为

扌　亻　口　贝

32　34　23　25

R　W　K　M

③四字词：取每个字的第一码。例如，“汉字编码”的五笔字形代码为

氵　宀　纟　石

43　45　55　13

I　P　X　D

④多字词：取第一、二、三及末一个汉字的第一码，共四码。例如，“电子计算机”的五笔字型代码为

日　子　讠　木

22　52　41　14

J　B　Y　S

(3) 英文输入码是一种汉字词组输入码。这是因为，一个英文单词往往对应若干个中文词组，通过键入英文单词输入的是中

文词，而不是单字。

尽管同形词、同音词、同义词比同形字、同音字、同义字少得多，但汉字词组输入码仍存在着重码问题。例如，拼音词组输入码 bj 表示的同音词有：半截、半径、保健、北京、背景、笔记、比较、必将等。当键入汉字词组输入码后，相应的同形词、同音词、同义词出现在提示区，用户从中可选择自己所需要的词组。解决由同形词、同音词、同义词引起的重码问题，主要依靠汉字词组输入编码方案本身。

7. 汉字输入编码的评测

汉字输入编码方案已达上千种，新的编码方案正层出不穷，不同的人喜欢使用不同的输入编码，因此，目前强求统一输入方法还为时过早。然而，通过评价各种汉字输入编码，可以推动汉字输入技术的优化，促进汉字输入方法的逐步规范化和标准化。当然，最重要的还是通过实际使用，自然优选最佳方案，逐步趋于几种。

汉字输入编码的评测方法大致可分为两种：定量评测和定性评测。定量评测根据某些统计或计算可得到定量指标；而定性评测得到尚不便统计或计算的定性指标。下面仅简要介绍定量评测和定性评测的基本内容，以便对评价汉字输入编码方案的标准有一个初步的感性认识。

(1) 定量评测

①汉字字符集的容量：汉字字符集中所含汉字的个数。

②码元数量：在汉字输入编码中所用代码键和功能键的个数。

③码元熵值：根据码元数量和码元的使用频度，计算平均每个码元的信息量。

④汉字熵值：根据汉字个数和汉字的使用频度，计算一个汉字平均所用的最低信息量。

⑤平均码长：根据汉字的频度分布，计算平均每个汉字所用的码元个数。

⑥汉字编码效率：理论码长的下限（即汉字熵值）与实际平均码长之比。

⑦键入速率：单位时间内击键输入汉字的字数，通常用字数／分来表示。

⑧重码数：重码数等于重码字总数（即总共有多少个重码字）减去重码组数（即有多少组重码字）。

⑨重码率：根据重码组中重码字的使用频度计算重码字的出现率。

⑩非常规代码数：有时为了区别过多的重码字或按基本规则无法给出某些汉字的代码时，不得不附加某些规则或特殊定义才可，这样的汉字代码称为非常规代码。非常规代码指按附加规则或特殊定义方式进行编码的字码数。

⑪非常规代码出现率：根据非常规代码数和非常规代码的使用频度，计算非常规代码出现率。

⑫多码数：一个汉字与多个代码相对应的数量（简码也按多码计算），称为多码数。它等于总代码数减去编码的总字数再加上重码数。

⑬学习曲线：在操作员学习编码的过程中，根据实测结果绘制的键入速度和错码率相对于时间变化的一组曲线。错码率是错码次数占总字数的百分比，漏码或多余字码亦按错码处理。

⑭编码学习期：操作员从学习编码开始到不查码本输入编码，错码率稳定下降至2%且键入速度达8个字符／分以上所需要的最短时间，常以小时为单位。

⑮编码熟练期：操作员键入速度达35个字符／分以上且错码率不高于1.5%所需要的最短时间，常以小时为单位。

⑯集外字符：汉字编码字符集以外的字符。

(2) 定性评测

①编码方案的论证是否合理和充分，是否具有独创性。

②在编码规则方面：编码规则的数量多少；编码和拆字规则的逻辑性和规律性是否简明扼要，是否前后一贯；编码规则对用户要求的高低等。

③在记忆量大小方面：编码规则是否容易记忆；非常规编码的字数和规律性；附加规则或规定是否繁琐等。

④在检索和兼容方面：对于有疑难的集内字是否有检索的手段及其难易程度；编码的兼容性及用户进行选择的自由度；对集外字有无处理能力及处理方法的难易等。

⑤在与机器的联系方面：编码的设计是否考虑了与计算机的特点相结合；可使用的计算机类型，是否需要增加其它设备或采取其它措施；最大码长；软件程序所占存储量的大小；是否需要人机相互作用及人机相互依赖的程度等。

3.1.3 智能化汉字输入

智能化汉字输入，是指用电脑的智能辅助汉字输入，它是编码式汉字输入方法的一种发展。

从某种意义上讲，汉字语音输入和汉字字形输入是智能化汉字输入的高级阶段。然而，这两种输入方式离实用化还有一段距离。我们这里涉及的智能化汉字输入是指汉字键盘的智能化汉字输入。本节仅介绍智能化汉字输入技术的四种功能：联想功能、翻译功能、自学习功能和模糊码功能。

智能化汉字输入技术是独立于各种汉字输入编码方案的，也就是说，它借助于电脑可以辅助任何一种现有的汉字输入编码方案，也可以用于不同于现有方案的任何一种新的汉字输入编码方案。它容易掌握，不需要专门训练，而且能以较快的速度输入汉字，从而提高了汉字输入效率。

1. 联想式汉字输入方法

在汉语中，汉字在大多数场合下都不是孤立存在的。从汉语的上下文关系看，以一个汉字为词头的词，多的有几十个，少的也有几个。例如，以“新”字开头的词有新年、新奇、新生、新式、新书、新闻、新鲜、新兴、新型、新颖、新旧等。这就是说，一个汉字后面可以跟着别的汉字而构造出许多词。联想式汉字输入方法正是利用这一特性来模拟人的联想能力。用户可根据自己的专业范围，预先在电脑中构造汉字的联想结构，以后每当输入一个汉字或词后，电脑会根据上下文信息的相关性，自动地联想出与它相关的若干个字或词，并把它们显示在提示区，例如，每当输入“新”字后，提示区中便会出现下列汉字：年、奇、生、式、书、闻、鲜、兴、型、颖、旧等。用户可直接从中选取所需要的字或词，而无需键入编码了。我们把由一个汉字或词联想出的与它相关的汉字单字称作联想字；把由一个汉字或词联想出的与它相关的两个或两个以上汉字（一个词或词的一部分）称作联想词。例如，“新”字的联想字有：年、奇、生、式、书、闻、鲜、兴、型、颖、旧等；“新”字的联想词有：陈代谢、加坡等，它们是词的一部分。由此可见，联想词并不一定是词，这样定义是为了叙述和处理上的方便。

这样，从一个字或一个词可以引出一组联想字或联想词，其中每个字或词又可以引出另一组联想字或联想词。通过逐级联想，可以产生大量词组、短语甚至句子。图 3.6 给出了联想结构的一个例子。除由“新”字引出的联想字和联想词之外，还可由“新加坡”联想出国立大学、中文与东方语言信息处理学会、国家图书馆、国家电脑局等联想词，从“国立大学”又可继续联想出信息系统与电脑科学系、物理系等联想词，以此类推，可形成联想词的关系链。这样，从“新”字便可联想出“新加坡国立大学信息系统与电脑科学系”这一词组。

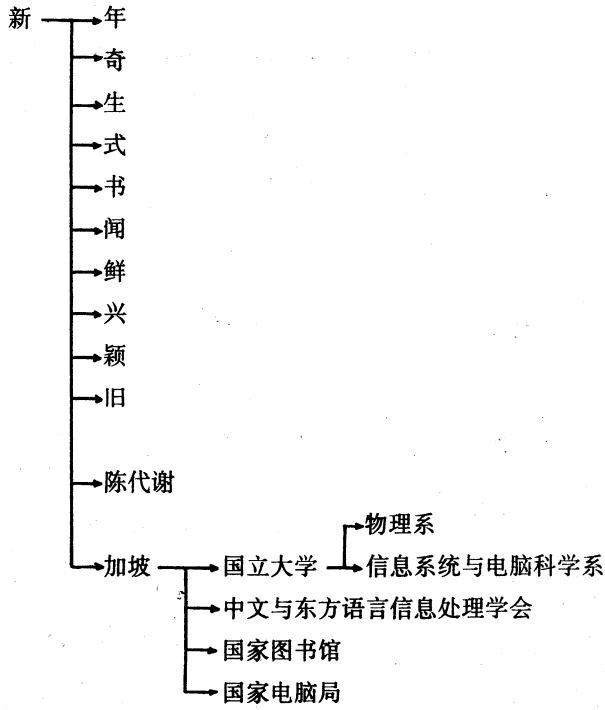


图 3.6　联想结构

实际上，只要编排适当，联想字也可以形成关系链，只是形成多级关系链较为困难些。图 3.7 给出了联想字多级关系链的例子。

联想式汉字输入方法支持各种汉字输入编码文案，可以通过编码式汉字输入方法和联想式汉字输入方法的配合来输入汉字。编码式汉字输入方法往往需要按几次键才能输入一个汉字或词，而联想式汉字输入方法输入一个联想字或联想词只需要按一次键即可。因此，要尽量采用联想式方法输入汉字，实在在提示区中找不到需要的联想字或联想词时再用编码式方法来输入。

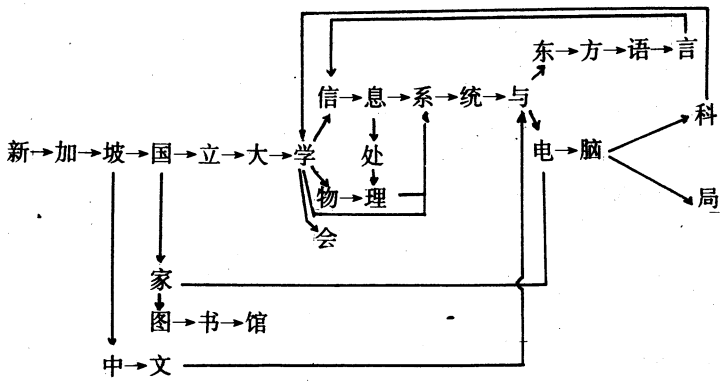


图 3.7　联想字的多级关系链

由于从一个字或词可以联想出多个联想字和联想词，而提示区一次能够显示的联想字和联想词的个数有限，因此，当联想字和联想词较多时，需要采用翻页的方法去查看和选择所需要的字或词。这势必会影响输入速度，与联想式汉字输入方法的宗旨相违背。为了解决这一矛盾，要权衡究竟应收入多少个联想字和联想词，关键取决于联想结构的编排和组织。

在联想结构中究竟应收入哪些联想字和联想词，这与它的使用场合有着密切的关系。用户可根据自己的专业范围和使用环境，随意构造和随时修改联想结构的内容。例如，用户可把产品名称、单位名称、人名、地名等经常使用的词组加入联想结构，这犹如打字员把常用词组所含的汉字放在一起一样。如果联想结构编排得好，那么基本上不必考虑汉字的编码，便可迅速地输入汉字。

联想式汉字输入方法的实现，随中文系统的不同而不同。下面，我们以联想式汉字系统为例来说明如何实现联想式汉字输入方法。

联想式汉字系统是通过联想字词典和联想词组词典来实现联

想式汉字输入方法的。为了叙述方便起见，我们把联想字词典和联想词组词典统称为联想词典。

(1) 联想词典源文件的格式：简言之，联想词典源文件就是字或词与它们相关的联想字或联想词的对照表。

①联想字词典源文件的格式：源文件每行左边为联想引导词(由一个或多个汉字组成)，右边是该词的若干个联想字。引导词与联想字之间空一格，联想字之间不间隔。

例如，

新　年奇生式书闻鲜兴型颖……

时　间候刻差代期事钟常光……

控制　论流机……

②联想词组词典源文件的格式：源文件每行左边为联想引导词（由一个或多个汉字组成)，右边跟着它的一串联想词，引导词与联想词之间、各联想词之间均空一格。

例如，

新　陈代谢　加坡　……

新加坡　国立大学　中文与东方语言信息处理学会　国家图书馆　国家电脑局　……

国立大学　信息系统与电脑科学系　物理系　……

中国　科学院　人民　银行　……

汉　字信息处理系统　语拼音　字输入方案　……

(2) 联想词典的自动生成：步骤如下：

①用中文字处理软件或编辑程序，按规定格式编辑联想词典源文件。

②调用通用的词典生成程序，把联想词典源文件转换为目标文件，常把联想词典目标文件简称为联想词典。

③把联想词典换名为系统认可的联想词典名。

④每当启动时，联想词典自动装入到系统中。

(3) 联想词典的修改：修改系统中已有的联想词典，可采用

以下两种方法:

①现场修改联想词典: 这一方法适于在使用现场随时对联想词典进行局部修改。按某一规定的功能键 (例如, Alt-0), 系统自动调用词典修改程序, 进入修改联想词典状态。如果要增加或减少一个联想字或联想词, 首先要输入这个联想字或联想词的引导词, 然后再输入这个联想字或联想词。在输入引导词和联想字或联想词的过程中, 可切换汉字输入方案, 用常驻的任何汉字输入方案输入汉字, 但不能输入 ASCII 字符。

②修改联想词典源文件, 重新生成联想词典。

这一方法通常用在需要大量修改的场合。用中文字处理软件或编辑程序修改联想词典源文件, 再调用通用的词典生成程序重新生成联想词典。

如果联想词典源文件留有备份, 则可直接在源文件基础上进行修改; 否则, 必须首先调用系统提供的通用词典还原程序, 把要修改的联想词典还原为源文件, 然后再去修改源文件。联想词典还原为源文件的步骤是: 首先, 键入要还原的联想词典的目标文件名; 然后, 要么显示还原后的源文件, 要么键入还原后的源文件名, 将还原后的源文件存入磁盘。

2. 翻译式汉字输入方法

现存的汉字输入编码方案, 要么重码太多, 输入速度慢, 要么规则复杂, 不易普及。针对上述问题, 出现了一种翻译式汉字输入方法。这种汉字输入方法吸取机器翻译的设计原理, 事先在系统中建立了含有常用词汇知识库和惯用语法规则。当用户输入汉字时, 例如, 用拼音码输入汉字, 不必在一大堆同音字或同音词中选择所需要的汉字或词, 只要不停地输入汉语拼音, 系统便会自动地帮你挑选适当的同音字或同音词, 显示在屏幕正文区上, 并可不断地更正, 从而迅速地完成一个句子或一篇文章的输入。

如果要修改输入的内容，用户可随时将光标调至错字或词的位置上，按动规定的功能键，屏幕提示区上便会显示所有的同音字或同音词，供改正错误。

此外，有的翻译式汉字输入方法还允许用户自建字典或词典，并可通过鼠标器代替键盘输入，而且还能处理破音字、变调字和同码异形字。

3. 自学习功能

无论是编码式汉字输入方法，还是联想式汉字输入方法，都可有自学习功能。这是一种自适应的输入方法。采用自适应的输入方法，不仅可以动态调整提示区中字和词的显示顺序，而且可以直接输入前面曾输入过的字或词。

(1) 编码式汉字输入方法的自学习功能：各种汉字输入方案，无论是字编码输入方案，还是词编码输入方案，都可有自学习功能。每当输入一个汉字或词，中文系统便统计它的使用次数，提示区中的同音字或同音词、同形字或同形词，同义字或同义词的显示顺序会按使用频度自动调整，使用频度较高的字或词自动排到提示区的前面，从而达到常用先见的效果，帮助用户迅速发现使用频度较高的字或词；或使频度较高的字或词直接在屏幕正文区上显示出来，只当不是所需要的字或词时，才从提示区中选择，代替正文区中原来显示的字或词。由于这种动态排序的功能，再加上用户可在使用现场随时修改编码词典，使中文系统中的汉字输入方案随着时间的推移变得越来越适合用户的需要，因此也越来越顺手好用。

(2) 联想式汉字输入方法的自学习功能：中文系统可以根据实际使用情况动态调整提示区中联想字或联想词的排列顺序，用过的联想字或联想词会自动移到提示区的前面。这样，经过一段时间，中文系统会根据联想字或联想词的使用频度，把最常用的联想字或联想词集中到提示区的最前面，不常用的联想字或联想

词会逐渐移到提示区的后面，达到常用先见的效果。

除了用户可以在使用现场随时修改联想词典外，联想词典中的联想结构本身也可以在使用过程中，根据对实用信息的统计分析，自动地进行修改，使中文系统在一定程度上具有自组织能力。

(3) 抽屉式汉字输入方法：除了词组等汉语的固有结构外，对于讲述某一具体问题的具体文章，有关的字和词出现次数往往较高，而对于不同的作者，个人惯用的字和词也不尽相同。考虑每个人和每篇文章的特殊性，除了把使用频度较高的字或词放在提示区的前面外，还可以把正文区中前面曾经输入过的字和词直接放到当前要输入字或词的位置上。例如，用光笔点到正文区中的一个字或词，就可以在正文区当前光标所在位置显示出这个字或词。这种汉字输入方法在正文区中查找前面曾经输入过的字和词，就象在抽屉里查找东西一样，因此亦称为抽屉式汉字输入方法。

4. 模糊码功能

模糊码功能支持各种汉字输入方案。正如操作系统中文件名可用"?"和"*"等当作通用文件名字符一样，汉字输入码也可用"?"和"*"等当作通用输入码字符，亦称为模糊码元。在汉字输入码中，"?"代表码元集合中任一码元，例如，可代表任一字母、数字或其它特殊字符，究竟可以代表哪些字符，取决于汉字输入方案的码元集；"*"代表任意多个"?"，最多至最大码长。

例如，假定在英文输入方案中具有模糊码功能。键入"Comput?r"，可在提示区中显示"计算机"、"电脑"等词。键入"Comput*"相当于键入"Comput?"、"Comput??"等，也就相当于键入 Compute，Computer 等英文单词，因此提示区中会出现"计算"、"计算机"、"电脑"等词。

又例如，假定在仓颉输入方案中具有模糊码功能。键入"日

”，提示区中列出首码是“日”的字：巴、日、曰、旦、早、即、旱、旺等。键入“日月”，提示区中列出首码为“日”、第二码为“月”的字：明、冒、晖、晕、暖、盟、暝、暧等。

从上面可以看出，模糊码功能方便用户直接找寻音、形、义相似的汉字或词，特别是，当不能确切记得一个字或词的输入编码时，模糊码功能更为有用。

在有的中文系统中，通过事先按一个规定的功能键，系统便进入模糊状态。在模糊状态下，系统在键入任何码元后面都附加一个“*”。例如，键入“1”相当于“1*”，键入“12”相当于“12*”等等。对于某些汉字输入方案，这种方式可以使用户随着键入码元的过程，在提示区中随时看到模糊码所匹配的字和词，一旦发现其中有所需要的字或词，就可以立即选用，而不必键完这个字或词的输入编码的全部码元。

3.2 汉字输出

汉字输出是指输出汉字字形。本节介绍汉字字体、点阵、字模等与字形有关的概念。

汉字字形又是从汉字字库中产生的。本节介绍几种存取汉字字库的方法。

汉字输出主要指显示输出和打印输出。本节分别介绍汉字显示输出和汉字打印输出的若干问题。尤其是着重总结用打印控制命令实现的各种汉字打印功能。

3.2.1 汉字字形

字形（Font）在排版中原本是具有一定尺寸和形式的一组铅字。在电脑中沿用这一词，表示字的形状和大小。

下面，介绍与汉字字形有关的字体、点阵、字模等概念。

1. 字体 (Character Style 或 typeface)

字体表示在书写形式上具有相同风格和特点的一组字。同一种文字具有多种不同的字体，它与字的尺寸大小无关。

(1) 汉字字体：按书写形式不同，可把汉字分为不同的字体。每种字体各包含一个字体族。每个字体族由一种基本字体(正体或方体) 和若干种由基本字体变化而来的美术字体组成。

基本字体有宋体、仿宋体、楷书体、黑体、圆体、明体、隶书体、行书体、草书体、单线体等。例如，

宋体 仿宋体 楷体 **黑体**

美术字体包括长体、扁体 (平体)、粗体、瘦体 (细体)、斜体、粗框体、方框体、中空体、立体、阴影体、反白体、方点体、横条纹体、直条纹体等，以及由上述字体组合成的字体。例如，图 3.8 例举一些美术字体。

(2) 简体、繁体、异体、旧体：按笔画写法不同，可把汉字分为简体、繁体、异体和旧体。常用的是简体和繁体。关于简体字和繁体字的情况，请详见第四章。

(3) ASCII 字符的字体：分为半形字符和全形字符两种。无论是半形字符，还是全形字符，目前的操作系统越来越趋向于设置多种 ASCII 字符的字体。

ASCII 字符的字体包括 NORMAL (STANDARD)，BOLD，ITALIC，ROMAN，GREEK，ORATOR，PSTANDRD，HELVETICA，GOTHIC，SMALL，UNDERLINE，SCRIPT，FOREIGN，MATH，LINEDRAW，SYMBOL、OUTLINE，STRIKEOUT，SWISS 等，以及由这些字体组合而成的字体。例如，图 3.9 例举了 ASCII 字符的几种字体。

2. 汉字字形的数字化表示

尽管汉字字形千变万化，错综复杂，但由于汉字是方块字，每个汉字都同样大小，因此，无论汉字笔画多少，总可以把每个

图 3.8　美术字体

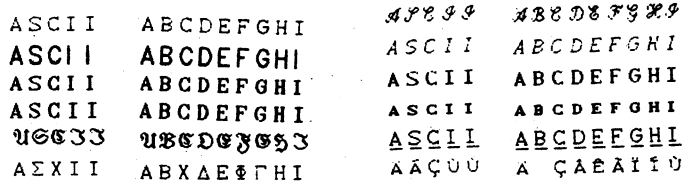


图 3.9　ASCII 字符的字体

汉字放在同样大小的方块中，从而可以把这一方块看作是由点组成的一个矩阵，简称为点阵（Dot Matrix）。例如，对于 16×16 点阵的汉字；每个方块字横向竖向各 16 个点，共计 256 个点。

点阵中每个点可以是黑白两种颜色之一，如果黑点表示组成汉字的最小单位——位点，那么白点表示汉字的背景，反之亦然。这样用点阵描绘出的汉字形称为汉字点阵字型。

在电脑中，通常用二进制数字来表示点阵。如果用二进制数1表示黑点，用二进制数0表示白点，则一个16×16点阵的汉字字形就可以用一串二进制数（256位二进制数）来表示。这种方法称为点阵的数字化。例如，图3.10中“点”字的点阵就可以用256位二进制数表示，由左向右，从上到下逐点记录，从而形成一个二进制数字串。

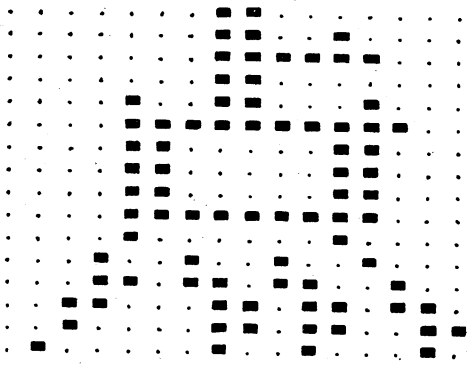


图3.10 “点”字的16×16点阵

由于二进制数写起来太长，使用不方便，故常采用八进制数或十六进制数来代替二进制数。在点阵中，并列的四个点可以用一位十六进制数来表示。因此，用十六进制数表示一个16×16点阵的汉字，只需要64个十六进制数字即可。例如，图3.10中“点”字的点阵数字化信息的二进制数表示和十六进制数表示如下：

行序号	二进制数表示	十六进制数表示
1	0000000110000000	01 80
2	0000000110010000	01 90
3	0000000111111000	01 F8
4	0000000110000000	01 80
5	0000100110001000	09 88
6	0000111111111100	0F FC
7	0000110000011000	0C 18
8	0000110000011000	3C 18
9	0000110000011000	0C 18
10	0000111111111000	0F F8
11	0000100000010000	08 10
12	0001001001001000	12 48
13	0001101101100100	1B 64
14	0011000110110110	31 B6
15	0010000110110011	21 B3
16	0100000100100010	41 22

汉字字形经过点阵的数字化后而转换成的一串数字，称为汉字的数字化信息，或简称为字形信息。

为了与电脑普遍采用八位二进制信息为一个字节这样的规定相符合，汉字字形信息多半采用16×16、24×24这样一些8的倍数点阵。一个16×16点阵的汉字含有256个点，需要用32个字节表示；而一个24×24点阵的汉字含有576个点，需要用72个字节表示，其信息量显然比16×16点阵的信息量大2.25倍。

从实际应用看，16×16点阵是最简单的汉字字形点阵，除少数笔画特别复杂的汉字需要做些变通外，基本上能表示GB2312中所有简体字的字形。24×24点阵则可以表示出宋体、

仿宋体、楷书体、黑体等多种字体的汉字。例中，“点”字的 24×24 点阵如图 3.11 所示。

顺便提一下，在 16×16 点阵中，一般总是在点阵的左侧或右侧留出一列空白。总信息量仍为 16×16，但在侧边上有一列信息全为 0。这是因为，16×16 点阵一般用于显示，汉字显示时每个字之间需要有一定的字间距。而行间距则由输出设备用软件或硬件方法添加。

3. 字模

字模在印刷中原本是浇铸铅字用的模型。在电脑中沿用这一词，表示用于产生字形的点阵模式。在电脑中，字模与字形的概念往往没有严格的区别，因此常常混为一谈。

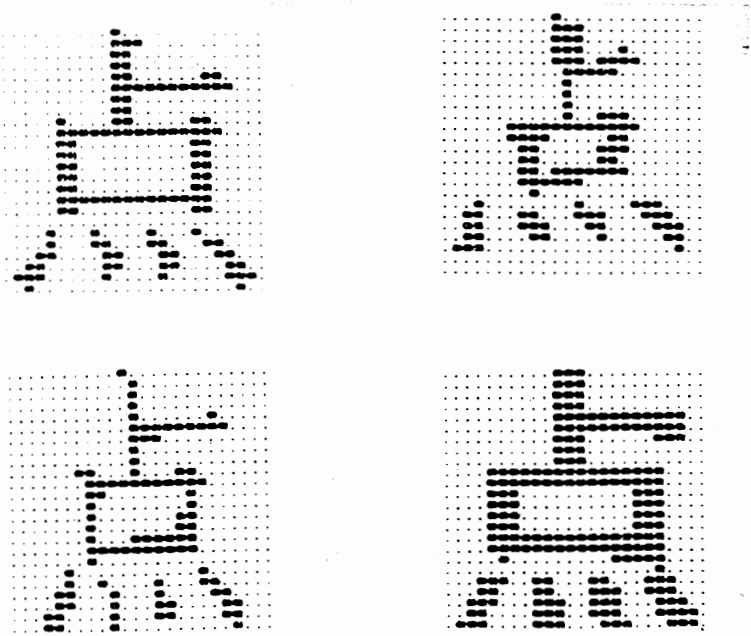


图 3.11 “点”字的 24×24 点阵

字模按构成字模的字体和点阵可分为不同的类别。汉字字模按构成字模的字体可分为宋体字模、仿宋体字模、楷体字模、黑体字模等。构成字模的字体一般均是基本字体，汉字字形是在这些字模基础上放大、缩小、反向、旋转、加背景及其它变形而来的，例如，美术字体。汉字字模按构成字模的点阵大小可分为16×16点阵字模、24×24点阵字模、32×32点阵字模、48×48点阵字模、64×64点阵字模、96×96点阵字模等。汉字字形的点阵位点数目越多，则输出的汉字清晰度越高。ASCII字符也可按字体和点阵的不同而具有不同的字模。

中文系统常把具有不同点阵和不同字体的字模编号，称为字号。例如，1号字，2号字，……。字模以二进制文件的形式存入在存储器中，构成字模库。

3.2.2 汉字字库

汉字字形数字化后，以二进制文件的形式存放到存储器，构成汉字字模库。汉字字模库亦称为汉字字形库，简称为汉字字库。

汉字字库向各种输出设备提供字形信息。有些字形信息可直接提供给输出设备使用，有些字形信息则需要进行必要的加工处理才能使用。另外，各种不同的输出设备对提供字形信息的速度要求也不一样。因此，实际上所采用的汉字字库的构造方法是多种多样的。

下面，介绍三种汉字字库的构造方法。

1. 直接存取的汉字字库

直接存取的汉字字库是指直接从主机基本内存存取字形信息。内存在五、六十年代大多采用磁心作为存储元件，到了七十年代后，随着大规模集成电路存储器芯片的发展，现在大多采用

随机存取存储器 RAM（Random Access Memory）或只读存储器 ROM（Read Only Memory）。例如，CCDOS1.0 版的 16×16 点阵字库就是直接存取的汉字字库，当系统启动时，汉字字库装入到事先申请好的内存空间中去。

2. 多级存储的汉字字库

(1) ROM／RAM 和磁盘：汉字字库占存储量很大，仅 16×16 点阵字库，按 87 个区计算，就占存储量 261696 个字节(约 256KB)。而 24×24 点阵字库，按 87 个区计算，占存储量 588816 个字节（约 576kB)。然而，根据汉字使用频度的统计，一级汉字 3755 个字已占使用汉字的 99.9%以上，而二级汉字的使用频度很低，因此可采用多级存储方案，将一级汉字存放在 ROM 或 RAM 存储器，而二级汉字存放在磁盘上。这样的存储方案既解决了占内存的问题，又不大影响存取速度。

至于非汉字图形字符亦要根据使用频度决定是放在 ROM 或 RAM 中，还是放在磁盘中，例如，有的中文系统把 1–3 区的间隔符、标点符号、运算符、单位符号、序号、数字、字母和第 9 区的制表符放于 ROM 或 RAM 中，而其它区字符放于磁盘中。

有的中文操作系统把这种汉字字库分级存储的概念交给用户，由用户根据不同的需要来选择不同的分级存储方案。例如，CCDOS 4.0 给出字库驻留选择：

①驻留一级字库：1–55 区常驻内存，其它区放在磁盘。

②驻留全部字库：把一级字库和二级字库全部驻留内存。当用户对内存容量要求不高或内存容量较大但要求输入响应速度快时，使用这一选择。

③不驻留：一级字库和二级字库全部放在磁盘。当用户对内存容量要求较高或内存容量不大但要运行大的应用程序时，使用这一选择。

④任意：用户选择一个汉字作为分级存储的界限，把前一半字库放于内存中，后一半字库放于磁盘中。中文系统自动采用先内存后磁盘的方式读取所需的汉字。

(2) 基本内存、EMS、Extend、汉卡和磁盘：汉字字库的多级存储还不限于基本内存和磁盘，也可以将字库装入 640K DOS 基本内存之外的 EMS 存储，以及 80286 或 80386 机的 1M 以上存储器，甚至可安装汉卡来存放汉字字库。

实质上，用于快速存取汉字字库的内存储器有四种：汉卡(字库卡)、EMS 扩存、Extend 扩存和主机基本内存。为了合理地使用系统中的存储资源来存放汉字字库，提高系统的运行效率，有的中文操作系统（例如，UCDOS）按照上述优先次序进行字库存储分配。

基本内存是指 DOS 直接管理和应用软件直接使用的地址空间，它从 0 地址开始，最大可到 640K。这部分存储空间是绝大部分软件的唯一运行空间，是比较宝贵的资源，因此汉字字库不宜占用基本内存。

EMS 扩展内存是一种以页映射方式存取的存储系统，它最初是为了增加 XT 机的寻址量而设计的。它只在 1M CPU 空间内开设一个窗口（64K），然后由软件可将这个窗口切换到更大的物理存储器的不同部分，但在同一时刻程序设计人员所能见到的只是窗口所对应的那一部分。用这种方式，可将 XT 的存储容量扩充到 8M 之多。由于 EMS 只为较少数的软件作辅助存储用，因此在安装汉字字库时，应首先考虑使用 EMS 扩存。EMS 存储器是通过 Config. sys 文件中指定的设备驱动程序来检测安装的。在 AST 286 机上，必须在 Config. sys 文件中加上 device＝REMM. SYS 命令才能安装上 EMS。高版本的 DOS 已可直接设置 EMS。

Extend 扩展内存是指系统中地址在 1M 以上的内存，只有以 80286、80386 为 CPU 的微型电脑（例如，AT，PS／2，

286，386 等）才能有这种存储器。在 DOS 环境下，1M 以上的存储不能由软件直接使用，因此应优先考虑用它安装汉字字库。然而，有一个值得注意的问题：有些软件（例如，AUTOCAD）可能会使用 Extend 扩存，这样就会与中文系统发生冲突。其结果，会导致显示和打印的汉字出现字形缺损的现象。解决冲突有两种途径：一是控制应用软件，不让它使用 Extend 扩存，或者进行人工地址分配，让应用软件只使用某个地址段，而给中文系统让出部分空间；二是禁止中文系统使用 Extend 扩存，或者指定中文系统可用 Extend 存储的地址。

用户根据具体的环境和不同的需要可选择不同的分级存储方案。例如，为了实现高速打印，可把 24×24 点阵汉字字库分割为四个部分：第一部分装入 EMS，第二部分装入 Extend，第三部分装入基本内存，最后一部分留在磁盘上。系统按照上面的优先次序自动进行分配，前面两种存储器（EMS 和 Extend）不足以装下整个字库时，系统报告字库总字节数、装入 EMS 的字节数、装入 Extend 的字节数和剩余的字节数。此时，用户可选择将剩余字节中的多少 KB 装入基本内存，而将剩余的留在磁盘上。这个值可根据当前基本内存的剩余量和应用软件的内存要求而定。当然，如果只打印汉字文本文件而不运行应用软件，则可在基本内存中装入较多字节数。

3. 多用户共享的汉字字库

在对汉字字库存取速度要求不高的情况下，为了有效地发挥汉字字库存储器的作用，降低硬件成本，可采用多个用户共享的汉字字库，即一个汉字字库可以同时为若干个终端或若干个设备所使用。

3.2.3 汉字显示

汉字显示是在中文环境下实现的。一般的中文系统均设置两

种环境。通过按转换键可在中文环境和西文环境之间来回切换。所谓西文环境，实质上就是纯西文环境。所谓中文环境就是中西文兼容的环境。两者的主要区别是：在西文环境下，每个字符编码为8位，占一个字节，只能运行西文软件，适合运行未经汉化的西文软件，ASCII字符均可正确地显示或打印出来，当然，在中文环境下输入的汉字在西文环境下就会变成两个西文图形字符。在中文环境下，每个汉字等全形字符编码为两个8位，占两个字节；而ASCII字符等半形字符编码为一个8位，占一个字节。全形字符的显示和打印宽度均是半形字符的二倍。中西文环境的切换，实质上就是切换键盘输入模块、显示模块和打印模块。注意：在汉化软件执行过程中切换中西文环境，其结果不定。

各种中文系统的中文显示环境不同，主要体现在屏幕的显示行数和列数不同。这是由显示器的分辨率、显示适配器、显示字形的点阵数、显示方式（字符显示方式或图形显示方式）、正文区与提示区的分布等因素决定的。例如，在图形显示方式下，VGA采用16×16点阵，可以显示25行正文和1行提示，保持西文显示行数。EGA采用11×16点阵（16×16点阵的压缩形式），也可以显示25行正文和1行提示，保持西文显示行数；若采用16×16点阵，则只能显示19行正文和1行提示。CGA采用16×16点阵，只能显示10行正文和1行提示。又例如，长城0520C使用了分辨率为640×450并且具有8种颜色的彩色／图形适配器，配以高分辨率彩色显示器，实现了每屏25行每行40个汉字的显示能力，其中25行正文，3行提示，从而保持了中西文显示环境的一致性。

为了适应不同显示器的需要，有的中文操作系统提供几种版本，中文软件也随之具有不同的版本。也有的中文操作系统提供几种不同的中文显示环境，供用户在安装时根据自己现有的显示器选择所需要的显示环境。例如，CCDOS 4.0对于单色显示器

和高分辨率彩色显示器（640×400）显示25行汉字，对于一般彩色显示器（640×200）显示11行汉字。即使对于同一种显示器，中文操作系统也往往提供几种不同的中文显示环境。用户可随时在各种显示环境之间切换，以适应各种不同的需要。例如，联想式汉字系统具有28行、15行等几种显示格式，用户可通过屏幕格式选择项挑选或改变屏幕显示格式。

汉字显示具有两种显示方式：字符显示方式和图形显示方式。采用图形显示方式显示汉字的中文系统，可在一个显示器上同时显示文本和图形。采用字符显示方式显示汉字的中文系统，虽然可在一个显示器上同时显示文本和图形，但作图时需要按转换键或键入命令从文本工作方式切换为图形工作方式，作图完毕再按转换键或键入命令从图形工作方式退回到文本工作方式；或除在一个显示器上以字符显示方式显示文本外，另接一个带有彩色／图形适配器的显示器显示图形。

中文系统在图形显示方式下具有图形加工能力。图形加工具有下列功能：

(1) 清除图形：清除屏幕上的图形。

(2) 图形存盘：把屏幕上显示的图形以图形文件形式存入磁盘。

(3) 图形读盘：把磁盘中的图形文件调出，并显示在屏幕上。

(4) 打印图形：把屏幕上显示的图形打印出来。

对于彩色显示器，一般中文系统都具有设置和选择屏幕颜色的功能，包括背景颜色和前景颜色（字符或图形的颜色）。

3.2.4 汉字打印

汉字打印是指汉字与西文字符的混合打印，即中西文兼容的打印。

汉字打印功能是用控制键或打印控制命令来实现的。打印控

制命令分为基本命令和宏命令两类。基本命令基本上与打印机控制命令相对应。宏命令是由多个基本命令组成的更高级的打印控制命令，完成较复杂的打印功能。用户可自定义宏命令，从而摆脱用户总是被打印机控制命令牵着走的被动局面。

打印控制命令的格式一般如下：

〈引导符〉〈标识符〉〈参数表〉

〈引导符〉有的采用 ESC (IBH 或 27)，有的采用 ASCII 字符或 ASCII 字符串，亦可由用户自行定义，甚至可随时更换引导符。〈引导符〉的定义最好是书写方便，而且容易被打印驱动程序所接受；更重要的是，不能与经常用于打印的 ASCII 字符相混淆。例如，可采用～或、等作为引导符。

〈标识符〉为 ASCII 字符或 ASCII 字符串。对于基本命令，它亦称为控制符。对于宏命令，它亦称为宏标识符。

〈参数表〉::=〈参数〉|〈参数表〉〈间隔符〉〈参数〉

〈参数〉一般有两种形式：一种是二进制代码形式，若参数值＜256，则可用单字节表示，若参数值＞256，则可用双字节表示，一般高位字节在后；另一种是十进制数字串。显然，用第二种形式提供参数较为方便。

由于参数有两种形式，因此打印控制命令也具有两种格式，一般由不同的〈标识符〉来区分具有相同功能的两种不同格式。

下面，较详细地概括和总结汉字打印功能。

1. 打印方式

打印方式包括文本方式和图形方式。

文本方式使用原 BIOS 中的打印驱动模块，将 ASCII 字符直接传送给打印机，用打印机自身的字库打印，因此打印速度快，打印头使用寿命也长。

图形方式使用汉字打印驱动模块，从汉字字库中选取字模，把汉字字形当作图形打印出来。

根据不同的需要，可按照图形方式和文本方式的不同组合，选择下列各种打印方式:

(1) 中文方式（汉字方式）

中英文均以图形方式打印，速度较慢，但打印出的字体优美，在文件中汉字占较大份量时可选用中文方式打印。

(2) 英文方式(ASCII 方式)

中英文均以文本方式打印。英文方式只适于打印英文文件，速度较快。若文件中含有汉字，则会打印出不沦不类的符号。

(3) 中英文混合方式

中文以图形方式打印，英文以文本方式打印。由打印驱动程序控制两种方式的转换，速度取决于中英文所占比例的多少。

(4) 中文打印机方式

中文打印机就是带汉字字库的打印机，可直接使用中文打印机打印中英文，打印速度快。但由于汉字字库中往往只有一种字模，无法进行字形变换和多字体打印，也无法打印用户造的字。如果需要字形变换或打印用户造的字时，必须转回使用中文系统本身提供的汉字字库。

2. 字形变换

字形变换有下列几项内容:

(1) 汉字、全形字、半形字三者之间的变换

汉字及全形字与半形字的宽度比例一般默认为 2，有的中文系统允许调整宽度比例值。

(2) 汉字字模之间的变换

不同点阵或不同字体的字模之间的变换。

(3) 字形放大或缩小

字形按比例横向（水平）或按比例纵向（垂直）放大若干倍或缩小若干倍。

鉴于放大后的汉字出现锯齿状（特别是放大倍数较大时尤为

明显)，有人用算法将它作平滑处理。平滑放大的原理是：用补偿和消去一些边界点的办法来弥补放大后字形边缘的锯齿状。通过平滑处理，能产生高质量的汉字放大字形。

(4) 字形旋转与颠倒

图 3.12 例举了汉字字形的若干种旋转角度和颠例方向。

图 3.12　字形和旋转与颠倒(之一)

图 3.12 中的字形是按逆时针方向旋转，亦可按顺时针方向旋转。显然，旋转 0 度为横印，旋转 90 度为直印（竖排版）。

由于半形字一般不跟着汉字旋转，因此要使英文字母、数字等 ASCII 字符旋转打印，必须使用全形字。

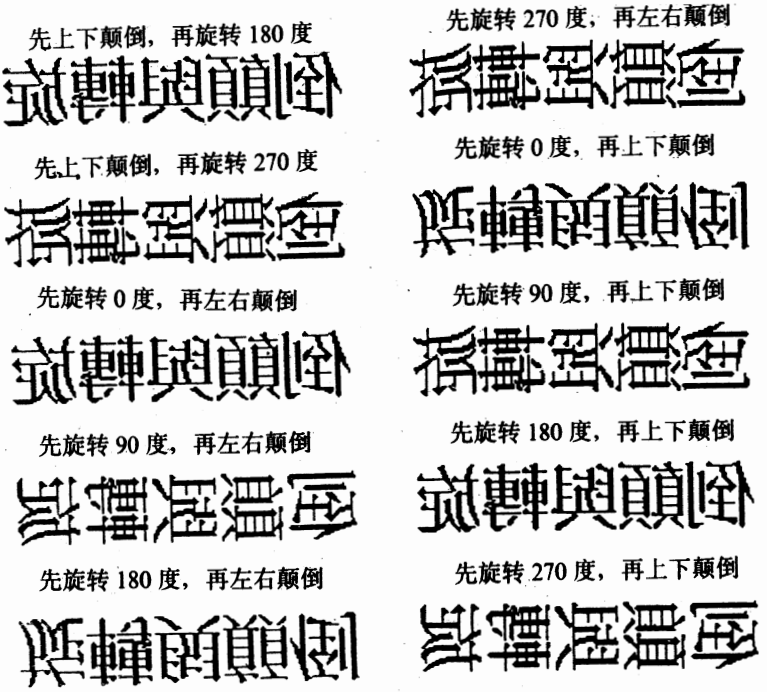


图 3.12　字形的旋转与颠倒(之二)

(5) 字模反向

字模点阵取反，则可按反相方式打印汉字，呈黑底白字效果。这就是所谓的反白体字。

(6) 加重打印

将输出的内容重叠打印几遍，从而实现汉字的加重打印（亦

称为加浓打印)。

(7) 背景

汉字加背景（或称字形套网）是利用背景字符或背景模式来实现的。

有的中文系统在字库中提供了背景字符，用作打印内容的背景，或淡化打印内容。例如，在 24×24 点阵宋体字库中提供一些背景字符，它们的区位码分别是 0660—0681。图 3.13 列出了这些背景字符及其对应的区位码。

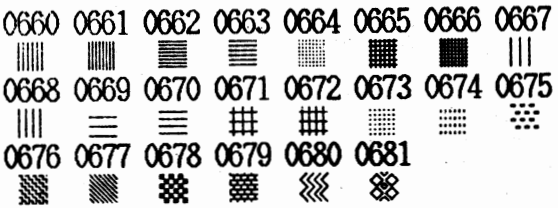


图 3.13 背景字符

除了中文系统提供的背景字符之外，用户还可以通过造字的方法建立自己所需要的背景字符。

有的中文系统还提供了背景模式，并予以编号（亦称为区纹代码）。用户可根据编号选择所需要的背景。图 3.14 给出了几种背景模式。

除了中文系统提供的背景模式之外，用户还可以自定义背景模式：首先指定编号，然后在屏幕上描绘自己需要的背景模式，以后便可通过这个编号来使用这个自定义的背景模式。当然，每个中文系统自定义背景模式的方法不同，对自定义背景模式的个数也有一定的限制。

用户可以利用中文系统提供的有关打印命令来对汉字加背景。例如，使用“或”操作，打印内容与背景字符或背景模式相“或”后印出具有背景的汉字。又例如，使用“与”操作，打印内容

与背景字符或背景模式相“与”后淡化打印内容。图 3.15 给出一些带背景的汉字。

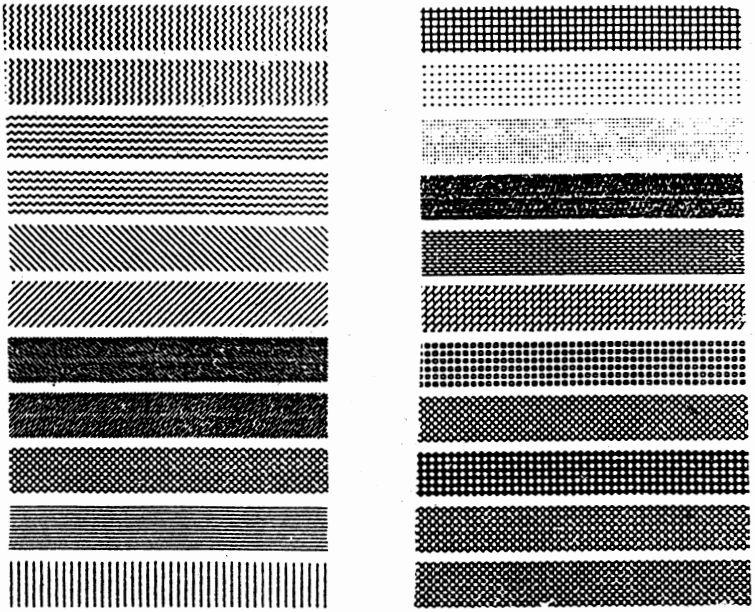


图 3.14　背景模式

3.　打印密度设定

打印机是利用点组成字打印在纸上，因此点与点的距离称为点密度。打印密度是指水平和垂直每英寸打印的点数。

点密度大小影响打印字的大小。每英寸点数越多，密度越高，字就越清楚，但字越小。

不同型号的打印机有不同点密度，因此，不同型号的打印机有不同的行距单位和行距值，每页行数和每行字数、点数也不同。

有些打印机提供了多种打印密度，供用户选择，而有些打印

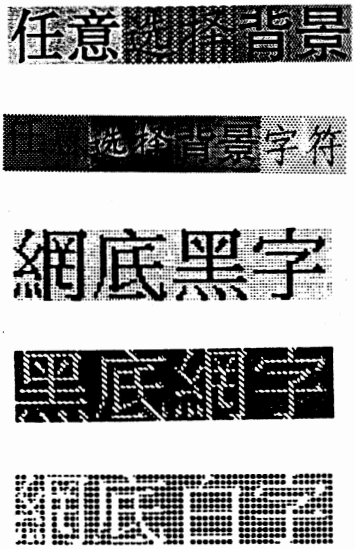


图 3.15　带背景的汉字

机则不具有此项功能。例如，EPSON LQ−1000，NEC P2200 按 24 针打印时有五种点密度 D0−D4，LQ−1500 可选择 D0−D3 四种，而 TOSHIBA P1351 则只有一种点密度。

具有多种密度输出的打印机，可通过设定不同的点数来改变打印密度。设定的点数要求接近或等于对应打印密度应有的点数。至于应该选择哪种密度，用户可自行测试，通过改变出厂默认值，重新设定点密度，来选择自己所需要的密度。图 3.16 中的表给出了几种型号打印机所具有的不同的点密度(每英寸点数，单位：DPI)。

4. 行距、行间距和字间距

行间距表示打印一行后的走纸距离，即一行底部与下一行顶部点的距离，或两个相邻行之间的间隔。行间距亦称上下字距。

代码	密度					
	D0	D1	D2	D3	D4	D5
B0	60	120	120	240		
B1	60	120		180	360	90
B2	60	120	120	240	80	90
B3				180		
B4				160		
B5			160	180		
B6	60	120	120	180	80	90
B9		120		180		
B11				180	360	
B12	60	120	120	240	80	90
B13				180		
B14				180		

图 3.16　打印机点密度对照表

字间距表示字与字间的间隔。字间距亦称左右字距，或字符列间距。

当使用字符集（例如，GB2312 基本集）中的制表符制表时，行间距和字间距应取为 0。

行距表示一行顶部与下一行顶部间的距离。行距与行间距之间的关系是：

行距＝行间距+当前字符高度

一旦上述一者确定，另一者随之确定。行距、行间距确定后，并不随以后字符高度的变化而变化。

当设定的行距小于当前字模高度时，发生叠打。当行距为 0 时，两行完全叠印在一起，以此可以实现加重打印。

打印驱动程序对行距设定的处理方法是：在本行信息接收

后，才发生前行的换行动作，因此本行的行距设定命令作用效果在前一行与本行之间开始。

行距、行间距和字间距以点为单位，或以英寸为单位。对于不同的打印机和不同的打印驱动程序，有不同的行距单位、行距值和可改变行距的范围。例如，对于 24 针打印机 P1350、P1351 的行距单位为 1／48″，即针密度为每英寸 48 点；M2402、NEC9400 的行距单位为 1／120″，即针密度为每英寸 120 点；M 1724 的行距单位为 1／160″，即针密度为每英寸 160 点；TOSHIBA 3037、M2024L、LQ1500 的行距单位为 1／180″，即针密度为每英寸 180 点。

5. 定位

所谓定位，就是定位打印头，上下左右移动打印头若干点。按定位形式可把定位分为绝对定位和相对定位两种形式。

(1) 绝对定位

绝对定位包括水平定位、垂直定位以及两者的结合形式。

水平定位一般以点为单位，且可反复多次重复定位。

垂直定位是指行内垂直定位。一个打印行的高度一般为打印针数，例如，9 针、24 针等。当放大打印时，打印行的高度就是针数的整数倍。如果一行内有若干个不同比例的字符，这些字符或者按顶对齐排成一行，或者按底对齐排成一行，这种样子不大美观。为了能把大小不同的字符按需要安置在一行中不同的高度位置上，常采用层定位的方法进行垂直定位。层以点为单位，每一行点为一层，由上而下的层数就是打印针数。

水平定位与垂直定位结合在一起，以点为单位，在水平和垂直两个方向上反复定位，可把字符、图形、制表线等定位到任意位置。例如，图 3.17 给出了汉字定位的例子。

(2) 相对定位

相对定位大致有以下几种形式：

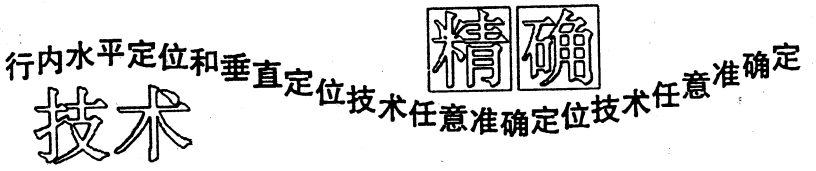


图 3.17　水平定位与垂直定位

①水平相对正向移动

打印头从当前位置向右移动若干个点。

②水平相对负向移动

打印头从当前位置向左移动若干个点。

③垂直相对向下移动

打印头从当前位置向下移动若干个点。这种定位形式相当于层定位，新层比原层降低若干个点。用它形成下标较为方便。

④垂直相对向上移动

打印头从当前位置向上移动若干个点。用它形成上标较为方便。

6. 上标和下标

上标和下标亦称为上角标和下角标。利用上标和下标可打印出复杂的数学公式、化学方程式等。

有些中文系统在字库的空白区中加入上标和下标符号，用户也可以通过造字的方法来定义自己需要的上标和下标符号。然而，这种构造上标和下标的方法在使用上受限制。

为了增强上标和下标打印的灵活性，一般采用下述方法来打印上标和下标：上标通过上移若干点印出；下标通过下移若干点印出；上下标通过上标上移若干点、下标下移若干点印出。以这种方法打印的上标和下标，可采用各种字体，也可放大或缩小，

还可多层嵌套。例如，图 3.18 给出上标嵌套、下标嵌套和上下标嵌套的例子。

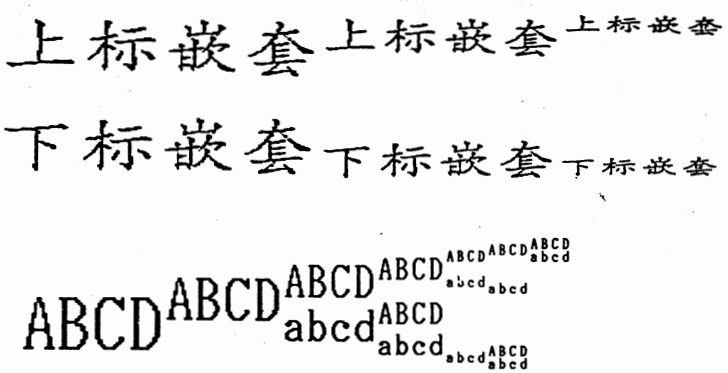


图 3.18　上标和下标的多层嵌套

7.　下划线、中划线、上划线、左划线和右划线

对于任何字体的字、任何放大倍数的字，都可以设定下划线(底线)、中划线（删字线)、上划线(顶线)、左划线和右划线。上划线和下划线主要用于横印，左划线和右划线主要用于直印。中划线用于表示删除字符。

这些线均有单线和双线之分，而且线的粗细可以由用户自行设定。

8.　字符排列方式

字符常按下列几种排列方式打印:

(1) 左对齐打印

每行字符靠左边对齐打印。左边界是指每行左端开始打印的位置。

(2) 右对齐打印

每行字符靠右边对齐打印。右边界是指每行右端终止打印的

位置。

(3) 居中打印

每行字符左边和右边留出的空白位置相同。

(4) 调整方式打印

每一行按指定的字数排齐打印，到达指定的字数时换行，原输入或编辑的回车换行符号不起作用。

(5) 空行

换 n 行就是空 n−1 行。

(6) 空格

水平方向上(包括一行的左边)留出若干个空白位置。

上述各排列方式中的位置，既可以按字符来计算，又可以按点来计算。

9. 页式打印

下面，介绍有关页式打印的几个问题。

(1) 页格式

页式打印就是按页格式打印。页格式如图 3.19 所示。

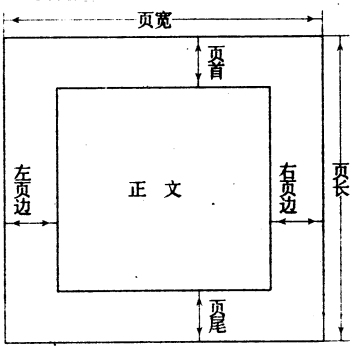


图 3.19 页格式

按页式打印，要首先确定每页行数、每行字数（列数或行

宽)、上下左右页边、页长和页宽。

对于不同型号的打印机和不同的打印驱动程序，每页行数和每行字数的最大极限值也不同，依打印机型号以及所选的点密度而定。

上页边亦称页头或页首，指页顶与正文顶部之间的距离。

下页边亦称页尾，指正文底部与页底之间的距离。

左页边亦称左边界，指页左边与正文左边之间的距离。

右页边亦称右边界，一般指正文右边与页右边之间的距离。有的指页左边到正文右边之间的距离。

页长＝页头长度+正文长度+页尾长度。

页宽＝左页边宽度+正文宽度+右页边宽度。

页长、页首、页尾既可用行数来计算，又可用点数来计算。同样，页宽、左页边、右页边既可用字符个数来计算，又可用点数来计算。

(2) 页首定义

页首定义用来定义页标题类型，页首空白和页标题间隙等。

页标题类型包括：左对齐，右对齐，居中，偶数页左对齐、奇数页右对齐等等。

页首空白是指每页开头处的空白，即页顶到页标题之间的距离，可按行或点来计算。

页标题间隙是指页标题与正文顶部之间的空白距离，可按行或点来计算。

(3) 页尾定义

页尾定义用来定义页尾注类型、页号和页尾注间隙等。

页尾注类型与页标题类型相似。

页尾注间隙是指正文底部到页尾注之间的空白距离，可按行或点来计算。

页号亦称为页码，是页的计数值。用户可指定开始页号，默

认初值为 1。

(4) 换页

在页式打印过程中，有以下几种换页形式:

①绝对换页

在打印过程中，连续打印，每满一页自动换页，另起一页打印。

②条件换页

在打印过程中，满足某种条件（例如，本页剩余不足 n 行）时开始新的一页，否则不换页，继续打印。

③暂停换页

每打印一页，暂停；按任意键继续打印。

(5) 页缓冲区

页缓冲区亦称为页暂存区。页式打印可用页缓冲区命令来控制。例如，设定页缓冲区的宽度、地址和大小，左移或右移页缓冲区指针，打印页缓冲区中指定行数的信息，清除页缓冲区等等。

10. 打印纸的控制

下面，介绍有关控制打印纸的若干问题。

(1) 打印宽度

一般说来，80 列打印机每行最多输出 80 个 ASCII 字符，136 列打印机每行最多输出 136 个 ASCII 字符。

每行最多可打印的字数，一方面取决于打印机的型号及所选的点密度，点密度（每英寸的点数）决定了每行最多可打印的点数，另一方面取决于所选择的字模点阵的大小（每个字的横向点数）、横向放大倍数和字间距。计算公式如下:

$$\text{每行最多字数} = \frac{\text{每行最多点数}}{(\text{每字横向点数} \times \text{横向放大倍数}) + \text{字间距}}$$

以 EPSON LQ-1500 打印机的 D3 点密度为例，每行最多打印点数为 2448，如果按 24×24 点阵字模打印，水平不放大，字间距假定为 0，则每行最多可打印 102 个汉字。

(2) 行馈给与页馈给

有两种控制打印机走纸的方式：行馈给与页馈给。如果是行馈给 (Linefeed)，则每次把纸向前推进一行。如果是页馈给 (Formfeed)，则每次把纸向前推进一页。

(3) 页头的定位

为了确定打印纸新的一页的开始位置，使打印纸的页头定位到正确的位置，往往要决定打印前后是否换页。有四种控制换页的方式：

①打印第一页前换页；

②打印最后一页后换页；

③打印第一页前和打印最后一页后各换一页；

④打印第一页前和打印最后一页后均不换页。

(4) 打印份数

决定每次打印的拷贝份数。

(5) 字折绕(Word Wrap)

字折绕是指字符串超过右边界时自动换行的能力。对于打印的字折绕，要决定超过右边界的字是换行，还是舍弃。

11. 打印方向

一般打印机均有两种控制打印方向的方式：单向打印和双向打印。单向打印是指打印机只从左向右打印；双向打印则是从左向右、从右向左两个方向打印。

双向打印速度快，但由于有的打印机精度不高，双向打印的竖线呈现锯齿状；这时可选择单向打印，以牺牲打印速度来保证精度和定位准确，尤其是当打印表格或放大打印汉字时效果更

好。

有的打印机的打印方向需要通过打印机上的开关来切换。有的打印机仅有单向打印或双向打印。

12. 制表

下面，介绍几种用打印机绘制表格的方法。

(1) 用画线命令或制表命令制表

用户可用多种方式和多种线型，实现各种复杂表格的打印。例如，根据对角线顶点的坐标可打印方框。

为了简化起见，有人把表格线分解为两种基本类型：横线和垂线组。横线要给出线型、垂直位置和水平起止坐标。垂线组要给出各垂线的线型、水平坐标和垂直起止位置。水平坐标和垂直位置均以点值给出。任何复杂的表格都可用这两种表格线组成。

(2) 用字符集中提供的制表符打印表格

例如，用 GB2312 图形字符代码表中第 9 区的制表符（见图 3.20）来打印表格。可用区位码来输入这些制表符。有些中文系统特意设置了一种常用输入方案，用于输入制表符等常用字符。

又例如，用 BIG-5 码或 IBM5550 码中提供的制表符打印表格。可用内码或仓颉码输入这些制表符。如图 3.21 所示。

注意，在用制表符打印表格线时，必须设置行间距和字间距为 0，否则垂线和横线之间有空隙。有的打印系统可自动设定行间距和字间距。当打印表格开始时，自动置行间距和字间距为 0；待打印表格结束后，恢复打印表格之前的行间距和字间距。

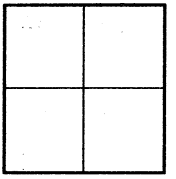
(3) 用系统规定的按键产生制表符

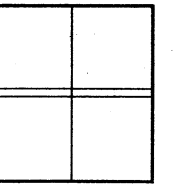
有些中文系统提供了用按键产生制表符的功能。按下规定的制表功能键后，便可通过使用规定的制表键来输入制表符。

例如，在 CCDOS 4.0 中，按下功能键 Alt-F5 后，便可使用小键盘来输入制表符。当按下制表键后，提示区将该键对应的制表符显示出来，供用户选择。再次按下 Alt-F5 键取消制表功

区位码	字符	名称	区位码	字符	名称
0904	—	x-x	0942	┤	Y-x-y
0905	—	X-X	0943	┤	y-x-Y
0906	|	y-y	0944	┤	Y-x-Y
0907	|	Y-Y	0945	┤	Y-X-y
0908	┄	虚3x-x	0946	┤	y-X-Y
0909	┄	虚3X-X	0947	┤	Y-X-Y
0910	┆	虚3y-y	0948	┬	x-x-y
0911	┆	虚3Y-Y	0949	┬	x-X-y
0912	┈	虚4x-x	0950	┬	X-x-y
0913	┈	虚4X-X	0951	┬	X-X-y
0914	┆	虚4y-y	0952	┬	x-x-Y
0915	┆	虚4Y-Y	0953	┬	x-X-Y
0916	┌	x-y	0954	┬	X-x-Y
0917	┌	X-y	0955	┬	X-X-Y
0918	┌	x-Y	0956	┴	x+y-x
0919	┌	X-Y	0957	┴	x+y-X
0920	┐	-x-y	0958	┴	X+y-x
0921	┐	-X-y	0959	┴	X+y-X
0922	┐	-x-Y	0960	┴	x+Y-x
0923	┐	-X-Y	0961	┴	x+Y-X
0924	└	x+y	0962	┴	X+Y-x
0925	└	X+y	0963	┴	X+Y-X
0926	└	x+Y	0964	┼	x+y-x-y
0927	└	X+Y	0965	┼	x+y-X-y
0928	┘	y-x	0966	┼	X+y-x-y
0929	┘	y-X	0967	┼	X+y-X-y
0930	┘	Y-x	0968	┼	x+Y-x-y
0931	┘	Y-X	0969	┼	x+y-x-Y
0932	├	x+y-y	0970	┼	x+Y-x-Y
0933	├	X+y-y	0971	┼	x+Y-X-y
0934	├	x+Y-y	0972	┼	X+Y-x-y
0935	├	x+y-Y	0973	┼	x+y-X-Y
0936	├	x+Y-Y	0974	┼	X+y-x-Y
0937	├	X+Y-y	0975	┼	X+Y-X-y
0938	├	X+y-Y	0976	┼	X+y-X-Y
0939	├	X+Y-Y	0977	┼	x+Y-X-Y
0940	┤	y-x-y	0978	┼	X+Y-x-Y
0941	┤	y-X-y	0979	┼	X+Y-X-Y

图 3.20　GB2312 图形字符代码表中第 9 区的制表符





符号	仓颉码	BIG-5 内码	符号	仓颉码	BIG-5 内码
ㄏ	ZXIP	A27A	ㄱ	ZXIQ	A27B
ㄧ	ZXIN	A278	—	ZXIM	A277
ㄚ	ZXIK	A275	ㄐ	ZXIJ	A274
十	ZXIG	A271	ㄒ	ZXII	A273
ㄨ	ZXIH	A172	ㄴ	ZXIR	A27C
ㄩ	ZXIS	A27D	二	ZXIX	A2A4
ㄞ	AXIY	A2A5	ㄓ	ZXJA	A2A6
ㄔ	ZXJB	A2A7			

图 3.21　BIG-5 码中的制表符

能。制表键及其对应的制表符如图 3.22 所示。

又例如，在联想式汉字系统中，当使用 PE 时，按住 Shift 键，可通过使用小键盘输入制表符，制表键及其对应的制表符如图 3.23 所示。

13.　直接打印

直接打印亦称为立即打印，用于把直接在打印命令后面列出的字符直接送往打印机。亦可把直接送往打印机的字符用规定的特殊符号括起来，例如 $ ABC $ 将 ABC 送往打印机。

如果直接送往打印机的字符是可打印字符，则可直接打印出这些字符，从而直接从打印机得到所需要的信息。

如果直接送往打印机的字符是功能控制码，则可直接使用打

方向 / 数字键	对 应 的 制 表 符
7 / Home	┌ ┌ ┌ ┌
—	─ ─ ┆ ┆
8 / ↑	┬ ┬ ┬ ┬ ┬ ┬ ┬ ┬
9 / PgUp	┐ ┐ ┐ ┐
4 / ←	├ ├ ├ ├ ├ ├ ├ ├
5 / +	┼ ┼ ┼ ┼ ┼ ┼ ┼ ┼ ┼ ┼ ┼ ┼ ┼ ┼ ┼
6 / →	┤ ┤ ┤ ┤ ┤ ┤ ┤ ┤
1 / End	└ └ └ └
2 / ↓	┴ ┴ ┴ ┴ ┴ ┴ ┴ ┴
3 / PgDn	┘ ┘ ┘ ┘

图 3.22　CCDOS 4.0 的制表键及其对应的制表符

Home	↑	PgUp	┌	┬	┐	Ins	=	—
←		→	├	┼	┤	Del	=	|
End	↓	PgDn	└	┴	┘			

图 3.23　联想式汉字系统中 PE 的制表键及其对应的制表符

印机的控制功能。特别是对于设定打印驱动程序未提供而打印机原有提供的功能更为有效。例如，EPSON LQ−1000 打印机本身提供全速／半速打印控制功能（Esc S 0 全速打印，Esc S 1 半速打印）。若打印驱动程序未提供相应的命令，则可通过直接打印来实现全速／半速打印控制功能。

14.　屏幕打印

屏幕打印亦称为屏幕拷贝。开机后，可随时按屏幕打印键，打印当前屏幕上显示的信息。

在打印屏幕上显示的图形之前，可设置打印参数，例如，纵向和横向比例、打印方向等。由于打印机点密度的关系，屏幕打印的图形会较小，因此一般应放大打印或更改打印的点密度。

15. 图形打印

图形打印包括打印内存图形、屏幕图形、图形文件等。图形文件可以是多种格式，例如，某一中文系统规定的格式、扫描仪格式、传真格式等。对于图形打印，可提供开窗、定位、比例、旋转、与、或、非、异或等多种操作。

屏幕图形就是屏幕上显示的图形。可把屏幕图形存盘，从而形成图形文件。图形可以用任何手段做出，例如，用BASICA、LOTUS 1-2-3、AUTOCAD等软件做出的图形。当文章中需要插图时，可在文件中的适当地方加上相应的图形文件调用命令，从而实现图形和文字的混合排版。

16. 彩色打印

对于彩色打印机，通过给出编号来设定打印颜色，例如，黑色、黄色、粉红色、蓝色、浅蓝色、紫色、橙色、橙红色、绿色、红色、咖啡色等等。

3.3 汉字造字

3.3.1 汉字造字程序

大多数中文系统都提供了汉字造字程序。提供造字程序的目的是让用户自行设计和使用系统所没有的汉字或其它图形字符，以满足用户特殊用途的需要，而不受系统原字库的限制。

通用汉字造字程序往往适用于多种字体、多种点阵的字形，用户可以根据菜单提示选择所要造的字形种类，并提供相应的字

库及其文件名。所造汉字及其它图形字符均设置在字符集的保留区。例如，GB2312 基本集的第 10～15 区和第 88～94 区为保留区，可造字 1222 个，区位码分别为 1001～1594 和 8801～9494。当然，在基本集的其它某些区中还有一些零散的空白位置可供造字，比如，第 9 区的 0980～0994，第 55 区的 5590～5594。又例如，Microsoft MS-DOS 中文版总共可造 5640 个新字，使用的内码范围为 DF30～FCFD。还例如，简体字 5550 中文 DOS 规定用户可自定义 500 个字，内码为 22301～22830。至于保留区是存储在原字库中，还是存储在中文系统另外设置的用户字库中，取决于不同的中文系统。

造字程序一般具有下列几种功能:

1. 造字

造字就是字形编辑，编辑汉字及其它图形字符的点阵字形。造字分为以下两种功能:

(1) 改字

修改字库中已有的汉字或其它图形字符的字模点阵，改造为用户所需要的字形;

(2) 加字

在字库的空白位置上定义新的字模点阵，产生新的汉字或其它图形字符（包括新的部首)。

造字操作步骤如下:

(1) 启动造字程序，进入造字程序主菜单。

(2) 选择所造字形的点阵和字体，并键入相应的字库文件名。

(3) 进入字形编辑屏幕，按屏幕提示给出的编辑功能，编辑汉字或其它图形字符的字形。在编辑之前，若改字，则用任一输入方案键入要修改的汉字或其它图形字符；若加字，则键入或选择（从造字程序提供的可用位置的内码中选择）新造字的内码

(区位码或国标码)。在编辑之后，要把字形存入磁盘。

(4) 退出造字程序。

2. 删字

删字就是删除字库中指定位置的一个或一段区域内的汉字或其它图形字符的字模。在删字之前，可在屏幕上显示这些字符的字形。

3. 字模复制

字模复制亦称字模拷贝，就是把字库中指定位置的一个或一段区域内的汉字或其它图形字符的字模依次拷贝到另一指定位置的一个或一段区域中去。在复制之前，可在屏幕上显示这些字符的字形。

4. 改码

改码是指更改字库中某个汉字或其它图形字符字模的编码(内码、区位码或国标码)。改码过程是：首先，可用任一输入方案键入要改码的汉字或其它图形字符，并在屏幕上显示出该字符的字形；然后，键入新编码，亦可用其它输入方案键入新编码原来代表的汉字或其它图形字符，从而要改码的汉字或其它图形字符获得这个新编码。例如，“燕”字的区位码原为 4964，将它的区位码改为“冰”字的区位码 1789 后，实际上是用“燕”字的字模复盖了“冰”字的字模，字库中区位码为 1789 和 4964 的位置均存了“燕”字的字模，亦即“冰”字的字模从字库中消失了。改码之后，不论用任何输入方案，只要键入“冰”字所对应的汉字输入码，则一定得到“燕”字。

5. 查询

所谓查询，就是在屏幕上显示字库中汉字或其它图形字符的

字形、内码（区位码或国标码）、输入编码等有关信息，以便辅助造字或检查造的字是否正确。

3.3.2 字形编辑

字形编辑就是在字形编辑屏幕上，按屏幕提示给出的编辑功能，编辑汉字或其它图形字符的字形。下面，分别介绍字形编辑屏幕和字形编辑功能。

1. 字形编辑屏幕

一般汉字造字程序的字形编辑屏幕都设有造字区和提示区，有的汉字造字程序还设有参考区、原形区和放大区。

(1) 造字区

造字区亦称编辑区，是造字的工作区，用于编辑字形。

(2) 提示区

提示区亦称功能区，用于显示字形编辑功能及当前工作状态。

(3) 参考区

参考区亦称参照区或暂存区，用于拼字或存放造字的中间结果，暂存备用。可把一个字先读到参考区，进行加工后，再把它复制或叠加到造字区。参考区与造字区同时放于屏幕上，通过控制键可使光标在两个区之间来回移动。亦可设置几个参考区，通过按键把某一参考区的字形读到屏幕上。

(4) 原形区

原形区亦称造字窗，用于把造字区中所造的字形以正常大小显示在屏幕上，即在原形区内显示一个实际大小的汉字或其它图形字符。

(5) 放大区

放大区用于把造字区中所造的字形放大若干倍后显示在屏幕上。

(6) 标记区

为了对造字区或参考区中的部分字形进行操作，往往在造字区或参考区中构造一个矩形区域，然后再对这个区域进行操作，这个指定区域称为标记区。对标记区的操作可以有复制、引用、清除、填充、反向等。

2. 字形编辑功能

汉字造字程序一般具有下列字形编辑功能:

(1) 光标移动

通过光标控制键可使光标在造字区或参考区内上移、下移、左移、右移、左上移、右上移、左下移或右下移，还可以使光标移到左上角、右上角、左下角或右下角。

(2) 写点和抹点

在造字过程中，有时要在无点的地方写上点，也有时要把有点的地方抹掉，还有时仅需要移动光标，既不写点也不抹点。这是光标位置的三种属性，可通过规定的属性键来改变属性。若不按动属性键，则光标移过的位置仍保持原来的属性：连续写点、连续抹点或既不写点也不抹点。也就是说，在未改变属性之前，凡光标所到之处均保持写点、抹点或不写点也不抹点的状态，直到改变为新属性为止。

(3) 读字

读字亦称取字、叫字、呼字。读字就是用任一输入方案把所要修改的汉字或其它图形字符的字形显示到造字区或参考区。读字的目的是利用中文系统中原有的字形，以便修改为所需要的字形。

(4) 字形清除

字形清除也称为字形删除，它不同于删字，是从屏幕上删除字形，而不是从字库中删除字模。字形清除包括：清除整个造字区或参考区；清除某一标记区；删除当前光标上方、下方、左

方、右方、左上方、右上方、左下方或右下方的所有点；删除当前光标所在行或列（同时把光标移动到指定位置）；删除不需要的点。

(5) 字形填充

把整个造字区或参考区均写满点，也可把某一标记区填满点。

(6) 字形插入

在光标所在处往上、往下、往左或往右插入一空白行、空白列或空白点。

(7) 字形移动

字形移动包括：整个字形向上、向下、向左或向右平移，以删除不需要的字形；在某一标记区内字形向上、向下、向左或向右平移，例如，将光标所在行开始的上边或下边各行顺序向上（或向下）移动一行，光标所在行下边（或上边）各行不动；将光标所在列开始的左边（或右边）各列顺序向右（或向左）移动一列，光标所在列清除。

(8) 字形压缩

字形按比例向上、向下、向左或向右压缩。

(9) 字形旋转

字形按逆时针或顺时针旋转指定的角度。

(10) 字形反向

造字区或参考区中黑白反向，无点的位置写点，有点的位置抹点，某一标记区中的字形也可反向。

(11) 字形粗细变换

把字形的笔画变粗或变细。

(12) 字形复制

字形复制亦称字形拷贝。它不同于字模复制，字模复制是复制字库中的字模，而字形复制是将某一标记区内的字形拷贝到造字区或参考区的另一指定位置。

(13) 拼字

拼字亦称合并字形，把造字程序提供的偏旁部首或参考区内加工后的字形叠加到造字区中原有的字形上，从而合并为一个新的字形。下面，以 24×24 点阵“”字为例，来说明拼字的方法。

①读字：把“他”字读入造字区，把“淑”字读入参考区。如图 3.24 所示。

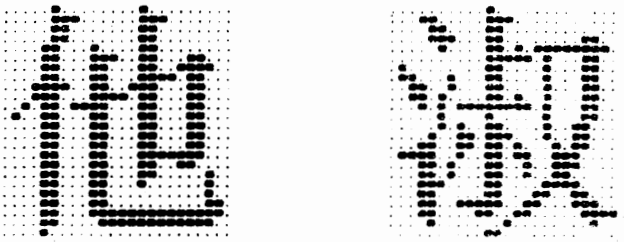


图 3.24　读字

②取“他”字的单立人偏旁“亻”，取“淑”字的右边偏旁“叔”部分。如图 3.25 所示。

图 3.25　取偏旁

③拼字：将参考区的字形拷贝到造字区，与造字区的字形合并成一个新字“”。如图 3.26 所示。

④将造字区字形存盘。

(14) 字形打印

打印造字区或参考区中的字形。

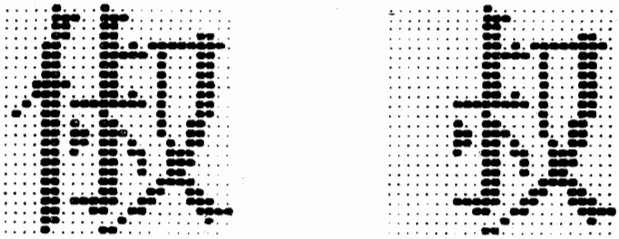


图 3.26　拼字

3.3.3　如何使用新造的字

对于在旧字位置上造的新字（即修改字库中已有汉字或其它图形字符的字形），仍可按旧字的汉字输入编码输入新字。由于新字的字形、字音、字义与旧字不同，因此，除内码（区位码或国标码）之外，旧字的字形、字音、字义输入码可能不适合新字，需要重新编入相应的汉字输入编码方案。

对于在空白位置上造的新字（即在字库的空白位置上新增加的汉字或其它图形字符的字形），在将它编入相应的汉字输入编码方案之前，只能用内码（区位码或国标码）输入它。

造字应同时考虑显示字库和打印字库，显示字库中汉字或其它图形字符供显示用，打印字库中汉字或其它图形字符供打印用。此外，新造的字采用多字体打印时，还要考虑不同点阵和不同字体的打印字库。

3.4 汉 卡

由于汉字字量大，汉字字库占的存储量也很大。仅就 16×16 点阵字库而言，按 87 个区计算，就占 261696 个字节。如果将字库放于 ROM 中，则占据运行软件所需要的内存空间，使有些软件无法运行。如果将字库放于磁盘上，则由于访问磁盘次数频繁，严重影响软件的运行速度。

一般应用软件是在西文操作系统基础上开发的。也就是说，软件利用西文操作系统之外的内存空间进行存储分配。由于中文操作系统及其汉字字库占据了一定的内存空间，致使许多软件在中文操作系统上运行空间不够用，因此，这些原来在西文操作系统上运行的软件无法移植到中文操作系统上运行。图 3.27 给出在西文操作系统上和在中文操作系统上运行软件的内存空间比较示意图。

目前，尽管很多微型计算机扩充了 RAM，但由于一般应用软件是以 640K 内存为基础开发的，因此，如果软件未提供扩展内存的功能（使用 640K 之外内存）的功能，即使硬件扩充了 RAM，而且操作系统支持这种扩充功能，也无法在中文操作系统上运行存储空间要求较大的软件。

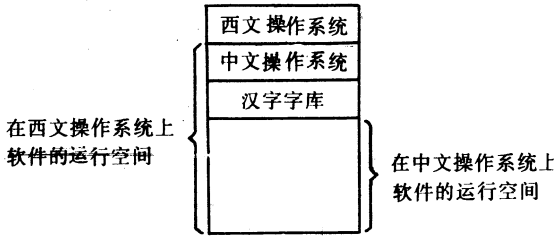


图 3.27 在西文操作系统上和在中文操作系统上软件的运行空间

为了解决上述采用软字库方式存储和存取汉字字模存在的问题，可采用硬字库方式存储和存取汉字字模。有效的办法就是在机器中安装汉卡（插入扩展槽中）。汉卡亦称为中文卡。大部分汉卡装有16×16点阵汉字字库，也有的还装有部分或全部24×24点阵汉字字库。只装有汉字字库的汉卡有时称为字库卡。

除装有汉字字库外，有的汉卡还装有某些汉字输入编码方案；有的汉卡用于构造汉字字形发生器和显示缓冲区，按字符方式显示汉字，致使DOS的INT10显示模块不必修改，使程序语言和应用软件显示的运行环境不变，因此，在以这种汉卡支持的中文操作系统上运行的西文软件，大多数无需汉化便可直接显示汉字，从而最大限度地减少了软件汉化的工作量。

汉卡是由ROM或RAM芯片制成的。有的汉卡是把信息固化在ROM上，有的汉卡则是用RAM存取信息。后者中的字库可随时切换，例如，宋体、仿宋体、楷书体、黑体字库的切换，繁体字库与简体字库的切换，汉字基本集与汉字辅助集的切换，多种语言文字字库的切换等等。

总之，一方面，汉卡可节省内存空间，保持应用软件在西文操作系统上的原运行环境，可以运行大程序；另一方面，由于汉卡是由ROM或RAM构成的，运行速度可大大加快。但是，由于汉卡造价较高，对于不要求运行较大程序的用户，宁愿使用以软字库方式存储和存取字模的中文操作系统；另外，有的汉卡破坏了中西文软件的兼容性，使一些西文软件在汉卡环境下无法运行或丧失一些功能。

3.5 中文操作系统支持的中文软件

由于中文操作系统与西文操作系统的兼容程度不同，因此它们支持的中文软件和公用程序与西文软件和公用程序的兼容程度

也不同。鉴于上述原因，西文软件和公用程序要移植到中文操作系统上，大多需要汉化。又由于中文操作系统与西文操作系统的兼容程度不同，软件汉化工作量也不同。例如，对于显示 11 行的中文操作系统，就无法保持西文数据表软件的 25 行显示；而显示 25 行的中文操作系统则对西文数据表软件的显示行数无需汉化。除了汉化软件之外，各种中文操作系统上也独立开发了适合于本系统的中文软件和公用程序，。比如，慧星、赛诸葛、神龙双子星等。

微型机上各种中文操作系统支持的常用中文软件和公用程序如下:

(1) 程序设计语言

BASIC，FORRAN，COBOL，C，PASCAL，LISP，PROLOG，FORTH，MACRO ASSEMBLER 等;

(2) 数据库管理系统

dBASE，Clipper，Foxbase，Quicksilver，Informix，SQL，Datastore，Paradox，Oracle，R: base 等。

(3) 字处理软件

Wordstar, PE, Word，Word Perfect，Displaywriter，Perfessional Write，Norton Editor 等;

(4) 数据表软件

LOTUS 1－2－3，Multiplan，Supercalc，Vicicalc，VF，QA 等;

(5) 计算机辅助设计与图形软件

AUTOCAD，MASTER GRAPHICS，dGRAPH 等。

(6) 组合软件

Symphony，Framework，Knowledgeman，ABLE－ONE，GEM Desktop 等;

(7) 统计软件

SPSS, STAISTICS, STATPAK 等;

(8) 公用程序

各种汉字多字体打印程序、汉字造字程序、汉字排序程序、合并程序、拷贝程序、内码转换程序（例如，简体字与繁体字转换程序，国标码与 IBM 5550 内码转换程序，公会码、BIG-5 码、通用码、IBM 5550 码、电信码、倚天码、王安码、精业码、HP 码等各种内码的转换程序）。

在中文操作系统上运行软件的关键问题是保持它们原来在西文操作系统上的运行环境，也就是说，保持西文软件的原有功能，保持中西文软件的兼容性。而中文软件运行环境的主要问题是软件运行空间。由于中文操作系统在西文操作系统基础上增加了键盘输入模块、显示模块、打印模块等程序模块、并增加了汉字字库，它们占据了软件原来在西文操作系统上运行的内存空间，因此，有些基于西文操作系统开发的软件在中文操作系统上无法运行，或无法运行较大的程序。解决这一问题，有下列几种办法:

(1) 安装汉卡，用以存储汉字字库或程序模块。

(2) 将汉字字库全部或部分驻留在磁盘上。

(3) 程序模块的内存分配采用覆盖技术，尤其是打印模块占空间较大，应分成若干个功能独立的子模块，内存中应只存放当前正执行的那一部分程序，从而节省内存空间。

(4) 使用具有内存扩展功能的应用软件，扩展使用 640K 内存外的 RAM，例如，dBASE IV 1.1 版和 LOTUS 1-2-3 3.0 版均提供了 EMS 功能。

(5) 把中文操作系统设计成模块化结构，每个模块具有一定的独立性，例如，汉字字库管理模块、汉字显示模块、汉字键盘输入模块、汉字打印模块等。根据不同的需要，可随时把某些模块撤消；从而减少系统占用的内存空间。例如，当输入时，可撤消汉字打印模块；当打印时，可撤消汉字键盘输入模块。

(6) 汉字字库采用压缩信息存储方法，例如，向量存储法、

部件组字法等，用以减少字库的存储量。

(7) 将汉字字库装入 640K DOS 基本内存之外的 EMS 存储及 80286 或 80386 的 1M 以上存储器。尽管 286、386 等高档微型机具有 1M、2M 或更多的存储空间，而且高版本 DOS 也支持这些存储，但至今具有这种内存扩展功能的应用软件还不多。目前，绝大多数软件的唯一运行空间仍是地址为 0～640K 的基本内存。为了减少中文操作系统对基本内存的占用量，可把汉字字库放于地址为 640K 以外的 EMS 扩展内存或地址为 1M 以上的 Extend 扩展内存。

第四章　繁体字与简体字兼容的中文系统

繁体字与简体字的沟通，是中文电脑发展的必然趋势。众所周知，汉字有繁体字与简体字之分。新加坡和我国大陆主要使用简体字，而台湾省、香港、马来西亚及其它地区的大部分华人则主要使用繁体字。鉴于繁体字与简体字的选字范围、排列顺序、输入法、交换码、内码不统一，致使中文软件难以推广使用。为了推动中文电脑的发展，提高中文软件的适应性，促进贸易、文化交流、资料检索、图书管理、历史研究的发展，建立繁体字与简体字兼容的中文系统势在必行，繁体字中文系统与简体字中文系统的沟通更是迫在眉睫。

本章简述繁体字中文系统与简体字中文系统的主要区别，并讨论有关繁体字与简体字兼容的软件包和中文系统的若干问题。

4.1　繁体字中文系统与简体字中文系统的主要区别

目前，繁体字中文系统与简体字中文系统的主要区别在于：

1. 选取汉字的字量、字序不同

简体字中文系统大多采用 GB 2312–80 规定的字符集及其排列顺序，该字符集收录常用简体字 3755 个，次常用简体字 3008 个，其它图形字符 682 个，共计 7445 个。繁体字中文系统大多采用 BIG–5、IBM 5550、通用码、TAC 规定的字符集及其排列

顺序，例如，TAC 字符集收录常用繁体字 5401 个，次常用繁体字 7650 个，其它图形字符 234 个，共计 13285 个。

2. 汉字输入法不同

简体字中文系统大多采用拼音、五笔字型、国标、区位等汉字输入方案。繁体字中文系统大多采用注音、仓颉、内码等汉字输入方案，即使是一种输入方案，例如，注音，各种中文电脑的键盘配置图也不一样（见图 2.13 和图 2.14）。

3. 汉字的内码不同

为了提高汉字处理的效率，加快信息传输的速度，繁体字中文系统和简体字中文系统大多采用两字节编码技术，尽管如此，繁体字与简体字的内码的码值及编码方式仍相异很大。尤其是，台湾采用的汉字内码目前正处于万"码"奔腾的局面。

4.2 繁体字与简体字兼容的软件包

目前，我国大陆和台湾已出现多种繁体字与简体字转换的中文系统。由于一方面，这种中文系统还不是真正的繁体字与简体字兼容的中文系统，另一方面，往往是在原中文系统基础上用外壳(SHELL)方式实现的，因此，我们把它们称为繁体字与简体字兼容的软件包。

1. 繁体字与简体字兼容的软件包的几种形式

下面，例举繁体字与简体字兼容的软件包的几种形式，由此可见一斑。

(1) 建立具有繁体字与简体字对应关系的繁体字库和简体字库，因此，既可以输入简体字，输出繁体字；又可以输入繁体字，输出简体字。

(2) 建立繁体字与简体字的内码对照表，因此，可实现繁体写与简体字的相互转换，既可以把简体字中文软件移植到繁体字中文系统上，又可以把繁体字中文软件移植到简体字中文系统

上。

(3) 在中文系统中设置简体字与繁体字两种方式，显示程序和打印程序控制两种不同的字库——繁体字库和简体字库，通过两种方式的切换，在简体字方式下可运行具有简体字内码的中文软件，而在繁体字方式下又可运行具有繁体字内码的中文软件。

作者与赵大为同志曾在 1985 年研制了一个繁体字与简体字兼容的软件包 FJDOS。FJDOS 属于第一种形式的繁体字与简体字兼容的软件包。微型计算上的各种版本的简体字中文系统，只要加上 FJDOS，便可改造为繁体字与简体字兼容的中文系统。FJDOS 中含有 16×16 点阵和 24×24 点阵的繁体字库。繁体字库的选字范围和排列顺序，是以 GB 2312–80 为背景的，是在简体字库的基础上用繁体字替换对应的简体字而成的，也符合第一辅助集的要求。根据不同的需要，用户可随时通过命令切换简体字库和繁体字库。因此，既可输入显示简体字，打印简体字；又可输入显示简体字，打印繁体字；也可输入显示繁体字，打印繁体字；还可输入显示繁体字，打印简体字。FJDOS 中的繁体字库是在 GB 2312 简体字库基础上改造而成的，它保留了简体字库中的一般符号、序号、数字、英文字母、日文假名、希腊字母、俄文字母、汉语拼音字母、汉语注音符号以及非简化汉字，而用繁体字替换简体字库中对应的简化字。这些繁体字的选择是以《简化字总表》和《第一批异体字整理表》为准的。除一个简体字对应一个繁体字的情况外，FJDOS 考虑了一个简体字对应多个繁体字的情况，并对繁体字的输入、显示和打印做了相应的改变。除此之外，FJDOS 还提供了繁体字文章与简体字文章转换程序。

2. 一个简体字对应多个繁体字

简体字与繁体字的对应关系，除一个简体字对应一个繁体字的情况外，还存在着一个简体字对应多个繁体字的情况，可把它们分为以下几类:

(1) 多个繁体字简化为一个简体字。例如，“發”和“髮”字均简化为“发”字。

(2) 繁体字简化为原来已存在的汉字。例如，“瞭”字简化为“了”字，而“了”字原来就存在。因此，这种汉字具有两重性：从繁体字简化这个意义上说，它是简体字；而从它原来就存在这个意义上说，它是繁体字。

(3) 繁体字简化后，在某种意义上仍保留这个繁体字。例如，“像”字简化为“象”字，但在“象”和“像”字意义可能混淆时，仍然用“像”字。又例如，藉口、凭藉的“藉”字可简化为“借”字，但慰藉、狼藉的“藉”字仍用“藉”。

(4) 上述三类情况的组合。例如，“幹”和“乾”字均简化为“干”字，属第一类情况；干涉和天干的“干”字原来就存在，又属第二类情况；乾坤、乾隆的“乾”字不简化，尚属第三类情况。

对于上述一个简体字对应多个繁体字的情况，取使用频度较高的繁体字替换简体字库中对应的简体字，而使用频度较低的繁体字放于繁体字库的空白位置上（比如，第 11 区到第 12 区）。例如，“發”和“髮”字均简化为“发”，但“發”比“髮”的使用频度高，因此取“發”字替换简体字库中的“发”字，而把“髮”字放于繁体字库的空白位置上。此外，对于繁体字简化后仍保留这个繁体字的情况，繁体字常常已存在于简体字库中，因此，只要保留原繁体字，而不必再向繁体字库的空白位置放置繁体字。例如，“藉”字有时简化为“借”字，但“藉”和“借”字均存在于简体字库中，因此只要在繁体字库中仍保留这两个汉字即可。

对于正体字和异体字，一般人习惯把笔画少的正体字看作简体字，例如，“异”的异体字“異”，因此可把这些异体字当作繁体字来处理。

根据上述原则，在 GB2312 基本集的 6763 个汉字中，对于一个简体字对应多个繁体字的情况，简体字库与繁体字库的对应关系如下：

简体字	繁体字		简体字		繁体字			简体字		繁体字		
1658	1658	1101	1669		1669	1102		1777		1777	1103	
摆	擺	襬	板		板	闆		表		表	錶	
1780	1780	1104	1802		1802	1105		1828		1828	1106	
别	別	蹩	并		并	並		布		布	佈	
1837	1837	1107	1953		1953	1108		1969		1969	1109	
才	才	纔	痴		癡	痴		冲		冲	衝	
1986	1986	1110	2029		2029	1111		2084		2084	1112	1113
出	出	齣	唇		唇	脣		呆		呆	獃	騃
2117	2117	1114	2177		2177	1115		2192	2194	2192	2194	
当	當	噹	淀		淀	澱		迭	叠	迭	叠	
2212	2212	1116	2223		2223	1117		2302		2302	1118	
冬	冬	鼕	斗		斗	鬥		发		發	髮	
2365	2365	1119	2418	2420	2418	2420	1120	2441	3912	2441	3912	1121
丰	丰	豐	覆	复	覆	復	複	干	乾	幹	乾	干
2540	2540	1122	2545		2545	1123		2546		2546	1124	
谷	谷	穀	雇		僱	雇		刮		刮	颳	
2550	2550	1125	2647		2647	1126		2669		2669	1127	
挂	掛	挂	合		合	閤		哄		哄	鬨	
2683	2683	6565	2690		2690	1128		2756		2756	1129	
后	后	後	胡		胡	鬍		回		回	迴	
2767	2767	1130	2779	6623	2779	6623		2781		2781	1131	
汇	匯	彙	伙	夥	伙	夥		获		獲	穫	
2802	2802	1132	2803		2803	1133	1134	2824		2824	1135	
饥	饑	飢	迹		蹟	迹	跡	几		幾	几	
2850	2850	1136	2888		2888	1137		2910		2910	1138	
家	家	傢	鉴		鑒	鑑		姜		姜	薑	

简体字	繁体字		简体字		繁体字				简体字			繁体字		
2960	2960	1139	2969	2972	2969	2972			3001			3001	1140	
杰	傑	杰	藉	借	藉	借			尽			盡	儘	
3062	3062	1141	3077		3077	1142			3143			3143	1143	
巨	巨	鉅	卷		卷	捲			克			克	剋	
3205	3205	1144	3206		3206	1145			3207			3207	1146	
昆	昆	崑	捆		捆	綑			困			困	睏	
3259	3259	1147	3265		3265	1148			3269			3269	1149	
累	累	纍	泪		淚	泪			厘			厘	釐	
3279	3279	1150	3290		3290	1151			3343			3343	1152	
里	裏	里	历		歷	曆			了			了	瞭	
3417	3417	1153	3456		3456	1154			3462			3462	1155	
卤	鹵	滷	仑		侖	崙			罗			羅	囉	
3486	3486	1156	3508		3508	1157			3520	7159	8765	3520	7159	8765
脉	脈	脉	猫		貓	猫			么	幺	麽	么	幺	麽
3525	3525	1158	3541		3541	1159	1160	1161	3554			3554	1162	
霉	霉	黴	蒙		蒙	矇	懞	濛	弥			彌	瀰	
3570	3570	1163	3579		3579	1164			3860			3860	1165	
面	面	麵	蔑		蔑	衊			栖			棲	栖	
3890	3890	1166	3907		3907	1167			3909			3909	1168	
弃	棄	弃	千		千	韆			签			簽	籤	
3979	3979	1169	3990		3990	1170			4165			4165	1171	
秋	秋	鞦	曲		曲	麯			舍			捨	舍	
4182	4182	1172	4193		4193	1173	1174		4242			4242	1175	
沈	沈	瀋	升		昇	升	陞		适			適	适	
4341	4341	1176	4353		4353	1177			4381			4381	1178	
松	鬆	松	苏		蘇	囌			笋			筍	笋	
4392	4392	1179	4408		4408	1180	1181	1182	4419			4419	1183	
它	它	牠	台		台	臺	檯	颱	坛			壇	罎	

简体字		繁体字			简体字		繁体字		简体字		繁体字		
4531		4531	1184		4537		4537	1185	4715		4715	1186	
涂		塗	涂		团		團	糰	席		席	蓆	
4721		4721	1187	1188	4743		4743	1189	4744		4744	1190	
系		系	係	繫	纤		纖	縴	咸		鹹	咸	
4763		4763	1191		4781	4383	4781	4783	4782		4782	1192	
线		綫	線		像	象	像	象	向		向	嚮	
4855		4855	1193		4866		4866	1194	4869		4869	1201	
凶		兇	凶		锈		銹	鏽	绣		綉	繡	
4875		4875	1202		4893		4893	1203	4950		4950	1204	
须		須	鬚		旋		旋	鏇	岩		岩	巖	
5076		5076	1205		5131		5131	1206	5158	7622	5158	7622	
异		異	异		涌		湧	涌	于	於	于	於	
5184		5184	1207		5185		5185	1208	5164	6637	5164	6637	
郁		鬱	郁		吁		籲	吁	余	馀	余	餘	
5189		5189	1209		5224		5224	1210	5232		5232	1211	
御		禦	御		愿		願	愿	岳		岳	嶽	
5238		5238	1212		5247		5247	1213	5254		5254	1214	
云		雲	云		韵		韻	韵	灾		災	灾	
5264		5264	1215		5290		5290	1216	5328		5328	1217	
脏		髒	臟		扎		扎	紮	占		佔	占	
5359	6301	5359	6301		5387	6571	5387	6571	5427	7683	5427	1218	7683
折	摺	折	摺		征	徵	征	徵	只	祇	只	隻	祇
5430		5430	1219		5434		5434	1220	5438		5438	1221	
志		志	誌		致		致	緻	制		制	製	
5451	7981	5451	7981		5460		5460	1222	5476		5476	1223	
钟	鍾	鐘	鍾		周		周	週	朱		朱	硃	
5502		5502	1224		5528		5528	1225					
注		註	注		准		準	准					

考虑一个简体字对应多个繁体字的情况，繁体字的输入较之简体字的输入也有所改变。对于区位、国标、电报等输入方案，繁体字的输入编码有些变化，例如，按区位码输入，使用频度较高的繁体字的区位码与对应的简体字相同，而使用频度较低的繁体字则需要根据处于繁体字库的位置重新赋予区位码。对于拼音等字音输入方案，除用使用频率较高的繁体字替换输入提示区上对应的简体字外，还在提示区的另外位置上插入使用频度较低的繁体字。对于仓颉等字形输入方案，由于繁体字的字形与简体字不同，因此必须相应改变它们的输入编码。

3. 繁体字文章与简体字文章转换程序

繁体字文章与简体字文章转换程序可将繁体字文章自动转换为简体字文章；反之，亦可将简体字文章转换为繁体字文章，但对于一个简体字对应多个繁体字的情况，要考虑上下文关系，由用户选择需要的相应繁体字。在简体字文章转换为繁体字文章的过程中，当遇到一个简体字对应多个繁体字情况时，简体字自动转换为与它对应的使用频度较高的繁体字，并且光标停在这个繁体字上闪耀，此时用户可按规定键在这个位置上循环显示其它使用频度较低的繁体字，从而选择自己所需要的繁体字。当然，根据上下文关系，亦可将简体字文章自动转换为繁体字文章，例如，遇到“发”字时，转换程序查找上下文。若是出发的“发”字，则转换为“發”字；若是头发的“发”字，则转换为“髮”字，无须人工干预。

4.3 繁体字与简体字的兼容的中文系统

繁体字与简体字兼容的软件包仅仅是建立繁体字与简体字兼容的中文系统的一种尝试，还不能说是真正的繁体字与简体字兼容的中文系统。我认为，真正的繁体字与简体字兼容的中文系统，至少应能解决以下几个问题：

(1) 中文系统同时含有繁体字库和简体字库，并统一考虑繁体字与简体字的内码，以便可以混合显示或打印繁体字和简体字。

(2) 设置繁体字与简体字的内码对照表，以便实行繁体字与简体字之间的相互转换，既可以显示或打印繁体字，又可以显示或打印简体字。

(3) 鉴于目前的各种汉字输入方法难于统一，中文系统应兼容现存的各种汉字输入方案，既能输入简体字，也能输入繁体字，为此，应采取汉字输入编码方案自动生成方法。用户可随时把自己熟悉或喜欢的汉字输入方案，通过字处理软件或编辑程序建立汉字与键盘符号的对应关系，自动生成中文系统所支持的汉字输入编码方案。

(4) 为了推广中文软件的应用，要考虑繁体字中文软件移植到简体字中文系统和简体字中文软件移植到繁体字中文系统的种种问题，最终达到既可运行繁体字中文软件，又能运行简体字中文软件。

近年来，海峡两岸电脑工作者进行接触，双方商定筹设中文信息标准化工作组，下设中文标准编码、汉字字形开发、中文输入法、中文信息基础研究四个专业小组。工作组的宗旨是研究世界范围内汉字信息处理与标准化技术的发展，近期工作是商讨国际标准规范下的中文信息标准编码，并使其具有交换处理、显示及表示的良好功能，推动中文信息技术国际标准化工作。

尤为备受瞩目的是，海峡两岸将在国际标准组织会议中，就“基本多文种平面”(Basic Multilingual Plane)的配置，做更新的建议。

所谓基本多文种平面(B.M.P.)，就是国际标准化组织(ISO)依照文字的种类，不分民族及国家的区隔，把全世界各民族的文字与符号加以统一编码以应用于电脑通信的全球性标准化行动。基本多文种平面可分成四个区块，每个区块又分为

A(Alphabet)与 I(Icon)两部分，前者存放拼音字母，后者存放象形表意文字。目前，依照 ISO 的分配，I00 为日文集区，I10 为中国大陆字集区，I11 则存放朝鲜文字集。每一个 I 字集可存放 7500 个汉字。由于日本、朝鲜所用的汉字较少，所以不构成问题。但是对中文而言，不论中国大陆的 GB2312 或台湾的 CNS 11643 都不够用。

由于中国、日本、朝鲜所用的汉字有极高的重复率，例如，“日”字三国都有，但如果按目前的编码方式，却有三种不同的编码。因此，海峡两岸双方决议确定所谓“汉字字符集”(HAN CHARACTER COLLECTION)，从中国大陆简体字、繁体字、日文汉字、朝鲜文汉字、汉代汉语通用字表和台湾的 CNS 11643 等六种字符集中选字，收入所有的常用字。选字的原则是：以字形为主，并兼顾字义和字源。其中，

(1) 字形完全相同者，全部选入，采用单一编码；

(2) 字形相似者，例如，繁体字与简体字，虽采用同一编码，但字形各国可自行订定，即一码多字；

(3) 字形不同者，全部选入，各自编码；

(4) 异体字选字原则，尚未决定。

汉字的分级、定位和排序原则如下：

(1) 分级原则

按照使用频度把汉字分为常用字 7500 个，次常用字 7500 个，罕用字 4000 个，动态再定义区约 1000 字。

(2) 定位原则

常用字放在 I10 区和 I11 区的上部，次常用字放于 I10 区和 I11 区的下部，罕用字和动态再定义区放于 I00 区。

(3) 排序原则

可考虑先笔画后部首或先部首后笔画或笔画等排序方法，目前尚未确定。

基本多文种平面新建议的示意图如图 4.1 所示。

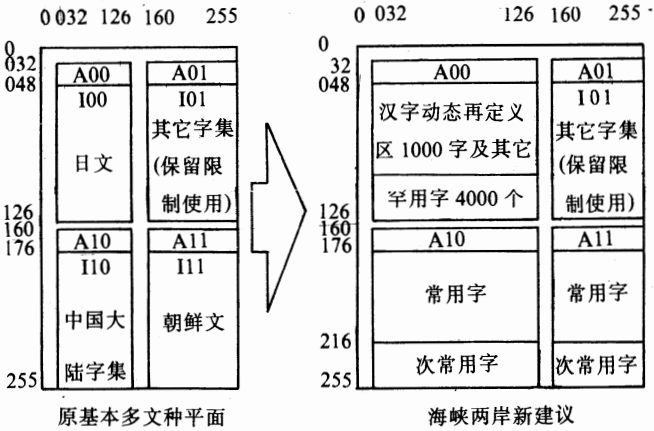


图 4.1　基本多文种平面的新建议

可以相信，随着中文信息国际标准化工作的深入发展，真正的繁体字与简体字兼容的中文系统必将诞生。

第五章　中文多用户系统与中文网络系统

随着计算机技术与通信技术的发展，两者的结合势在必行。多用户系统和网络系统是两者结合的必然产物。它们的形成过程，是从为解决远程计算、信息收集和处理而形成的专用联机系统开始，又在联机系统广泛使用的基础上，发展了以解决计算机之间相互通信和资源共享为目的的多用户系统和网络系统。

本章首先逐一简要介绍单用户系统、多用户系统和网络系统，并说明三种系统之间的主要区别，然后讨论中文多用户系统与中文网络系统的若干特殊问题。

5.1　单用户系统、多用户系统和网络系统

计算机系统按中央处理机(CPU: Central Processing Unit)和随机存取存储器（RAM: Random Access Memory）的使用情况可分为单用户系统、多用户系统和网络系统。三者的主要区别是：多用户系统和网络系统能实现通信和资源共享，而单用户系统不能；多用户系统的终端机执行程序时使用主机的 CPU 和 RAM，而网络系统的终端机执行程序使用本机的 CPU 和 RAM。在实际应用中，有时往往会把多用户系统和网络系统混为一谈。本节逐一简要介绍这三种计算机系统，并说明这三种计算机系统的主要区别。

5.1.1　单用户系统

单用户系统的主要特点是资源独占。单用户系统在同一时间内，仅允许一个用户操作该系统，系统内的所有资源在同一时间

仅供该用户使用。也就是说，单用户系统不允许多个用户同时使用 CPU、RAM、硬盘驱动器、软盘驱动器等硬件资源，也不允许多个用户同时使用软件资源和数据资源。如图 5.1 所示。

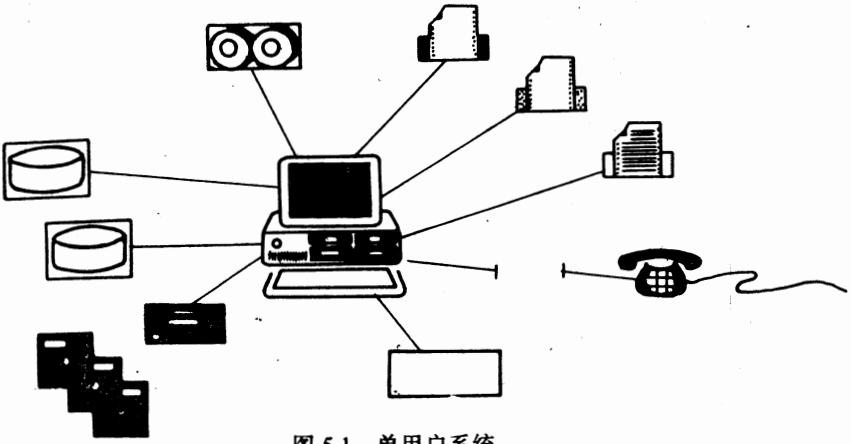


图 5.1　单用户系统

虽然单用户系统可以通过调制解调器(Modem)进行联机通信，被远端用户遥控操作，但某一用户在操作时，其他用户就不能使用这个系统。由于通信线路被独占使用，因此线路利用率很低，而且通信也不大方便。

有些单用户系统通过分时功能使 CPU 同时执行多项数据处理任务，称为多任务系统。例如，OS／2。在多任务系统中，CPU 及其它资源的管理是以任务为单位的，一个任务就是一个程序及其数据在 CPU 上的一次动态执行过程。在多任务系统中，允许使用多台打印机同时输出，或某一打印机假脱机做持续长时间的打印。然而，对于单用户多任务系统，仍只能被一个用户操作。

5.1.2　多用户系统

多用户系统是通过通信线路把主机与多台终端机连接起来的

计算机系统。多用户系统除联机通信外，还允许多个终端机用户同时使用该系统，共享主机的硬件资源、软件资源和数据资源，CPU 分时执行各用户的程序。如图 5.2 所示。

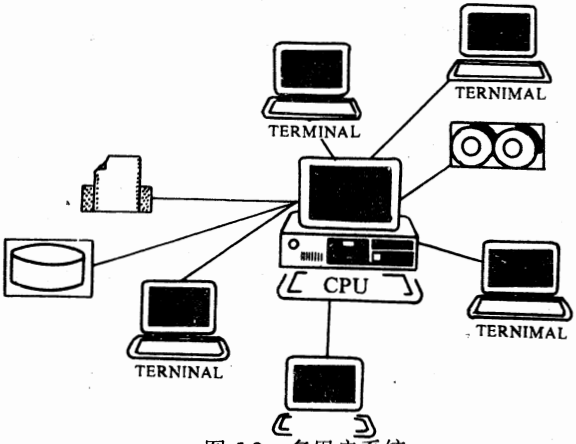


图 5.2　多用户系统

多用户系统的主机 CPU 负责控制所有的事情。当终端机用户提出使用要求时，会占用 CPU 的一些时间，因此每当增加一个新用户时，都会使 CPU 的执行速度变慢一些。当用户较多时，用户会感到执行速度很慢，需要在终端机前等待。为此，应限制同时操作的用户个数。

多用户系统的终端机通常只有显示器和键盘，很少拥有自己的 RAM 或磁盘存储器。虽然计算机亦可用作终端机，但仍然受主机的 CPU 控制。

多用户系统以大、中、小型机最为普遍；近年来，随着微型机的广泛使用，微型机多用户系统也逐渐出现，例如，UNIX，XENIX，MOS 等等。

5.1.3　计算机网络

计算机网络是通过通信线路把多台计算机连接起来的计算机

系统。在网络系统中，计算机之间可以相互通信，而且多个用户可以共享网络中的所有硬件资源、软件资源和数据资源。如图5.3所示。

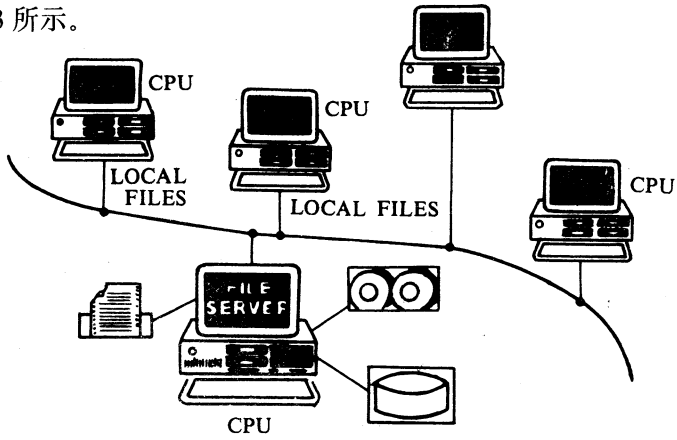


图 5.3　网络系统

计算机网络的文件服务器(主机)统筹管理程序和数据，用户可通过工作站（终端机）把文件服务器中的程序传输到本机的RAM中，当本机CPU执行程序需要数据时，可迅速存取文件服务器中的数据。除此之外，由于工作站拥有自己的CPU和RAM，因此具有单用户系统的功能，可当作单机独立使用。

计算机网络按其连接方式可分为集中式网络、分布式网络、环形网络或混合式网络。集中式网络通过通信线路把一台中心计算机与多台终端机连接在一起，常见的有星型结构和树型结构。分布式网络通过通信线路把分布在不同地点且具有独立功能的多台计算机互相连接起来。环型网络通过通信线路把多台计算机按环状连接起来。由于环型网络一般是各计算机分散控制，因此也可以属于分布式网络。

随着微型机的发展，局部网络日益广泛应用。局部网是在有限地区范围内把几台乃至几十台小型微型计算机及其外围设备连

接起来的计算机网络。连接距离一般在数百公尺到几公里，比远程网络连接要方便多了。它适用于一个中等规模地理区域里机关、工厂、医院、仓库、银行、学校等单位的管理调度、控制和通信。随着微型机价格的不断下降，微型机局部网络必将迅速发展。目前，常用的微型机局部网络有 Novell 网、Ethernet 网、3COM3+网、Plan 网、Omninet 网、IBM PC 网等。

5.1.4 三种计算机系统的主要区别

下面，从四个方面来讨论单用户系统、多用户系统和网络系统之间的主要区别。

1. 在同一时间内执行程序的方法

三种计算机系统在同一时间内执行程序的方法不同。

当单用户系统执行程序时，必须首先把磁盘中的程序及其数据送到 RAM 中，CPU 也被这个程序所占用，因此不能再做其它工作。

当多用户系统终端机用户执行程序时，虽然可以共享主机的程序和数据，但只能与别人共用 RAM，只能分时使用 CPU 来执行这一程序。

当计算机网络终端机用户执行程序时，通过网络系统把主机文件服务器内的相关程序传输到终端机的 RAM 内，而数据仍然留在文件服务器上。当终端机 CPU 执行程序需要用到数据时，再通过网络系统到文件服务器中存取。

总之，多用户系统终端机执行程序时使用主机的 CPU 和 RAM；网络系统终端机执行程序时使用本机的 CPU 和 RAM，但数据在主机文件服务器中存取；单用户系统执行程序时，当然使用本机的 CPU 和 RAM，数据也在本机中存取。

2. CPU 的使用情况

三种计算机系统 CPU 的使用情况不同。

单用户系统用自己机器上的 CPU 去执行自己的程序。

在多用户系统中，当终端机传送数据给主机 CPU 时，一次传送一个字符。在任何时间内，任何一个终端机用户键入一个字符，主机的 CPU 就必须对它做适当的反应，当这个字符进入主机 CPU 后再传送回终端机的屏幕上显示。这样一来一回就占用了主机 CPU 一次执行轮回，而第二个字符要进入主机 CPU，必须等到主机 CPU 下一次又转到这个终端机，这个字符才能进入主机 CPU。总之，主机 CPU 必须顾及各终端机的用户。

在网络系统中，当终端机用户键入一个字符时，本机自己的 CPU 对键入的字符做反应，并显示在本机的屏幕上。网络系统把各终端机所执行的程序转载到相应终端机各自的 RAM 上，然后各终端机的 CPU 再分别执行各自 RAM 上的程序。

3. 通信与资源共享

多用户系统能够实现通信和资源共享，而单用户系统只能独占式地联机通信，但不能共享资源。

所谓资源共享是指多个用户共享硬件资源、软件资源和数据资源。硬件资源包括打印机、绘图机、大容量外存储器等外围设备。软件资源包括系统软件、应用软件等。数据资源包括数据文件、记录等。多用户系统一般采用多用户操作系统来实现资源共享，而网络系统一般采用网络软件来实现资源共享。由于网络软件一般是在操作系统基础上安装的，它所形成的界面提供了网络功能，因此常把网络软件称为网络外壳。

在多用户系统或网络系统的主机上连接打印机，可通过以下两种方法来共享汉字打印功能:

(1) 采用带汉字字库的打印机;

(2) 多用户系统或网络系统本身带有传送图形的功能，例如，NOVELL 网的 CATCH 命令可将工作站的输出文本文件转换成点阵图形，而不是内码。

4. 解决多用户环境和网络环境的特殊问题

多用户系统和网络系统面临着在单用户环境下不会发生的潜

在问题，例如，数据保护、安全保密、多用户环境和网络环境下的出错处理等。对于这些问题，不同的多用户系统或网络系统、不同的软件层次、不同的软件，有着不同的解决方法。

数据保护是多用户系统和网络系统必须提供的一项功能。在多用户环境和网络环境下，如果多个用户恰好在同一时刻访问同一数据，那么更新的结果将是不可确定的：究竟接受了哪一个用户的更新呢？由于在多用户系统和网络系统中，几个用户要并行地访问共享的数据文件和程序文件，这一问题就有可能发生。这一潜在的问题称为冲突。如果不止一个用户在同一时刻试图增删改数据文件中的数据，就有可能发生冲突。假如允许发生冲突，可能没有一个用户的数据更新会获得成功并且数据保持不变，或者只有一个用户的数据更新获得成功，而所有其他用户还会以为他们也获得了访问并改变了数据。为此，多用户系统和网络系统必须实行数据保护，把对共享数据库的并行操作转换为串行操作，使多个用户表面上看起来同时在修改同一个数据文件，但实际上串行地执行多次更新，从而不会丢失数据，保持数据文件使之免受这种多个用户企图同时更新同一数据的影响。实行数据保护的措施是多种多样的，例如，对共享文件和共享记录加锁、设置文件的打开属性（独占或共享）和文件的访问属性（既能读又能写，只能读不能写，不能读也不能写）等。

安全保密是多用户系统和网络系统必须提供的另一项功能。在单用户环境中，安全保密问题不十分突出，而在多用户环境和网络环境中，程序和数据为多个用户共享，如果不提供完备的安全保密措施，后果是不堪设想的。为了建立、维护和确保每个用户数据的安全性、保密性和专用性，防止未授权的用户有意或无意的干预或访问。常见的安全保密措施有：注册保密、访问级别保密、数据加密等。注册保密是指：要想进入某一软件、某一应用程序、某一级菜单，或执行某项功能，或存取某一数据，必须正确地输入用户名、口令等注册值。访问级别保密是指：根据不

同的用户访问级别，提供不同的数据使用权和不同的操作类型；数据加密是指：把数据文件中的数据转换为密码形式。

多用户环境和网络环境下的出错处理是多用户系统和网络系统必须考虑的问题。在多用户环境和网络环境中除含有单用户环境中的出错信息之外，还含有特殊的出错信息，例如，试图锁定一个已被其他用户加锁的文件或记录。

5.2 中文多用户系统与中文网络系统

中文多用户系统和网络系统与西文多用户系统和网络系统无本质上的区别，关键在于在数据通信和数据处理中如何区分汉字和西文字符，从而在多用户系统和网络系统中实现中西文兼容。

本节主要讨论中文多用户系统和中文网络系统的几个特殊问题：中文终端、中文数据通信、中文联机仿真软件。

5.2.1 中文终端

终端是用来与计算机系统进行通信的一种输入输出设备，是人机对话的工具。

计算机系统往往有很多台终端，每台终端可以看成是一个用户。计算机系统由多用户操作系统或网络软件调动和控制各终端的操作。

早期的终端设备比较简单，例如，电传打字机、控制台打印机等，用来输入输出信息，称为哑设备。现在的终端设备往往采用 CRT 显示器，因此常称为显示终端。

近年来，随着微处理机技术的发展，终端的处理功能有了很大提高，出现了灵巧终端（Smart Terminal）和智能终端（Intelligent Terminal）。这两类终端本身均带有处理机，因而具有信息处理能力。

智能终端除了具有一般终端的功能外，往往还有一种或几种

智能处理能力。例如，图形处理终端具有图形输入输出、图形识别、图形修改等功能；语音对话终端具有语音识别、声音输入输出等功能；还有一些智能终端具有自动管理通信、自动检测、自动控制等功能。

中文终端（或称汉字终端）也是一种智能终端，它具有汉字输入、汉字显示、汉字打印、汉字屏幕编辑、汉字文件管理等功能。中文终端的硬件结构一般是由下列几部分组成的：中央处理机、汉字输入键盘、汉字显示器、通信接口、汉字印刷机、RAM 或 ROM 存储器、软盘驱动器等。在汉字终端中装入汉卡，可以大大地加快汉字存取速度，尤其是带有字符信息存储器的汉卡，可以达到西文终端同样的速度；更重要的是，由于显示方式是字符方式，不必修改 DOS 的 INT 10 显示模块，因此一般的仿真软件基本上不用修改便可运行，容易和 IBM、HP、VAX 等大中小型机联机。

汉字终端主要有下列两种功能：

(1) 汉字输入输出功能

汉字终端具有汉字与西文字符混合输入输出的功能。从汉字终端输入汉字，它把汉字输入码转换为内部码，传送给主机；汉字终端从主机接受汉字内部码，由汉字终端转换为汉字字形输出。

(2) 通信功能

汉字终端和主机之间必然要有通信接口：一是硬件上的通信接口，就是指终端和主机之间有一条信息传送的通路；二是终端和主机多用户操作系统或网络软件之间的软件通信接口，就是指软件规定的通信方式，如中断方式和询问方式等，以及软件规定的一些通信的控制代码。汉字终端的输入输出信息必须与多用户操作系统或网络软件及其它系统软件兼容。

综上所述，终端和微型计算机是两个不同的概念。终端仅是具有人机对话功能的输入输出设备，尽管它在硬件结构上与微型

机没有多大区别。两者本质上的区别在于：微型机有独立的处理能力，有完整的属于本身的操作系统和各种应用软件；而终端的所有软件都是为人机对话和各种智能而设计的，是在主机的多用户操作系统或网络软件控制下运行的。尽管目前国内外有不少微型机兼作终端用，例如，把微型机用作汉字终端。我们可把具有汉字处理能力的微型机系统看作是汉字智能终端与微型机的结合体，它们之间实质上仍然是各自独立的。

5.2.2 中文数据通信

通信就是用特定的方法通过介质或传输线将信息从一地传送到另一地的过程。通信可分为模拟通信和数据通信两大类。随着现代科学技术的发展，数据通信越来越成为通信技术的主流。

数据通信利用通信系统对二进制编码的字母、数字、符号以及数字化的声音、图象信息所进行的传输、交换和处理。通常是以计算机为中心，通过线路和终端直接连接起来形成联机系统，终端所产生的数据能及时地传送到中央处理机进行处理，而处理后的结果又能马上返送给终端。

中文数据通信与西文数据通信无本质的区别。两者的区别主要在于中西文字符编码集不一致。西文字符是七位或八位单字节编码，而汉字一般是七位或八位双字节编码。为了防止编码集发生重叠，发生二义性问题，中文数据通信主要要解决下列问题：

(1) 区别汉字和西文字符；

(2) 区别图形字符和控制字符；

(3) 区别不同代码体系中的字符。

同中文数据处理一样，中文数据通信解决上述问题的关键是汉字内码的设计和选择。

汉字内码是计算机内部加工处理用的汉字代码。按照用途可把汉字内码分为存储码，运算码、传输码三种。存储码用于存储汉字信息，用作汉字的机内表示和在磁记录媒体中的表示。运算

码用于参与各种操作运算。传输码用于计算机之间的通信（亦称通信码），或用于在计算机内部各部件之间传送汉字信息，比如，传送给显示器或打印机的内部传输码。为了提高汉字处理的效率，在允许的情况下，尽量把这三种汉字内码统一起来，例如，采用带标识位的二字节汉字内码。

然而，在某些联机通信系统中，为了实现信息传输，往往把每个字节的高位用作奇偶校验位，因而不能采用带标识位的二字节汉字内码作为通信码或传输码。例如，在微型机系统中，终端和主机的连接多半采用 RS 232 接口，而终端驱动模块常常采用七位传输码，第 8 位用作奇偶校验位。在这种情况下，如果不打算修改终端驱动模块而接收和传输汉字代码，在传输前后必须进行代码转换。通常在联机通信时，经过转换采用七位字节的通信码或传输码，例如，带标识码的两字节汉字内码、带标识码的三字节汉字内码、带标识码的四字节汉字内码（CCDOS 的通信管理模块就是用这种汉字内码作为传输码）。带引导码的汉字内码等（见 2.3.3 节）；而在进入主机或终端设备后，再转换采用带标识位的二字节汉字内码，供数据处理用。

为了区别图形字符（数据字符）和控制字符，常常利用转义字符 ESC 或 DEL 符来标识数据字符串中出现的控制符，即在每一个控制符前面都加一个转义字符 ESC 或 DEL，既可区别数据字符和控制字符，又可区别汉字和西文字符。例如，

DEL STX（7FH 02H）开始一个汉字数据块

DEL ETB（7FH 17H）结束一个汉字数据块

这是一种透明传递方式，就是说，控制符作为数据传输而不执行它们的控制功能。

究竟采用什么方法来区别不同种类的字符，视具体通信系统而定。不论采用什么方法，都应尽量压缩传输信息的长度，尽量减少代码转换的次数，以提高传输效率。

5.2.3 中文联机仿真软件

联机仿真是实现不同系统的主机与终端的联机通信的技术。中文终端与西文计算机系统的联机通信，有的是在计算机系统原西文终端上改造而成的，有的是在原仿真软件或硬件基础上改造而成的。这些均是通过中文联机仿真软件配合硬件来实现的。

在多用户系统或网络系统中实现中西文兼容，有两种方法：

1. 对主机系统(多用户操作系统或网络软件) 进行改造。这需要花许多力气去分析西文系统，然而，往往缺乏西文系统的设计资料，因此用这种方法去实现中西文兼容是很困难的。

2. 采用仿真技术。它包括规程仿真和终端仿真两种。

(1) 规程仿真

规程仿真技术是在不改变主机的终端设备的条件下，在主机和终端之间增加一个通信规程仿真器，来实现不同计算机系统的主机和终端的联机通信。通信规程仿真原理图如图 5.4 所示。

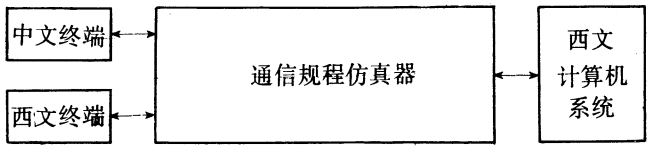


图 5.4 通信规程仿真器原理图

通信规程仿真器是不同计算系统的终端与主机的接口。它与中文联机仿真软件配合来实现通信规程。通信规程是在通信进程间控制信息流的一些约定。

中西文通信的主要差别在于信息格式和信息内容不同。中文通信规程仿真过程是：当中文终端向主机发送中文信息时，经过通信规程仿真器将中文代码转换为能被西文主机及其通信规程所接受的西文代码形式，送到主机去处理。当主机处理完后再经过通信规程仿真器把西文代码形式转换成中文代码形式，从而在联机通信过程中实现中西文兼容。

规程仿真一般是在中文终端基础上设计的，因此只需在通信控制规程一级进行仿真即可，无需对终端和主机作较大改动。

(2) 终端仿真

终端仿真是指在终端上通过仿真接口（软件或硬件或两者的结合）来仿真不同计算机系统的主机的原有终端功能，从而实现不同计算机系统的主机和终端的联机通信。

中文终端仿真的关键在于能仿真出原计算机系统的西文终端的全部功能，并把中文信息格式转换成能被主机接受的西文信息格式。

计算机系统原来都配有自己的西文终端来接收和发送信息。中文终端要想与西文计算机系统接收和发送中文信息，必须改造中文终端，使中文终端与主机接口界面和原西文终端与主机接口界面一样。也就是说，中文终端能够具有计算机系统的原西文终端的全部功能，而且能与主机进行中文信息的通信。中文终端仿真原理图如图 5.5 所示。

中文终端仿真需要在已有的中文终端和主机上做必要的软硬件改动。为此，必须熟悉中文终端、被仿的西文终端以及有关的硬件性能指标和软件功能模块。中文联机仿真软件的主要功能是实现中文信息的代码转换。

代码转换要根据主机代码体系来建立中文和西文的代码序列。例如，如果主机是八位 EBCDIC 代码体系，则代码转换要

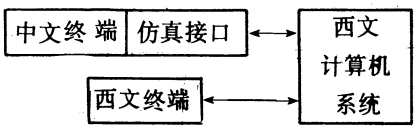


图 5.5　中文终端仿真原理图

实现中文代码和西文代码转换成 EBCDIC 代码体系下的代码。除此之外，还要求中文代码能映射到主机的合法代码区内，即要求中文代码为主机的合法代码。

第六章　中文程序语言与中文数据库管理系统

中文程序语言与西文程序语言、中文数据库管理系统与西文数据库管理系统的主要区别是：中文程序语言和中文数据库管理系统具有汉字处理功能。除此之外，它们保持西文程序语言和西文数据库管理系统的原有全部功能。

鉴于上述原因，本章一方面介绍程序语言和数据库管理系统的基本概念；另一方面，从汉字字符集、汉字名字、汉字数据、汉字的运算、汉字的比较、汉字的排序、汉字的查找、汉字的输入、汉字的输出、汉字文件处理、与汉字有关的函数、汉字注释等方面，概括和总结中文程序语言和中文数据库管理系统的特点，并分别讨论中文 BASIC 语言、中文 FORTRAN 语言、中文 PASCAL 语言、中文 dBASE 数据库管理系统的汉字处理功能。

6.1　程序语言概要

6.1.1　程序语言的发展历史

按照程序设计语言的发展历史，可把程序设计语言分为五类:

1. 面向机器的低级语言（50 年代）

例如，机器语言、汇编语言、宏汇编语言等。

2. 面向算法的高级语言（50 年代末，60 年代初）

例如，BASIC，FORTRAN，ALGOL，GOBOL，PL／I 等。

3. 结构化的程序设计语言（60 年代末，70 年代初）

例如，PASCAL，Ada 等。Ada 语言是美国国防部提出的一种语言，适用于数值计算、系统程序设计、实时控制和并行处理等方面。先从若干种招标语言中选出红色、绿色、黄色、蓝色四种语言，又从中选出红色语言和绿色语言两种。最后选出绿色语言，在此基础上改造成 Ada 语言。它是以世界上第一个女程序员的名字命名的。

另一方面，此时，某些特定的非过程语言被开发出来，作为辅助工具，例如，查询语言、报表生成器等，以提高软件生产力。

4. 以提高应用程序开发效率为特征的非过程语言(70 年代末，80 年代初)

非过程语言使人不必关心问题的解法和计算过程的描述，只要给出问题和输入数据，并指出输出形式，就能得到所需的结果。这里的非过程语言主要指面向对象的语言。

面向对象的语言致力于提高软件的生产率和可维护性，降低复杂性。它包括查询语言（例如，Query-by-Example，Online English，LOTUS1-2-3 等）、报表生成器（RPG Ⅱ／Ⅲ，Oracle 等）、图形语言（例如，FREELANCE PLUS，SAS／GRAPH，Tell-A-GRAF 等）、模拟语言（例如，GPSS，SIM SCRIPT 等）、应用程序生成器（例如，Template，ADAM，ADF，DIF 等）、测试数据生成器（例如，METACOBOL 等）、参数化应用软件包（例如，MRP 等）等等。

5. 人工智能语言（80 年代）

例如，函数式语言 FP，LISP 等，逻辑语言 PROLOG，GHC，Elephant，Lucid 等。

尽管程序语言正从过程语言向着非过程语言发展，但这种发展尚不成熟，需要相当长的时间。因此，目前过程语言仍在不断发展，例如，ADA83 发展为 ADA9x，FORTRAN77 发展为

FORTRAN 80，dBASE Ⅲ PLUS 发展为 dBASE Ⅳ。

我认为，程序语言的发展不宜贪大求全，否则会走向反面。例如，ALGOL 60 向何处发展？一种方向是把它发展为“无所不包”的大型通用语言 ALGOL 68 和 PL／I，成为“公共汽车”语言；另一种方向是把它发展为 PASCAL，力求概念简单化，从而导致一代新的程序语言的诞生。

6.1.2 微型机上常用的程序语言

下面，简要介绍微型机上常用的几种程序语言。

1. 汇编语言

汇编语言是一种面向机器的低级语言。它是机器语言的符号表示形式。通常是为某种特定机器设计的，因此不同类型的机器上所配的汇编语言各不相同。

汇编语言中的指令可分为以下三类:

(1)基本指令

它与机器指令的关系基本上是一一对应的。它用助记符代替机器指令的操作码，用标识符代替机器指令的地址码（操作对象)。

(2)伪指令

伪指令通常不对应机器指令，用于向汇编程序说明某些有关信息；或者设置机器指令系统中没有的指令，例如，如果计算机中没有硬件乘法指令，则可由汇编语言定义一条乘法指令，由汇编程序解释执行。

(3)宏指令

宏指令与一组机器指令相对应。用户可在汇编语言程序中，根据需要自行定义和使用宏指令。具有宏功能的汇编语言有时也叫做宏汇编语言。

汇编语言把机器指令符号化，因此比机器语言易读、易记、易修改、但是，汇编语言程序必须经过汇编程序翻译之后才能为

机器所执行。

与高级语言相比，汇编语言程序具有节省内存、执行速度快，并可精细地控制和使用机器资源等优点，因此常常用于系统程序、实时控制程序和常用标准子程序的设计。

2. BASIC 语言

BASIC 是 Beginner's All-purpose Symbolic Instruction Code（初学者通用符号指令代码）的缩写。

BASIC 语言简单易学，具有人机对话功能，便于修改和调试。它最初是为了便于教学而设计的，现在已广泛应用于各个领域。

BASIC 语言目前在微型机上有多种版本，例如，BASIC，BASICA，GWBASIC，True BASIC 等。

(1)BASIC

BASIC 的解释程序一般固化在微型机的 ROM 中，占 32KB 的容量。

(2)BASICA

BASICA 又称高级 BASIC，一般放在操作系统盘上提供给用户。它在 BASIC 基础上增加了绘图、奏乐、设置事件陷阱等功能。

(3)GWBASIC

GWBASIC 是在 BASICA 基础上发展来的，为 16 位微型机提供了良好的功能，提高了 BASIC 向上兼容的级别。

(4)编译 BASIC

上述 BASIC 语言版本均是解释性语言，BASIC 程序是通过解释程序逐句解释执行的。编译 BASIC 是编译性语言，BASIC 程序是通过编译程序翻译成机器代码后执行的，从而加快了执行速度，并使源程序加密。

(5)True BASIC

Ture BASIC 是一种结构化的程序设计语言。它没有标号，

因此转移不再根据行号为转移目标。在绘图、奏乐等方面效率也大大提高。而且，解释程序和编译程序并存。它还提供了较强的程序库。

3. FORTRAN 语言

FORTRAN 是 FORmula TRANslator（公式翻译）的缩写。

FORTRAN 是一种分块并列结构的面向过程的高级语言。它的发展经历了 FORTRAN Ⅱ，FORTRAN Ⅳ，FORTRAN77，FORTRAN80 几个主要阶段。

FORTRAN Ⅱ和 FORTRAN Ⅳ主要适用于数值计算。FORTRAN Ⅳ在 FORTRAN Ⅱ的整型和实型数据类型基础上，扩充了双精度型、复型、逻辑型和文字型数据类型。

FORTRAN 77 除了适用于数值计算外，还适用于非数值运算领域，增加了结构化语句，便于实现结构化程序设计。

4. COBOL 语言

COBOL 是 COmmon Business Oriented Language（面向商业的通用语言）的缩写。

COBOL 是一种适合于商业及数据处理的类似英语的程序设计语言。它的主要功能是描述数据结构和分析处理大批量的数据，包括对各种类型的数据进行收集、存储、传送、分类、排序、计算、打印报表等。大多数网状模型的数据库系统都采用 COBOL 作为宿主语言。使用这种语言，可以使商业数据处理的过程用标准的形式予以精确表达。

5. PASCAL 语言

PASCAL 是以一位十七世纪法国数学家命名的。

PASCAL 是系统地体现由 E · W · Dijkstra 和 C · A · R · Hoare 定义的结构程序设计概念的第一个语言。它是程序设计语言发展过程的一个里程碑。

PASCAL 是在 ALGOL 语言基础上发展起来的，但功能更

强，而且更容易使用。它既保留了 GOTO 语句，同时又增加了大量的控制结构。它提供了丰富的数据类型和构造数据结构的方法。除了整型、实型、布尔型数据外，还增加了字符类型、子域类型、记录类型、文件类型、集合类型和指针类型。

尤其有意义的是，PASCAL 的数据类型和控制结构同数学中某些集合运算在逻辑结构上存在着一致性。例如：

(1)由不同型对象组成的排列

数据类型：记录

控制结构：复合语句

数学：笛卡儿积

(2)由同型对象构成的重复

数据类型：向量或序列

控制结构：循环语句或重复语句

数学：正规事件中的 * 运算，即 A^*

(3)由不同型对象构成的选取

数据类型：类型析取

控制结构：分情形语句或条件语句

数学：集合并

(4)由顺序不确定的对象构成的集合

数据类型：集合

控制结构：不确定语句

数学：集合

(5)指向不规则顺序对象的成分

数据类型：指针

控制地构：GOTO 语句

数学：无

PASCAL 语言广泛应用于微型机和其它机器上。它既可用于科学计算，又可用于数据处理，还适合编写系统软件，也是极好的教学工具。

TURBO PASCAL 在标准 PASCAL 基础上做了扩充，具有字符串类型，并具有较强的全屏幕编辑功能和快速编译功能。近年来在微型机上十分流行。

6. C 语言

C 语言是从以 BCPC 语言为基础的 B 语言发展来的。由于 UNIX 操作系统是采用 C 语言编写的，因此，随着 UNIX 的普遍应用，C 语言也得到广泛使用。

C 语言是一种系统程序设计语言，它适用于编写操作系统、编译程序、解释程序等系统软件。

C 语言一方面具有高级语言的特点，具有先进的控制结构和数据结构，C 语言程序容易编写，容易阅读；另一方面，C 语言具有汇编语言的功能，可以处理计算机直接操作的大多数数据，直接完成硬件的算术或逻辑运算，因而 C 语言程序在空间和时间效率上都能和汇编语言程序媲美，目标程序质量较高，适用于编写系统程序。C 语言在简洁性与实用性、可移植性与高效率之间的矛盾得到了较好的解决。

微型机上常用的 C 语言版本有 Microsoft C，Turbo C，Lattice C，Instant C，C++等。

7. LISP 语言

LISP 是 LISt Processor（表处理）的缩写。

LISP 是一种函数式非过程语言，广泛应用于人工智能领域。

LISP 语言具有如下特点：

(1)LISP 程序的形式通常是一串函数定义，后跟一串带有参数的函数调用，函数之间的关系在调用执行时才体现出来。LISP 没有语句概念，也没有分程序结构或其它语法结构，语言中一切成份都是以函数的形式给出。

(2)LISP 在函数的构造上，同数学上递归函数的构造方法类似，从几个原始函数出发，通过一定的手段（例如，函数的复

合、递归等）构成新的函数。

(3)在 LISP 中，程序和数据在形式上是等价的。LISP 的唯一数据结构是 S–表达式，而程序本身也是用 S–表达式写的，因此可以把程序当作数据处理，也可以把数据当作程序来执行。

(4)递归是 LISP 的主要控制结构，而不象其它一些程序那样以迭代（循环）为主要控制结构。它的递归处理是基于递归定义的数据结构。

8. PROLOG 语言

PROLOG 是 PROgramming LOGic（逻辑程序设计）的缩写。

PROLOG 是一种逻辑型超高级语言，广泛应用于人工智能领域。

PROLOG 用不着规定计算机在什么时候应当做什么，也用不着告诉计算机如何求解问题，用户只要首先向计算机提供关于被解问题的知识和前提；然后直接向计算机提出要解的问题。根据这些知识和前提，PROLOG 会自动进行判断和推理，最后给出问题的解答。

PROLOG 具有如下特点:

(1)程序不需要说明运算的执行顺序，只需要描述清楚事物的逻辑关系。

(2)语句种类少，只有事实、规则和提问三种，容易为用户掌握。

(3)PROLOG 采用模式匹配和回溯技术自动地实现求解问题，从而具备了演绎推理的功能。

(4)适用于符号处理。

TURBO PROLOG 是一种编译型 PROLOG 语言，它比解释型 PROLOG 语言执行速度快，占内存少。

6.1.3 程序语言的语法、语义、语用

程序设计语言同自然语言一样，均包含语法、语义、语用三大要素。语法表示语言的形式或结构；语义表示语言的意义；语用表示语言的使用。

1. 语法

多年来，形式语言理论的发展为语法形式化描述奠定了基础。目前，很多程序语言采用巴科斯范式或语法图定义语法。因此，学习一点形式语言的基本知识，对阅读程序语言文本以及检查程序的语法错误是必要的。下面，简要介绍元语言、文法、推导与归约、语言、上下文无关文法、巴科斯范式、语法图、语法树、二义性等形式语言的基本概念。

(1)元语言

用于定义语言的语言称为元语言。元语言中的符号称为元符号。

例如，在英汉词典中，用汉语定义、解释和描述英语，因而汉语就是元语言，而英语是被定义的对象。词典中的分隔符、注解符等均是元符号。

(2)文法

文法是描述语言的语法结构的形式规则（即语法规则）。

由语言学家 Chomsky 在五十年代末引入的产生式文法，是一种描述能力很强的元语言。

产生式文法是一个四元组(V_N，V_T，P，S)，其中，V_N 代表非终结符集，V_T 代表终结符集，P 代表产生式集，S 代表开始符号。V_N、V_T、P 是有穷集，且 $V_N \cap V_T$ 为空集。产生式的形式为 $\alpha \rightarrow \beta$，α、β 为由终结符和非终结符组成的字符串，→是元符号，表示“可以用产生式的右部替换产生式的左部”。S 为非终结符。

例如，定义形式为 $a^nb^nc^n(n \geq 1)$的所有符号串的产生式文法如下：

①S→aBc

②B→b

③aB→aaBbC

④Cc→cc

⑤Cb→bC

其中，a、b、c是终结符，S、B、C是非终结符，S是开始符号。

(3)推导与归约

从文法的开始符号出发，反复连续使用产生式，用产生式的右部替换产生式的左部，经过一系列的替换，直到产生只由终结符组成的符号串为止。我们把这样一个替换序列称为推导，把它的逆序列称为归约。

在推导过程中，每一步所产生的符号串称为句型，最后一个只含终结符的句型称为句子。

例如，根据上述文法从开始符号推导出句子abc，aabbcc，aaabbbccc:

S⇒aBc　　产生式（1）

⇒abc　　产生式（2）

S⇒aBc　　产生式（1）

⇒aaBbCc　　产生式（3）

⇒aabbCc　　产生式（2）

⇒aabbcc　　产生式（4）

S⇒aBc　　产生式（1）

⇒aaBbCc　　产生式（3）

⇒aaaBbCbCc　　产生式（3）

⇒aaaBbbCCc　　产生式（5）

⇒aaabbbCCc　　产生式（2）

⇒aaabbbCcc　　产生式（4）

⇒aaabbbccc　　产生式 (4)

(4)语言

语言是由文法产生的所有句子的集合。

例如，由上述文法 G 定义的语言

$L(G)=\{a^nb^nc^n|^n$ 为正整数$\}$

$=\{abc, aabbcc, aaabbbccc, \cdots\}$

(5)上下文无关文法

Chomsky 把语言分为四个层次，0 型语言（短语语言）、1 型语言（上下文有关语言或上下文敏感语言）、2 型语言（上下文无关语言）和 3 型语言（正规语言），它们分别是由 0 型文法（短语文法）、1 型文法（上下文有关文法或上下文敏感文法）、2 型文法（上下文无关文法）和 3 型文法（正规文法）定义的。

程序语言一般是上下文无关语言，上下文无关文法适于描述程序语言的语法。

上下文无关文法对产生式文法中的每一个产生式 $\alpha\rightarrow\beta$ 都作下列限制:

①α 是单个非终结符;

②β 是任意非空符号串。

由此可见，上述文法不是上下文无关文法（实际上是短语文法），它所定义的语言也不是上下文无关语言（实际上是短语语言）。

例如，定义程序语言中常见的算术表达式的上下文无关文法如下:

<E>→<T>

<E>→-<T>

<E>→<E>+<T>

<E>→<E>-<T>

<T>→<F>

<T>→<T> * <F>

<T>→<T>/<F>

<F>→i

<F>→(<E>)

其中，<和>是元符号，用于把非终结符括起来，以便区别非终结符和终结符。

+，−，*，/，(，)，i是终结符，i表示常数、变量名等简单运算对象。终结符是被定义语言中的符号。

<E>，<T>，<F>是非终结符。非终结符亦称为语法变量、语法单位、语法实体、语法类等，它代表被定义语言的一定语法范畴。例如，算术表达式、赋值语句、过程等。<E>、<T>、<F>分别代表算术表达式、项、因子等语法范畴。

<E>是开始符号。开始符号是一个特殊的非终结符，它代表被定义语言中最终感兴趣的语法范畴。<E>代表算术表达式这个语法范畴。在程序语言中，最感兴趣的语法范畴通常是程序。

(6)巴科斯范式（BNF）

BNF是Backus Normal Form（巴科斯范式）或Backus-Naur Form（巴科斯-脑尔范式）的缩写。

BNF是一种适于描述上下文无关语言的元语言，而现在大多数程序语言都是上下文无关语言，因此，目前很多程序语言都采用BNF定义语法。

BNF与产生式的主要区别是：产生式的左部必须是单个非终结符；使用元符号::=代替→；允许合并具有相同左部的产生式，共用一个左部，各右部之间用元符号|分隔。::=读作“定义为”或“由……组成”，|读作“或者”。

例如，用BNF定义上述的算术表达式如下：

<E>::=<T>|-<T>|<E>+<T>|<E>-<T>

<T>::=<F>|<T>*<F>|<T>/<F>

<F>::=i|(<E>)

(7)语法图

语法图是一个有向图，它是描述语法结构的有力工具。目前，很多程序语言采用语法图定义语法。

语法图是根据文法规则（产生式或 BNF）绘制成的。它具有形象直观的特点。

例如，图 6.1 给出根据上述算术表达式的文法绘制的一种语法图。

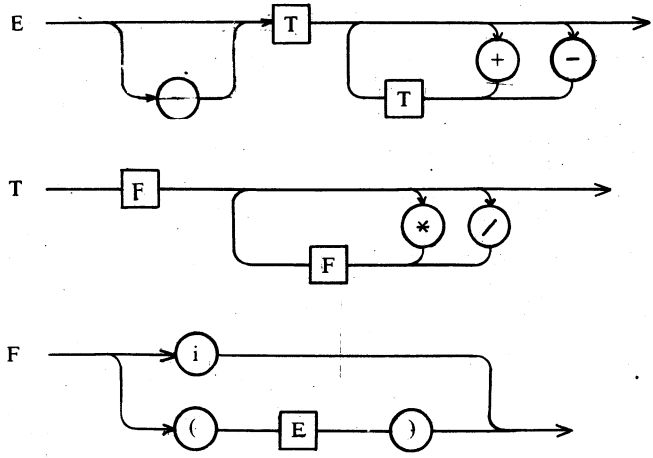


图 6.1　语法图

语法图中的矩形代表非终结符，圆形代表终结符。

(8)语法树

语法树是句子的语法结构的图形表示。根据 BNF，可从开始符号推导出句子，亦可把句子归纳为开始符号，归约是推导的逆过程。用语法树分析句子的过程是推导的逆过程，就是归纳过程。因此，语法树亦称为语法分析树。

例如，根据上述的算术表达式的 BNF 文法，从＜E＞推导出句子 (i * (i+i)) / i 的过程如下：

<E>⇒<T>
⇒<T>/<F>
⇒<F>/<F>
⇒(<E>)/<F>
⇒(<T>)/<F>
⇒(<T>*<F>)/<F>
⇒(<F>*<F>)/<F>
⇒(i*<F>)/<F>
⇒(i*(<E>))/<F>
⇒(i*(<E>+<T>))/<F>
⇒(i*(<T>+<T>))/<F>
⇒(i*(<F>+<T>))/<F>
⇒(i*(i+<T>))/<F>
⇒(i*(i+<F>))/<F>
⇒(i*(i+i))/<F>
⇒(i*(i+i))/i

用语法树分析句子 (i*(i+i))/i 的过程如图 6.2 所示。

我们看到，语法树是一棵倒置的树，树叶是终结符，树的内部结点是非终结符，树根是文法的开始符号。

(9)二义性

如果一个文法至少有一个句子存在两棵不同的语法树，那么该文法是二义性的。换言之，如果一个文法至少有一个句子存在两种不同的推导，那么该文法是二义性的。

例如，下列文法是具有二义性的文法:

<语句>::=IF <条件> THEN <语句>|
IF <条件> THEN <语句> ELSE <语句>

这是因为: 下面的句型

IF <条件> THEN IF <条件> THEN <语句> ELSE <语句>

对应有两棵不同的语法树（如图 6.3 所示）。

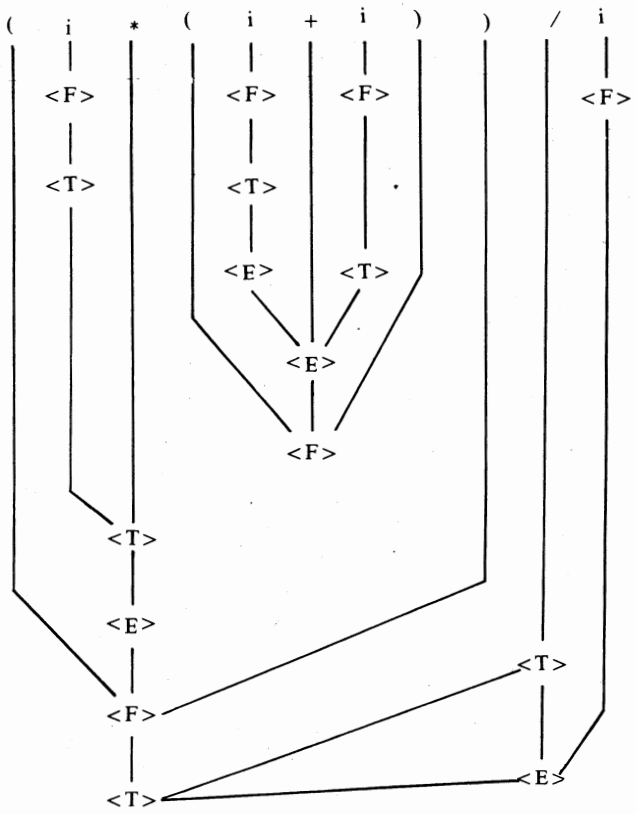


图 6.2　(i * (i+i)) / i 的语法树

同样，这一句型也存在有两种不同的推导，如下：

<语句> ⇒IF <条件> THEN <语句> ELSE <语句>

⇒IF <条件> THEN IF <条件> THEN <语句>

ELSE <语句>

<语句>⇒IF <条件> THEN <语句>

⇒IF <条件> THEN IF <条件> THEN <语句>

ELSE <语句>

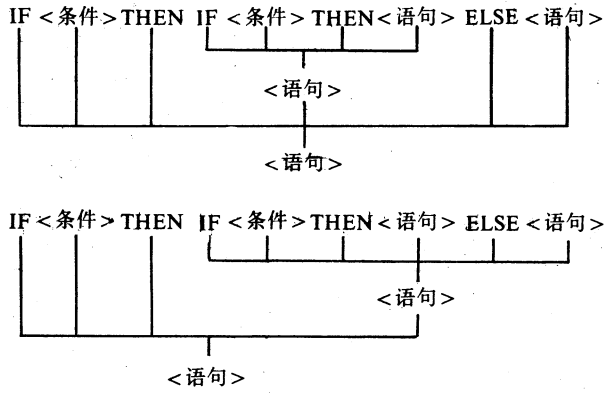


图 6.3 IF <条件> THEN IF <条件> THEN <语句> ELSE <语句>的两棵语法树

究竟 ELSE 子句属于哪一个条件语句？在所有包含条件语句的程序语言（例如，PASCAL）中都倾向于规定：ELSE 子句属于靠得最近的无 ELSE 子句的 IF；或者说，ELSE 必须匹配最后那个未得到匹配的 THEN。也就是说，为了消除上述文法的二义性，应按第二棵语法树或第二种推导识别上述句型。

2. 语义

近年来，国际上出现了许多形式语义的描述方法，例如，文法型语义：W 文法、缀词文法、属性文法、转换文法等；数学型语义：操作语义、指称语义、公理化语义、代数语义等；程序逻辑型语义：动态逻辑、算法逻辑、递归程序逻辑、时序逻辑等。然而，由于语义形式化描述是相当困难的，至今大多数程序语言的语义仍采用自然语言来描述。

3. 语用

语用研究语言符号与使用者的关系，即语用表示在语言的各个记号所出现的行为中它们的来源、使用和影响。程序语言中的语用包括语言的实现技术、程序设计方法、程序语言的发展历史等。程序语言是通过实现来使用的，由于语用的缘故，程序语言语法和语义的实现不可能满足所有程序和所有数据。例如，一程序语言在不同机器上实现时，数值表示范围往往不同。相对语法和语义来说，语用的发展则更晚。迄今为止，在绝大多数程序语言描述中尚未涉及到语用，或把语用纳入语义描述。

6.1.4 程序语言的数据结构

不同的程序设计语言提供了不同的数据结构和控制结构，我们在本节和下节中将只讨论大多数程序语言所共有的基本数据结构和基本控制结构。由于在不同的程序语言中，这些基本数据结构和基本控制结构的表示形式也不同，因此我们给出的表示形式只是一种示意而已，不属于任何一种具体的程序语言。

从某种意义上说，数据结构与控制结构的相互作用，就产生了程序。

程序语言的数据结构是由数据类型定义的。数据类型是由值集（它的定义域）和操作集组成。也就是说，一个数据的类型既决定了它所能具有的值，又决定了对它所能执行的操作。操作是通过操作符或函数实现的。

实质上，表达式可用函数记号来表示，例如，中缀形式（操作符在两个操作对象之间）a+b 可写为函数形式+(a，b)。因此，可认为中缀形式是函数形式的另一种表示方法，这仅仅是习惯和方便的问题。反过来，我们亦可认为函数名是一种特殊的操作符，参数是一种特殊的运算对象。由此可见，两者只不过是表示方式不同而已。

程序语言中的数据类型大致可分为纯量类型、构造类型、引

用类型三大类。下面将分别介绍这些数据类型，着重强调一些值得注意的地方。

1. 纯量类型

纯量类型是定义域由各不可分的分量组成的一种类型，纯量不含有任何可独立存取和操作的分量。

程序语言中常见的纯量类型有整型、实型、字符型（或字符串型）和逻辑型（布尔型）。这些类型常常是计算机硬件支持的类型。此外，程序语言通常还提供其它一些纯量类型，例如，枚举类型、复型、双精度型、日期型等，尽管硬件未直接支持它们，但提供这些类型有时是很方便的。

(1)整数类型

整数的取值范围因计算机而异。对整数的常见操作有：加、减、乘、除、比较（小于、大于、等于、不等于、小于或等于、大于或等于）、求绝对值、求幂、求模（求余数）、求最大值、求最小值等。整数除法产生整数结果，通常余数被忽略。

(2)实数类型

实数的取值范围因计算机而异。对实数的常见操作有：加、减、乘、除、比较（小于、大于、等于、不等于、小于或等于、大于或等于）、求绝对值、求幂、取整、自然对数、平方根、求最大值、求最小值、求模（求余数）、舍入、正弦、余弦、正切、反正切等。实数操作的精确度也因计算机而异。

(3)字符类型

有的程序语言的字符类型的值是指单个字符，而有的程序语言的字符类型的值是指字符串，后者有时也称为字符串类型。

在中文程序语言中，字符包括汉字和西文字符。汉字的取值范围取决于所采用的汉字字符集。西文字符的取值范围取决于所采用的西文字符集，例如，ASCII 字符集。在大多数的情况下，ASCII 字符是指可打印的 ASCII 字符；但在少数情况下，也包括非打印字符。

对字符的常见操作有：比较（小于、大于、等于、不等于、小于或等于、大于或等于）、字符与码值的转换、大小写字母的测定与转换等。

对字符串的常见操作有：连接、截取、插入、替换、删除、比较（小于、大于、等于、不等于、小于或等于、大于或等于、属于）、求长度、字符与码值的转换、大小写字母的测定与转换等。

字符或字符串之间的比较，实质上是比较码值的大小。

(4)逻辑类型

逻辑类型有时亦称为布尔类型。

逻辑值（布尔值）只有两个：真和假。通常用 true 或.T.表示真值，用 false 或.F.表示假值。常见的逻辑操作有：与(AND)、或（OR)、非（NOT)。

相同类型的数据（例如，整型、实型、字符型）进行比较，操作结果是逻辑真值或假值。

(5)枚举类型

枚举类型的定义（通过类型说明）列出或枚举出该类型数据所能具有的值。例如，

```
TYPE
  day=(Sunday, Monday, Tuesday, Wednesday Thursday,
       Friday, Saturday)
```

枚举类型的常见操作是：等于、不等于和赋值。如果程序语言规定枚举类型定义中的枚举值是有序的，则枚举类型是有序类型，它的操作可为：赋值、后继（succ)、前驱（pred)、比较(=、<>、<=、>=，<，>)。例如，下列表达式为真：

succ (Monday)=Tuesday

Pred (Thursday)=Wednesday

succ(Saturday)无定义

succ (pred (Tuesday))=Tuesday

Wednesday < Friday

实质上，纯量类型均可表示为枚举类型。例如，整数类型定义可枚举出取值范围内的所有整数，实数类型定义可枚举出取值范围内的所有实数，字符类型定义可枚举出汉字字符集中的所有汉字和西文字符集中的所有西文字符，逻辑类型定义如下：

```
TYPE
    boolean = (false, true)
```

2. 构造类型

构造类型与纯量类型的主要区别在于：构造类型的变量包括若干个分量，每个分量都是一个变量，它可以具有纯量类型或构造类型。在最低层上，构造类型变量的各个分量均具有纯量类型，这些分量可以象简单变量那样被赋值，或者用在表达式中。程序语言中常见的构造类型是数组类型和记录类型。

(1)数组类型

数组是具有相同类型的变量的有序集合。一维数组可称为向量。

数组分量亦称为数组元素或下标变量，它是由下标值来索引的（即根据下标值访问数组分量）。除整数型和实型外，有的程序语言还允许用枚举类型作为下标类型。例如，在 PASCAL 语言中，

```
TYPE
    direction = (x, y, z);
    vector = ARRAY [direction] OF real;
```

其中，direction 为由枚举值 x，y，z 定义的枚举类型，它被用作下标类型来定义数组类型 vector。

除数组分量参加运算外，有的程序语言还提供了对整个数组的操作。例如，

A := B

把数组 B 的所有元素分别赋值给数组 A 对应位置的元素中去

(A 和 B 是具有相同维数的数组)。

A：＝0

把数组 A 的每一个元素都置为零。

if A＝0 then……

如果数组 A 的所有元素都是 0，那么执行 then 后面的那个语句。

A：＝B＊C

把数组 B 和数组 C 对应元素的乘积赋值给数组 A 的对应元素中去（A，B，C 是具有相同维数的数组)。

APL 语言提供了丰富的数组操作，例如，矩阵乘法、矩阵求逆、解线性方程组等操作。

(2)记录类型

记录是由具有不同类型的若干个分量组成的。分量类型可以是任何类型，甚至可以是另一个记录类型。

记录分量亦称为域或字段，它一般是通过域名或字段名来访问的，有时是通过记录变量名连同域名一起访问的。

记录类型的定义（通过记录类型说明）描述了如何构造一给定类型的记录，可把这种定义想象为用于构造单个记录的样板或模型。例如，

```
TYPE person＝
    RECORD
        name: ARRAY [1 · · 8]  OF  char;
        sex:(Male, Female);
        age: interger
    END
```

在有些程序语言中，特别是数据库语言，除记录分量参加运算外，还提供了以记录为单位的操作，例如，观察记录、追加记录、插入记录、删除记录、修改记录、查找记录、记录的分类和索引等。

(3)析取类型

允许一个变量在不同的情况下取不同类型的值有时是方便的。某些程序语言允许类型定义规定合法值是两个或多个类型的选择，我们称这种类型为析取类型。

例如，假定已定义了枚举类型:

TYPE ManName= (Tom, Dick, Harry)

TYPE WomanName= (Mary, Diane, Anastasia)

则可定义析取类型 Name，该析取类型的变量可以取集合{Tom, …, Anastasia}中的任一值，如下:

TYPE Name=UNION (ManName, WomanName)

假设 x 是析取类型 Name 的变量。那么，如何确定 x 当前是属于类型 ManName，还是属于类型 WomanName 呢？由此可见，判别变量是属于析取类型定义中的哪一个类型的值，是析取类型的关键操作。

在 ALGOL W 语言中，引入了一种操作符 is，用来确定变量的值是否具有给定的类型，如下:

x is ManName

它表示: 每当变量 x 含有类型 ManName 的值就为真。根据 x 中所含值的类型，可使程序执行不同的动作。

在 PASCAL 语言中，是用变体记录来代替操作符 is 而得到判别析取的。为了定义析取类型，可编写如下程序:

```
TYPE
    Gender=(Man, Woman)
    Name=RECORD
            CASE kind: Gender  OF
              Man: (He: ManName);
              Woman: (She: WomanName)
            END
```

若 kind 等于 Man，则 He 域是有效的；若 kind 等于 Woman，

则 She 域是有效的。因此，可用对标志域 kind 的判别来代替 is 操作符，如下：

x · kind = Man

以 CASE 开始的记录说明部分称为变体部分，它位于普通的域定义（固定部分）之后。

3. 引用类型

在程序语言中，可把名字当作类型来处理，习惯上称为引用类型或指针类型。自然，引用作为一种类型，必须有引用变量、含有引用的变量、值引用函数以及返回引用当作值的函数，还必须有引用常量，变量名是最典型的引用常量。

对每个类型 T，都假定存在一个对 T 引用的类型，用 ↑T 来表示之。↑T 类型的变量所包含的值是含有类型 T 的值的变量的名字（或对变量的引用）。例如，↑integer 类型的变量的值是某个整型变量的名字。

同其它类型一样，引用类型也有相关的操作。除等于和赋值操作外，主要是间接引用操作，定义如下：

(1)若 X 是类型 ↑T 的值，则 X↑ 是 X 所引用的类型 T 的对象。

(2)若 Y 是类型 T 的一个变量，则 βY 是对 Y 的引用，即类型 ↑T 的值。

有些程序语言提供了显式引用类型。例如，PASCAL 对各种预先定义的类型提供了引用类型，即如果 T 是类型，则有一个类型 ↑T，它的变量包含类型 T 的对象名字。但是，PASCAL 仅提供了间接引用操作 X↑，而没有提供引用建立操作 βX。

有些程序语言未明显地提供引用类型。例如，FORTRAN 语言和 COBOL 语言均没有引用类型。尽管如此，引用的概念仍隐含在这些程序语言中。例如，赋值语句在给变量赋值时，需要对含有类型 T 的值的变量的引用 ↑T，这种引用操作不产生值，但却得到 ↑T 所命名的存储单元。同样，给数组元素或记录

的域赋值时，把结构存取都当成是产生一引用值的表达式，比如，赋值语句

A [i+5]：＝3.14;

首先对数组元素 A[i+5]进行下标地址计算，求得它的存储单元地址，然后把值 3.14 赋给该地址的存储单元中。又例如，过程调用在进行参数传递时，实在参数与形式参数结合的一种常用规则是引用调用，亦称为地址调用。这种结合规则，把实在参数的地址传送给形式参数，每当程序引用一个形式参数时，就直接存取作为实在参数被传递的变量。

下面，我们通过两个例子（图 6.4 和图 6.5）来说明含有显式引用类型的赋值。

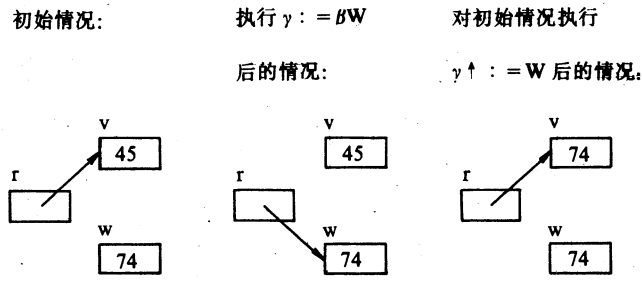


图 6.4　含有单个引用变量的赋值语句

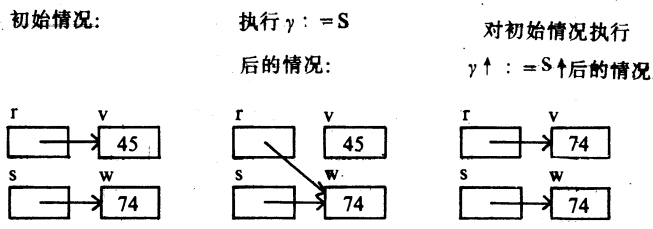


图 6.5　含有多个引用变量的赋值语句

假定 w 和 v 是类型 T 的变量，r 和 s 是类型 ↑T 的变量（T 为整型），其中，r 的初值为 v 的名字，s 的初值为 w 的名字。

图 6.3 给出了含有单个引用变量的赋值语句。r:=βw 把 w 的名字赋给 r，r↑:=w 把 w 的值赋给 r 命名或引用的变量。

图 6.4 给出了含有多个引用变量的赋值语句。r:=s（其中 r 和 s 都是 ↑integer 类型）在大多数程序语言中表示指针复制：改变 r 使得 r 和 s 含有相同的名字，r↑:=s↑把 s 所指向的变量的值赋给 r 所指向的变量。

象 ALGOL W 和 PASCAL 这样的一些程序语言，需要说明引用变量可以指向的变量的类型。而象 PL／1，BLISS 和大多数汇编语言不需要这种说明，一个指针可以指向任何类型的变量。这种不加任何限制的情况似乎会使语言的功能更强。但是，却因缺乏对变量指向什么的描述而使程序难以理解。

引用既是程序语言中最重要的和最强的功能之一，也是最难以捉摸和最危险的特性之一。通过引用可构造复杂的数据结构。引用是设计好结构的程序必不可少的重要工具。遗憾的是，引用也会产生不必要的复杂和不安全的数据结构，这种结构往往是很难理解的。因此，使用引用时必须小心谨慎。

4. 类型转换

程序语言往往有以下两种类型转换方式：

(1)显式类型转换

程序语言中提供类型转换函数，编写程序时引用这些函数去进行类型转换。例如，BASIC 中的 CHR$() 和 dBASE 中的 CHR() 把 ASCII 码转换为字符。

(2)隐式类型转换

对于两种不同类型值之间存在着一种自然的一致关系的情况，程序语言往往在语义描述中对类型转换做如下规定：

①把两种类型看成是析取类型，把对这两种类型数据的操作看成是析取类型上的操作。最常见的例子是整型和实数。在大多

数程序语言中都把这两种类型看作是析取类型

UNION (real, integer)

因此，对于整型变量 i 和实型变量 x，i+x 是允许的，只不过此处使用的加法运算符是一种新运算符，它是整数和实数之间的运算，而不是整数之间或实数之间的运算。

②硬性规定在某种场合下一种类型转换为另一种类型，这种隐式的自动转换称为强制。例如，对于上述 i+x，规定先把 i 的整型值转换成实型值，然后再做实型数的普通加法。强制只适用于某些场合。例如，FORTRAN 语言允许把实型值赋值给整型变量，赋值之前把实型值强制为一个整数；但不允许实型表达式作数组下标，而只能用整型表达式作下标。

大多数程序语言本身只包含几种隐式类型转换的规定，主要是通过提供类型转换函数来实现各种类型之间的相互转换。只有少数程序语言（例如，PL／1）所提供的强制几乎使任一类型都可强制为任一其它类型。

5. 隐式类型语言

如果在一程序语言中，每个变量的类型都是由一类型说明定义的，则我们称该语言为显式类型语言。

在隐式类型语言中，变量的类型是隐含的，不限制变量必须容纳已定义的类型的值。例如，在 APL 语言中，写出如下语句序列:

x←3

x←7 9 10

前一个语句置 x 为整数，后一个语句置同一个变量 x 为整数向量 <7，9，10>。

隐式类型语言提供了更强的功能和灵活性，但这样做需要相当大的开销。其一，编译程序对所处理的隐式类型变量不容易生成高效率的机器代码。其二，当使用隐式类型变量时，失去了某种程度的安全性和清晰性。例如，

y:＝6;

IF x＞0 THEN y:＝'ABC';

z:＝y+3;

这个程序是错误的：若 x＞0 则企图用整数 3 加字符串'ABC'。在显式类型语言中，编译程序会检查出这个错误。而在隐式类型语言中，只有在执行期间才能发现这个错误。

6.1.5 程序语言的控制结构

结构程序语言通常包含四种控制结构：顺序执行、分支、循环和过程。程序流（语句执行的逻辑顺序）是靠这四种控制结构来决定的。

结构程序语言中一般不提供 GOTO 语句，即使程序语言中提供了 GOTO 语句，也建议少使用或尽量不使用 GOTO 语句。这是因为：GOTO 语句使程序的静态结构与动态执行情况差异甚大，从而使程序难以阅读和理解，而且容易出错，也难以查错。

1. 顺序执行结构

顺序执行结构是指按语句排列顺序一条接一条执行的一组语句序列。它是一种最常用而又最基本的控制结构。

例如，

S_1

S_2

⋮

S_n

或 S_1; S_2; …; S_n

或 BEGIN S_1; S_2; …S_n END

后者称为复合语句，可把它当作单个语句来处理。其中，S 表示语句或命令，下标用于区分各个语句或命令。

2. 分支结构

分支结构亦称为选择结构或条件执行结构。

(1)双向分支结构

条件语句用于在两个控制分支中选择一个。有下列两种条件语句:

①若条件 C 为真，则执行语句序列 S；否则，跳过 S。

例如，

```
IF C THEN S
```

或

```
IF C
   S
ENDIF
```

②若条件 C 为真，则执行语句序列 S_1；否则，执行语句序列 S_2。

例如，

```
IF C  THEN  S1  ELSE  S2
```

或

```
IF C
   S1
ELSE
   S2
ENDIF
```

(2)多向分支结构

分情况语句用于在多个控制分支中选择一个。有下列三种分情况语句:

①依次查看条件 C_1，C_2，…，C_n，若条件 C_i 为真，则只执行的语句序列 S_i；否则，执行语句序列 S_*。

例如，

```
DO CASE
   CASE C1
      S1
```

```
    CASE C2
        S2
        ⋮
    CASE Cn
        Sn
    OTHERWISE
        S*
ENDCASE
```

它的语义等价于下列嵌套的条件语句：

```
IF C1
   S1
ELSE
    IF C2
      S2
    ELSE
    ⋮
                IF Cn
                   Sn
                ELSE
                   S*
                ENDIF
    ⋮
    ENDIF
ENDIF
```

(2)依次查看表达式 E_1，E_2，…，E_n，若表达式 $E_0=E_i$，则只执行语句序列 S_i；否则执行语句序列 S_*。

例如，

CASE E_0 OF

E_1 : S_1;

E_2 : S_2;

⋮

E_n : S_n;

OTHERWISE S_*

END

它的语义等价于下列条件语句:

BIGIN

IF $E_0 = E_1$ THEN S_1

ELSE IF $E_0 = E_1$ THEN S_2

⋮

ELSE IF $E_0 = E_n$ THEN S_n

ELSE S_*

END

③若表达式 E 的值落到 1 到 n 的范围内，则只执行语句序列 S_E；否则，执行语句序列 S_*。

例如,

CASE E OF S_1; S_2; …; S_n; OTHERWISE S_* END

它的语义等于价于下列第二种分情况语句:

CASE E OF

1 : S_1;

2 : S_2;

⋮

n : S_n;

OTHERWISE S_*

END

3. 循环结构

循环结构亦称为重复结构或迭代结构。有下列三种循环结构:

(1)把终止检查放在循环体前

例如,

DO WHILE C

S

ENDDO

或 WHILE C DO S

它的语义等价于下列条件语句:

L: IF C THEN BEGIN S; GOTO L END

(2)把终止检查放在循环体后

例如,

REPEAT S UNTIL B

它的语义等价于下列第一种循环语句:

S; WHILE ~C DO S

其中,~表示逻辑非。

由此可见,第一种循环语句把终止检查放在循环体前,因此,若条件第一次计算为假,则根本不执行循环体。第二种循环语句把终止检查放在循环体后,这样可保证循环体至少被执行一次。

(3)根据循环变量的不同值重复执行循环体

例如,

FOR $i:=E_1$ STEP E_2 UNTIL E_3 DO S

它的语义等价于下列第一种循环语句:

$i:=E_1$

WHILE $i \leqslant E_3$ DO

BEGIN S; $i:=i+E_2$ END

它的语义也等于下列条件语句:

$i := E_1$;

L: IF $i \leqslant E_3$ THEN S; $i := i+E_2$; GOTO L END

最常见的循环变量的值是一递增的步长为1的整数序列，当然也可以是递减的步长不为1的实数序列，甚至是离散型的单值元素序列，例如，

FOR $i := E_1, E_2, \cdots, E_n$ DO S

这种循环语句适用于控制变量的值变化不规律的场合。

4. 过程

过程用来命名一段相对独立的程序。不同的程序语言在不同场合下对过程有不同的称呼，例如，子程序、程式、函数等，并赋予不同的内容。

在程序设计中引入过程是很有用的，这是因为:

(1)把在若干个地方重复出现的程序集中到一个过程中，可简化程序的书写工作，从而节省一定的工作量，同时也可节省程序的存储空间。

(2)容易把大而复杂的问题分解为在功能上相对独立的若干个小而简单的问题，因而增加了程序的清晰性和易读性。

直观上说，过程调用就是对过程体中的程序进行适当的修改后用来替换过程调用语句。所谓适当的修改，就是用调用语句中的实在参数替换过程体中的形式参数，我们称这个替换过程为参数传递。

程序语言的参数传递规则大致包括以下几种：值调用、引用调用、值结果调用、结果调用、名字调用。下面，以下列程序为例来说明各种参数传递规则。

```
VAR i: integer;
    A: ARRAY [1 · · 3] OF integer;
PROCEDURE callby (f, g: integer);
  BEGIN
    g := g+1;
```

```
    f:=5*i;
  END;
BEGIN
  FOR  i:=1 TO 3 DO A[i]=i;
  i:=2;
  callby (A[i], i);
  print (i, A[1], A[2], A[3])
END
```

1.值调用 (call by value)

当调用过程时，把实在参数的值传递给形式参数。每当过程中引用这个形式参数时，就把它当作局部变量来使用。因此，该过程无法改变实在参数的值。

上述程序值调用的结果是:

i=2

A[1]=1

A[2]=2

A[3]=3

2.引用调用 (call by reference)

当调用过程时，把实在参数的地址传递给形式参数。每当过程中引用这个形式参数时，就通过这个地址去引用实在参数的值，即通过形式参数间接引用实在参数的值。显然，过程中对形式参数值的改变就是对实在参数值的改变。引用调用亦称为地址调用。

上述程序的引用调用结果是:

i=3

A[1]=1

A[2]=15

A[3]=3

3.结果调用 (call by result)

当调用过程时，把实在参数的地址传递给形式参数。每当过程中引用形式参数时，把它当作局部变量来使用。当退出过程时，根据这个地址把形式参数的值传给实在参数。

上述程序结果调用的结果是：由于进入过程时，形式参数没有任何值，因此执行 g:＝g+1 时程序出错；g 无定义。为此，应在过程开头处给 g 赋初值。

(4)值结果调用 (call by value／result)

当调用过程时，把实在参数的值和地址均传递给形式参数。每当过程中引用这个形式参数时，就把它当作局部变量来使用。当退出过程时，根据这个地址把形式参数的值传给实在参数。值结果调用可看作是值调用和结果调用的结合。

上述程序值结果调用的结果是:

i＝3

A[1]＝1

A[2]＝10

A[3]＝3

(5)名字调用 (call by name)

当调用过程时，用实在参数对过程中任一出现的形式参数均进行原文替换。每当过程中引用这个形式参数时，对在形式参数位置上所替换的实在参数表达式重新计算。

上述程序名字调用使过程体中的程序变为下列形式:

```
BEGIN
    i: ＝i+1;
    A[i]: ＝5 * i;
END
```

其结果为:

i＝3

A[1]＝1

A[2]＝2

A[3]=15

有返回值的过程称为函数，函数可以在表达式中调用。程序语言一般都提供一些标准函数，而且还提供用户自定义函数的功能。

有些程序语言还提供了递归过程。递归过程就是自己能调用自己（直接地或间接地）的过程。例如，过程 A 调用过程 B，而过程 B 又调用 A。递归过程为解决递归问题提供了有力的工具。然而，当递归问题的嵌套层次超过程序语言允许的递归嵌套层次时，或当程序语言未提供递归过程时，只好把递归程序转换为等价的非递归程序，用分而治之的迭代算法代替递归算法。

6.2 数据库概要

6.2.1 什么是数据库

顾名思义，数据库是存放数据的“仓库”。直观上说，计算机上使用的“仓库”就是磁盘（硬盘或软盘）、磁鼓、磁带或其它外存储媒介。

“数据库”一词的英文写法为 database 或 data base. base 是基地的意思，故 database 意指供给数据的基地，因此国内也有人把它译为“数据基”。

然而，给数据库下一个确切的定义是很困难的。这是因为：首先，数据库是近三十年来迅速发展起来的计算机软件的一门新兴学科，它目前还处在从实践向理论过渡的阶段，它的概念、原理和方法还在继续发展变化，人们对它的认识也有一个历史的发展过程；其次，数据库是一个相当复杂的系统，涉及面很广，很难用几句话严格、简明、准确地概括它的全部特征。鉴于上述原因，现有的数据定义众说不一。尽管如此，我们还是在下面列出几本书中有关数据库的定义，供大家参考。

《辞海》中定义：按不同的应用领域，分门别类地收集了若干按一定格式事先编好的数据，并把它们存于外存储器中，就构成了“数据库”，供用户共同引用。除具有数据的检索和存储功能以外，还具有数据的修改、增删和整理等功能。

《LEARNING dBASE Ⅲ PLUS》中定义：数据库是相关信息或数据的有组织的集合。我们每天都能碰到几个数据库，例如，通信录、电话簿、备忘录等。

《英汉计算机辞典》中定义：数据库是在计算机存储设备上合理存放的相互关联的数据的集合。这些数据集合具有如下特点：

(1) 尽可能不重复（即最小冗余）。

(2) 以最优的方式服务于一个或多个应用程序（应用程序对数据资源的共享）。

(3) 数据的存入尽可能地独立于使用它的应用程序（数据独立性）。

(4) 用一个软件统一管理这些数据，例如维护、增加、变更和检索这些数据。

6.2.2 数据模型

数据模型是对客观事物及其联系的数据描述，是数据库设计的核心问题。在观念世界中，我们用实体描述客观事物，而每一实体都具有若干属性。例如，实体“人”具有姓名、性别、年龄等属性。在数据模型中，把描述实体的数据称为记录，而把描述属性的数据称为数据项。数据模型不仅反映记录内部数据项之间的联系，而且反映记录之间的联系。记录有类型与值之分，记录类型是记录的框架，记录值是记录的内容，因而记录之间的联系包括记录类型之间的联系和记录值之间的联系。

下面，我们简要介绍常用的三种数据模型：层次模型、网状模型、关系模型。

1. 层次模型

数据的层次模型是以记录类型为结点的有向树或森林。它满足下列两个条件:

(1)有且仅有一个结点无父结点，这个结点就是树的根;

(2)其它结点有且仅有一个父结点。

例如，图 6.6 表示一层次模型。

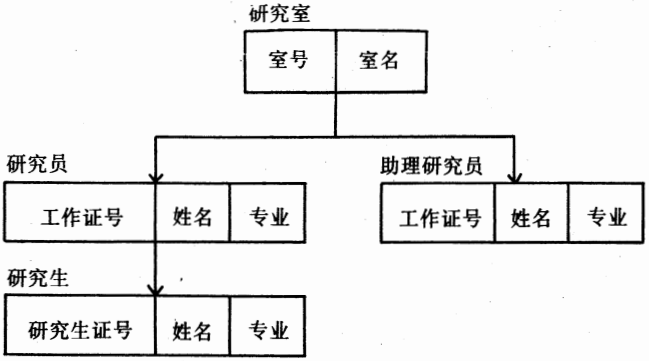


图 6.6　层次模型

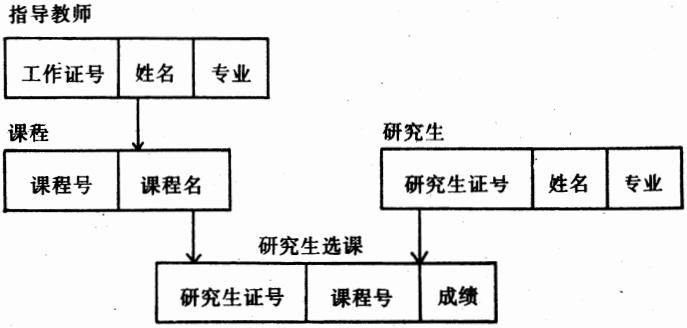


图 6.7　网状模型

2. 网状模型

数据的网状模型是以记录类型为结点的网状结构。它满足下列两个条件:

(1)可以有一个以上的结点无父结点;

(2)至少有一个结点有多于一个的父结点。

例如，图 6.7 表示–网状模型。

3. 关系模型

数据的关系模型把数据组成如图 6.8 所示的二维表形式。

小说人物表

姓名	性别	书名
孙悟空	男	西游记
林黛玉	女	红楼梦
诸葛亮	男	三国演义

图 6.8　二维表

二维表中的每行相当于关系模型中的一个记录，每列相当于各记录中同类型属性的数据项，亦称为字段或域。

关系模型是建立在集合代数理论基础上的。下面，我们用集合代数来定义二维表关系。

首先，定义字段。假设一组字段为 $D_1, D_2, \cdots, D_n$，每一字段定义为同类型属性值的集合（定义域)。

例如，小说人物表中的字段定义为:

D_1 = 姓名 = {孙悟空, 林黛玉, 诸葛亮}

D_2 = 性别 = {男, 女}

D_3 = 书名 = {西游记, 红楼梦, 三国演义}

其次，定义 $D_1, D_2, \cdots, D_n$ 的笛卡儿积为:

$$D_1 \times D_2 \times D_n = \{ < d_1, d_2, \cdots, d_n > | d_i \in D_i, i = 1,2,\cdots,n \}$$

其中，每一元素 $< d_1, d_2, \cdots, d_n >$ 叫做一个有序 n 元组。

例如，小说人物表中各字段的笛卡儿积定义为:

$D_1 \times D_2 \times D_3$ = 姓名 × 性别 × 书名

= {＜孙悟空, 男, 西游记＞, ＜孙悟空, 男, 红楼梦＞,
＜孙悟空, 男, 三国演义＞, ＜孙悟空, 女, 西游记＞,
＜孙悟空, 女, 红楼梦＞, ＜孙悟空, 女, 三国演义＞,
＜林黛玉, 男, 西游记＞, ＜林黛玉, 男, 红楼梦＞,
＜林黛玉, 男, 三国演义＞, ＜林黛玉, 女, 西游记＞,
＜林黛玉, 女, 红楼梦＞, ＜林黛玉, 女, 三国演义＞,
＜诸葛亮, 男, 西游记＞, ＜诸葛亮, 男, 红楼梦＞,
＜诸葛亮, 男, 三国演义＞, ＜诸葛亮, 女, 西游记＞,
＜诸葛亮, 女, 红楼梦＞, ＜诸葛亮, 女, 三国演义＞}

这里 $D_1 \times D_2 \times D_3$ 共有 18 个有序三元组。

最后，把在 D_1, D_2, …, D_n 上的关系定义为笛卡儿积 $D_1 \times D_2 \times \cdots D_n$ 的子集，用 $R(D_1, D_2, \cdots, D_n)$表示，其中 R 表示关系的名字。

例如，小说人物表中的关系定义为:

$R(D_1, D_2, D_3)$ = 小说人物表(姓名, 性别, 书名)

= {＜孙悟空, 男, 西游记＞, ＜林黛玉, 女, 红楼梦＞, ＜诸葛亮, 男, 三国演义＞}

由此可见，关系是字段的笛卡尔积的子集。一般说来，也只有取某一子集才有一定意义。在上例中，只有三个有序三元组构成了小说人物表这个关系。

总之，关系模型把数据之间的关系看成是二维表关系，而这种二维表关系又是建立在集合代数的关系理论基础上的，因此，建立在关系模型基础上的数据库称为关系数据库。每张二维表相当于集合代数中的一个关系，相当于关系数据库中的一个数据库文件。二维表中的每行相当于集合代数中的一个有序 n 元组，相当于关系数据库中的一个记录。二维表中的每列相当于集合代数中同类型属性值的定义域，相当于关系数据库的一个字段。

6.2.3 数据库管理系统

数据库管理系统简称为 DBMS（Data Base Management System），是操作和管理数据库的软件。一般说来，它包括以下功能:

(1)定义数据库

包括全局逻辑数据结构定义，局部逻辑数据数据结构定义，存储结构定义，保密定义以及信息格式定义等;

(2)管理数据库

包括对整个数据库系统运行的控制，数据存取、增删、修改、检索等操作的管理，数据完整性和安全性控制，并发控制等;

(3)建立和维护数据库

包括数据库的建立，数据库更新，数据库再组织，数据库结构维护，数据库恢复以及性能监视等;

(4)数据通信

具备与操作系统的联机处理，分时系统及远程作业输入的相应接口。

DBMS 通常由三部分组成:

(1)数据描述语言及其翻译程序;

(2)数据操作（或查询）语言及其编译（或解释）程序;

(3)数据管理子程序。

6.2.4 数据库语言

数据库语言有的是在现有语言的基础上改造扩充而成，有的设计成一种独立的语言。数据库语言可分为两大部分：数据描述语言（DDL: Data Description Language）和数据操作语言(DML: Data Manipulation Language)。

数据描述语言用于说明数据库管理系统所使用的数据结构，

对数据库或数据库的一部分给出逻辑数据描述。描述的对象是初等项、组项、记录和域以及数据库的特征和数据之间的关系。DDL 有的是在常用的高级语言（例如，COBOL）的基础上修改或扩充而成，有的设计成一种独立的语言。

数据操作语言用于对数据库中的数据进行各种操作，例如，存取、增删、修改、检索等。DML 分为两类：一类是宿主式数据语言。它对数据库的数据进行操作的语句是嵌入其它高级语言（例如，COBOL，FORTRAN，PL／1）或汇编语言之中的。被嵌入的语言称为该数据操作语言的宿主语言。另一类是自容式数据语言，亦称为数据查询语言。它是可以独立使用的数据操作语言，通常由一组命令组成（查询语言的功能往往不限于查询）。

6.2.5 数据库系统

数据库系统不是指数据库本身，也不是指数据库管理系统，而是指计算机系统中引进数据库后的系统构成。

数据库系统一般由数据库、数据库管理系统和数据库管理人员构成。

安装数据库系统后的计算机软硬件层次结构由里到外依次是：硬件、操作系统、数据库管理系统、应用程序。

6.2.6 微型机上常用的数据库管理系统

微型机上常用的数据管理系统有：dBASE，Clipper，Foxbase，Quicksilver，Informax，SQL，Datastore，Paradox，Oracle，R:base 等。

dBASE 是美国 Ashton-Tate 公司开发的关系数据库管理系统，是近年来微型机用户使用最广泛的软件，有人把它称为“大众数据库”。它在中国推广使用经历了 dBASEⅡ，dBASEⅢ，dBASEⅢ PLUS，dBASEⅣ四个历程。

dBASE Ⅲ在dBASEⅡ基础上增加了日期型和备注型两种数据类型，提高了数值精度，扩大了数据库容量和字段个数，增加了同时打开数据库及其它文件的个数，增加了内存变量个数，并把内存变量分为全局变量和局部变量；除对原命令增加许多新功能外，还增加了二十多条新命令和十几个新函数，改善了报表功能和屏幕输出格式，增加了 HELP 和 ASSIST 功能，增加了过程文件等。

dBASE Ⅲ PLUS 在 dBASEⅢ基础上，提供了更为友好的用户界面和新的数据目录处理方法，增强了调试功能和外部接口能力。特别是，dBASE Ⅲ PLUS 提供的网络版本在单用户版本的基础上增加了网络功能，使多个用户可在局部网环境中共享 dBASE。此外，原 dBASEⅢ程序可不加修改地在 dBASEⅢ PLUS 上运行，程序无需转换。

dBASE Ⅳ是 dBASE 的最新版本。它在 dBASE Ⅲ PLUS 基础上增加了浮点类型，对命令和函数做了数百处改进和扩充。它具有全新的用户界面，控制中心为用户提供了友好的非过程性界面。它具有结构化查询语言 SQL 功能和表格式查询 QBE 功能。它具有一个功能很强的应用程序生成器和一个内含的伪编译程序。网络功能也有所改进。dBASE Ⅳ Developer's Edition 是一个应用程序开发系统，它包括一个样本语言 Template 和一个伪编译程序 RunTime，为 dBASE 应用程序开发者提供了一个完整的程序设计环境。

6.3 编译程序、解释程序和伪编译程序

解释程序、编译程序和伪编译程序是三种不同形式的翻译程序。

就 BASIC 语言而言，一般有解释程序和编译程序两种版本，True BASIC 的解释程序和编译程序并存。

就 dBASE 系统而言，Ashton−Tate 公司的 dBASEⅡ、dBASEⅢ、dBASEⅢ PLUS 是解释程序，Nantucket 公司的 Clipper 和 Wordtech Systems 公司的 Quicksilver 是编译程序，Ashton−Tate 公司的 RUNTIME+、dBASE Ⅳ和 Fox Software 公司的 Foxbase 是伪编译程序。

1. 编译程序

编译程序把用高级语言写的源程序翻译成用汇编语言或机器语言表示的目标程序，再去执行目标程序。如图 6.9 所示。

根据逻辑功能的不同，编译阶段大致可分为以下五个步骤：词法分析、语法分析、语义分析、代码优化，代码生成。编译程序的结构也是由这五个逻辑部分组成的。这五个部分的主要逻辑功能是:

(1)词法分析

把程序中的字符拼成单词 (例如，标识符、常数、运算符、关键词、分隔符等)。

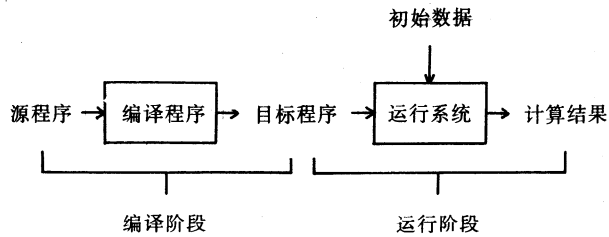


图 6.9　编译系统示意图

(2)语法分析

根据程序语言的语法定义，分析程序中单词之间的关系。

(3)语义分析

根据程序语言的语义描述，分析程序的意义。

(4)代码优化

为了提高目标程序的质量，在程序上实施一些变换。

(5)代码生成

生成目标代码。

编译程序和翻译外文具有相仿的翻译步骤，对照表如图6.10所示。

	编译程序	翻译外文
分析	词法分析	阅读原文，识别单词
	语法分析	分析句子的语法
	语义分析	分析句子的含义
综合	代码优化	修饰加工
	代码生成	写出译文

图6.10　编译程序和翻译外文

编译程序按扫描次数可分为一遍或多遍。对程序从头到尾扫描一次并做有关加工称为一遍。每一遍完成上述一部分或几部分工作。第一遍的加工对象是源程序。每一遍产生一个中间结果，称为中间语言程序。前一遍结果（中间语言程序）是后一遍的加工对象。最后一遍的结果是目标程序。

目前，编译程序大多采用分离编译方式，先把每个源程序模块分别翻译成目标程序模块，再通过连接程序把它们连接起来。

2. 解释程序

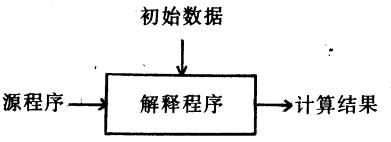


图6.11　解释系统示意图

解释程序不产生目标程序，而是直接解释执行源程序本身，按源程序中语句的动态次序逐句进行解释，并立即执行之。如图6.11所示。

解释程序对源程序进行逐行分析，检查错误，然后解释执行之。如果发现了错误，便可立即显示出错信息，并可现场校正错误。显然，这种交互式会话式的翻译对于程序调试和排除错误是有好处的。但是，如果一条语句的执行次数超过一次时（例如，在循环中），那么该语句每执行一次，就要重新分析，重新查错，重新翻译，因此，解释程序的效率较低。

由于编译系统执行编译好的目标程序，甚至是经过优化的目标程序，因此较之解释系统有以下两个优点：

(1)执行速度快；

(2)使源程序加密。

鉴于上述原因，可吸取各自的优点，先用解释程序去调试应用程序，再用编译程序去编译应用程序，执行它的目标程序。

由于解释程序与编译程序的工作方式不同，往往存在着不可避免的差别，因此解释型和编译型程序语言的语法和语义也有所不同，使用时一定要注意两者的差别。

3. 伪编译程序

伪编译程序实质上仍是一种解释程序，但它模仿和吸收了编译程序的特点，通过删除源程序中的注释和多余的空格，甚至通过词法分析（拼单词），把源程序转换成内部形式（一种特殊的代码），从而压缩了程序长度，提高了执行速度，并使源程序加密。

伪编译程序一般要先通过编码程序把源程序翻译为内部形式；然后通过连接程序把分离的已译码的内部形式的程序模块连接在一起，形成一个完整的内部形式的应用程序；最后，通过解释程序解释执行这种内部形式的程序。

注意，伪编译程序产生的程序不是可执行代码，因此不能独

立地运行，必须在解释程序的环境中由解释程序解释执行。

6.4 程序、应用程序与应用程序生成器

1. 程序

程序是指令的集合。程序设计不只是针对计算机的。从某种意义上说，任何按规定次序排列的一系列指令都称为程序。例如，烹调法实际上是一个程序。当使用烹调时，遵照规定步骤一步一步去烹调，每一步骤就是一条指令(比如，放 1／2 小勺的盐)。

计算机必须按准确的次序执行程序所包含的每一条指令。正确而有效地组织和编写这些指令，就是程序设计的任务。计算机程序通常是用程序语言编写的。程序语言中常把指令称为语句或命令。

2. 应用程序

一个程序执行一个基本任务或者执行全部任务，即整个程序设计项目。后者常称为一个应用程序（或应用软件)。可把一个应用程序分为若干个模块，每个模块执行一个基本任务，通常用一个主程序模块控制其它模块。因此，一个程序涉及整个程序设计项目或只涉及它的一个模块部分，一个应用程序则是包括一系列相关模块的全部程序。以 BASIC 为例，BASIC 语言是面向 BASIC 程序设计人员的界面，而 BASIC 应用程序则是面向最终用户的界面。也就是说，面向最终用户的界面不是 BASIC 程序语言本身，而且用 BASIC 程序语言编写的应用程序（或应用软件)。

3. 应用程序生成器

应用程序生成器（AG: Applications Generator）是应用程序的自动生成系统，是应用程序设计的辅助工具。由于应用程序生成器使程序设计实现了部分自动化，因此为应用程序开发人员

创造了良好的程序开发环境，提供了有力的程序开发工具，提高了程序设计效率和质量。

应用程序生成器常常通过菜单驱动方式帮助产生应用程序。对于初学者，能使您更快地入门，可通过学习应用程序生成器产生的程序来学习程序设计的基本方法。对于有一定程序设计经验的人，应用程序生成器可以帮助缩短程序开发时间，帮助开发较复杂的应用程序。

dBASE 应用程序生成器是目前微型机上流行的应用程序生成器，例如，Ahston-Tate 公司的 dBASE Ⅳ的 APGEN 和 Template，Software Botting 公司的 Flashcode，Fox&Giller 公司的 Quickcode 均是 dBASE 应用程序生成器。

dBASE 应用程序生成器用来自动生成 dBASE 应用程序，是一种扩充的数据库管理系统。安装具有应用程序生成器的数据库系统后的计算机软硬件层次结构由里到外依次是：硬件，操作系统，数据库管理系统，应用程序生成器，应用程序。

dBASE Ⅳ应用程序生成器 APGEN 用于生成以菜单为导引的应用程序。用户可通过菜单驱动方式和会话方式，按照自己的需求，描述和定义数据结构和程序结构，建立数据库文件、屏幕格式文件、报表文件和标签文件等，建立条形菜单、下拉菜单、文件表、结构表和值表等，并赋予相应的动作，包括数据库的更新、查询等操作，从而产生一个由菜单及其处理程序组成的应用程序。dBASE Ⅳ的样本语言 Template 为应用程序开发者按照自己的方式设计定做的应用程序提供了必要的资源。它为用应用程序生成器、屏幕格式、报表和标签来定做以菜单为导引的应用程序提供了新的设计工具，并提供了用这些新的工具自动编制应用程序的能力。

广义地说，除上述 dBASE 应用程序生成器外，dBASE 内部所含的屏幕格式生成器、报表生成器、标签生成器也是应用程序生成器。更准确地说，这些生成器属于应用程序生成系统的范

畴。

6.5 中文程序语言与中文数据库管理系统

中文程序语言与西文程序语言、中文数据库管理系统与西文数据库管理系统的主要区别在于：中文程序语言和中文数据库管理系统具有汉字处理功能，除此之外，它们应保持西文程序语言和西文数据库管理系统的原有全部功能。

为了保持西文程序语言和西文数据库管理系统的功能，达到中西文兼容，要保证汉字内码与西文字符编码在信息加工处理上的一致性，并把汉字纳入程序语言和数据库管理系统原有的字符类型或字符串类型。鉴于上述原则，在中文程序语言和中文数据管理系统中，凡是使用字符串的场合原则上都可以使用汉字串。严格地说，能够使用西文字母的场合一般都能使用汉字。但由于一个汉字占用两个西文字符的位置，汉字的处理与西文字符的处理有所不同。

本章首先从几个方面来概括和总结中文程序语言与中文数据库管理系统的特点，然后再分别讨论中文 BASIC 语言、中文 FORTRAN 语言、中文 PASCAL 语言、中文 dBASE 数据库管理系统的汉字处理功能。

6.5.1 中文程序语言与中文数据库管理系统的特点

本节将从汉字字符集、汉字名字、汉字数据、汉字的运算、汉字的比较、汉字的排序、汉字的查找、汉字的输入、汉字的输出、汉字文件处理、与汉字有关的函数、汉字注释等方面来讨论中文程序语言与中文数据库管理系统的特点。由于数据库管理系统的主要成分是数据库语言，为了叙述方例，在下面的讨论中把中文程序语言与中文数据库管理系统统称为中文程序语言。

1. 汉字字符集

中文程序语言的字符集是原西文程序语言规定的西文字符集与中文程序语言规定的汉字字符集的并集。

西文程序语言规定的西文字符集中包括哪些西文字符，取决于西文系统采用了哪种西文字符集，例如，ASCII 字符集，EBCDIC 字符集等；也取决于西文程序语言对西文系统所采用的西文字符集做了哪些限制，例如，西文 FORTRAN 语言规定西文字符集由四十七个基本字符组成，一般不允许使用这四十七个字符以外的字符。

同样，中文程序语言规定的汉字字符集中包括哪些汉字及其它图形字符，取决于中文系统采用了哪种汉字字符集，例如，中国大陆的汉字编码字符集（基本集和辅助集）、台湾的各种汉字字符集、用于特殊需要的扩展字符集等；还取决于中文程序语言对中文系统所采用的汉字字符集做了哪些限制。

程序语言对字符的加工处理，实质上是对它们的码值的加工处理。西文字符的码值取决于所采用的西文字符集及其代码体系，例如，ASCII 码，EBCDIC 码等。汉字的码值取决于所采用的汉字字符集及其内码形式，例如，GB2312 带标识位的二字节汉字内码、带标识码的三字节汉字内码、带引导码的汉字内码，台湾的 BIG-5 码、5550 码、通用码、TCA 码等。由此可见，一个西文字符含一个码值，而一个汉字却往往含有两个或两个以上的码值。

在中文程序语言中，应尽量避免汉字的码值与西文字符的码值发生冲突。例如，有的西文程序语言规定西文字符集为 ASCII 扩充字符集（码值为十进制 0～255），而中文程序语言规定的汉字字符集占用了码值 128～255 的位置，为了避免汉字的码值与西文字符的码值发生冲突，中文程序语言的字符集应是汉字字符集与 ASCII 字符集（码值为十进制 0～127）的并集。大多数情况下，字符是指汉字和可打印的 ASCII 字符（字母、数字、特殊字符和空格，码值为十进制 32～126）。在少数情况

下，字符是指汉字和 ASCII 字符。

在显示或打印时，一个汉字一般占两个西文字符的位置（汉字既不放大也不缩小）。因此，在计算字符个数时，一个汉字长度为 2。

2. 汉字名字

有的中文程序语言允许在变量名、过程名、文件名中使用汉字，一般把名字定义为以汉字或英文字母开头的汉字、英文字母、数字组成的字符串。有的中文程序语言则不允许在名字中使用汉字。

即使中文程序语言允许在名字中使用汉字，为了提高程序执行速度，也不提倡把汉字用作变量名、文件名等。

3. 汉字数据

中文程序语言利用西文程序语言中原有的字符类型或字符串类型来定义汉字数据。在字符型或字符串型数据（例如，字符型或字符串型常量、字符型或字符串型变量、字符型或字符串型数组、字符型或字符串型记录等）中，可把汉字与英文字母等同看待，只不过在计算字符个数时，一个汉字视作两个西文字符。这样，我们就可以利用字符型或字符串型表达式以及与字符型或字符串型数据有关的函数、语句和命令对汉字进行操作。

4. 汉字的运算

在中文程序语言中，凡是通过字符串运算符和字符型或字符串型函数对字符型或字符串型数据施加的运算，原则上均适用于汉字。例如，下列字符串的运算均允许字符串中含有汉字:

(1)字符串的连接（并置或合并）;

(2)从一字符串中截取一子字符串;

(3)在一字符串中插入另一字符串;

(4)从一字符串中删除一子字符串;

(5)用一字符串替换另一字符串中的一子字符串;

(6)把字符串赋给（传送给）字符型或字符串型变量、数组

元素（下标变量）或记录分量（字段或域）。

由于一个汉字占用两个西文字符的位置，因此，在汉字的运算过程中，当截取、插入、删除或替换字符串时，要格外小心，防止把一个汉字分为两截，否则会出现意想不到的结果。

5. 汉字的比较

字符串的比较实质上是比较字符的码值大小。在中文程序语言中，字符是指汉字和西文字符。字符的码值取决于所采用的汉字字符集及其内码形式和西文字符集及其代码体系。

例如，如果汉字采用 GB2312 高位为 1 的二字节汉字内码(码值为十六进制 80—FF)，西文字符采用 ASCII 码（码值为十六进制 00—7F)，那么，汉字的码值比西文字符的码值大，因此，西文字符排在汉字的前面。西文字符是按 ASCII 码值由小到大的顺序排列的。汉字是按 GB2312 规定的顺序排列的。GB2312 中汉字及其它图形字符的码值由小到大顺序是：其它图形字符、一级汉字、二级汉字（见 2.2.2 节)。

由于汉字串的比较是以其内码大小为依据的，因此，除等于和不等于比较有意义外，其它比较（小于、大于、小于等于、大于等于）没有多大意义。

6. 汉字的排序

排序亦称分类，是指按照一定的规则，根据每个项目中所包含的关键字，把各个项目分类或整理成有序序列。例如，数据文件中的记录通常是按输入的先后物理顺序存放的，而有时我们希望按某种有意义的顺序来重新组织数据文件，按某种逻辑顺序对记录重新排序。排序可使很多操作简化，例如，对已排序的数据查找时，可采用折半查找法加快查找速度。而且，已排序的数据中常常隐含着一些重要信息，例如，当学生考试成绩排序后，就确定了学生成绩名次及录取分数线等。

字符串的排序是指根据关键字中的字符串来排序的。最常用的字符串排序方法是按字符序排列顺序。所谓字符序，就是按字

符在字符集中的次序排序，也就是按字符的码值大小排序。由小到大为升序，由大到小为降序。在中文程序语言中，字符串是由汉字和西文字符组成的。如果汉字采用 GB2312 高位为 1 的二字节汉字内码，西文字符采用 ASCII 码，那么，由于 ASCII 码值小于汉字码值，ASCII 字符排在汉字前面，而汉字及其它图形字符又是按 GB2312 中规定的次序依次排列：其它图形字符、一级汉字、二级汉字（见 2.2.2 节）。因为一级汉字是常用汉字，而且是按汉语拼音字母顺序排列的，所以容易使人误以为汉字都是按汉语拼音字母顺序排列的。

由于汉字字符集中汉字往往未完全按一种排序方法来排列（例如，GB2312 一级汉字按汉语拼音字母顺序排列，二级汉字按部首顺序排列（见 2.2.2 节）），而且汉字内码形式也没有统一的标准，因此汉字交换码和汉字内码均不宜作序值用。鉴于上述原因，有的中文程序语言增加了汉字排序功能，或提供了汉字排序标准程序模块，用户可根据汉字的字音、字形或字义等属性对汉字排序，例如，拼音序、部首序、笔画序等。用户亦可根据自己的需要编写自己的汉字排序模块，按自定义的属性值排列顺序对汉字排序，例如，省市地名、单位、姓名等。这些属性值往往是数据文件中记录的某些属性，对这些属性值定义排序可以和属性值的数字化技术结合起来，即建立和使用汉字序值表。

例如，按我国省市自治区名排序，排列顺序是：先直辖市后省或自治区；从地理上先北后南再先西后东；地理位置以省会或自治区首府表示对应的省或自治区。图 6.12 给出它的汉字序值表。

汉字序值表由索引指针表和汉字属性值表组成。索引指针表用于表示属性值的数字代码及自定义的序值。不同的排序方法有不同的汉字序值表。利用汉字序值表可完成属性值数字码（序值）与相应汉字属性值之间的相互转换。在数据文件的关键字段中往往不存贮属性值，而存贮它们所对应的序值，因此记录不必

按属性值（汉字串本身）排序，而是按属性值的数字码（序值）排序。当显示和打印属性值时，再到汉字序值表中根据序值找到汉字串。

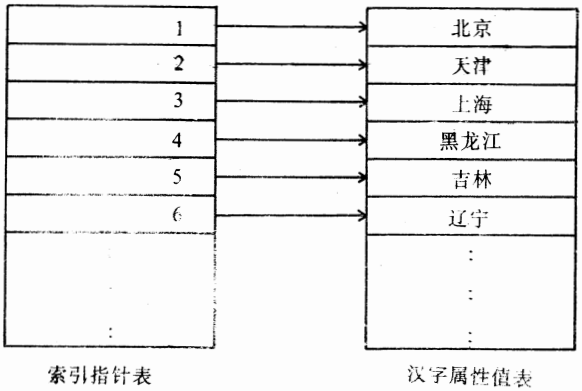


图 6.12　第一种形式的汉字序值表

汉字序值表的一种简单的表示形式是利用汉字属性值表的序号（相对地址）作为索引指针表。于是，图 6.12 中汉字序值表可简化为图 6.13 的形式。

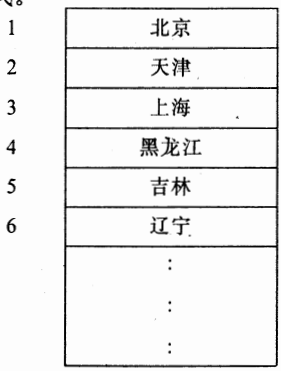


图 6.13　第二种形式的汉字序值表

汉字序值表的装载方法有以下几种:

(1)在程序语言中，可用数组或记录当作汉字序值表；在数据库管理系统中亦可用数据库文件当作汉字序值表。

(2)用外部数据文件装载汉字序值表

在确定一种汉字排序方法后，把装载这种汉字序值表的外部数据文件打开；以后每当查找汉字序值表时，直接在内存中使用该文件；当排序结束后，将该文件关闭。

(3)把汉字序值表作为库函数来调用

在确定一种汉字排序方法后，将这种汉字序值表以库函数的形式调入内存，进行查找。

7. 汉字的查找

在下述两种字符串的查找（检索）场合下均可在字符串中使用汉字:

(1)在一个字符串中或一个文本文件中查找另一个字符串，并确定位置。

(2)在数据文件中的大量记录中，按关键字中的字符串查找出某个或某些特定的记录。常用的查找方法有：顺序查找法、折半查找法、散列查找法。折半查找法的查找效率高，使用也方便，因此是一种行之有效的查找方法。但它要求事先要对数据文件中的记录按关键字排序。汉字的排序根据需要可采用字符序(把汉字内码当序值、程序语言提供的标准顺序、用户自定义的顺序)。通过索引文件查找数据文件中的记录，是程序语言中常见的查找方法。这种查找方法实质上就是折半查找法。

在日常生活中常用到索引技术，例如，按照图书索引目录去查找图书等。索引文件就是按索引技术建立起来的文件形式。建立索引文件就是对数据文件中的记录排序，但是，这种排序方法不是重新排列数据文件中记录的物理顺序，而是将数据文件中记录的逻辑顺序保存在索引文件中。也就是说，索引文件用于保存经排序的索引关键值及与之对应的数据文件的记录指针，而不包

含实际的数据记录。当要查找记录时，首先用折半查找法从索引文件中查找与指定值相匹配的索引关键值，然后由它所对应的记录指针找到要查找的记录。

8. 汉字的输入

对于中文程序语言，无论是在程序中交互式输入数据（在程序运行过程中，用户根据程序中设置的提示或询问，从键盘上输入信息），还是从数据文件成批输入数据，均可利用输入语句输入汉字。它依赖于中文系统所提供的汉字输入功能，例如，汉字输入方法。

9. 汉字的输出

中文程序语言可利用输出语句把汉字保存到磁盘中，或在屏幕上显示汉字，或在打印机上打印汉字。

在打印和显示汉字时，要充分考虑汉字的特点和中国人的习惯。例如，通过有关函数（比如，BASIC 中的 CHR$()，dBASE 中的 CHR()）来利用 ESC 序列，或通过有关语句或命令(比如，dBASE 中的 RUN)来调用中文打印公用程序，进行汉字多字体打印，并可打印出带有表格线的多字体汉字报表来。

10. 汉字文件处理

汉字文件包括汉字文本文件和汉字数据文件。在需要进行汉字信息处理时，程序中总离不开对汉字文件的处理。

文本文件是字符的有序集合。汉字文本文件与西文文本文件的主要区别是：汉字文本文件中含有汉字，而西文文本文件中只含有西文字符。

数据文件是按规定形式组织起来的数据的有序集合。数据文件是由一组记录组成，每个记录又可分成若干个字段。汉字数据文件与西文数据文件的主要区别是：汉字数据文件中的字符串允许含有汉字，而西文数据文件中的字符串只含有西文字符。

汉字文件的处理功能随中文程序语言而异。对于汉字文本文件，至少有建立、修改、输入、输出等操作。对于汉字数据文

件，至少有建立、修改、打开、关闭、输入、输出等操作。

11. 与汉字有关的函数和过程

中文程序语言中与字符串有关的标准函数和标准过程一般均适用于汉字。例如，字符串的运算、求字符串的长度、字符与码值的转换等函数或过程。

为了使用的需要，用户可自行定义一些有关汉字处理的函数和过程，这会给程序设计带来很多方便。

12. 汉字注释

对于中文程序语言，无论是功能性注释，还是状态性注释，在注释中均可使用汉字。注释一般是由注释语句给出的。在程序中加汉字注释，对于中国人阅读和理解程序是有利的。

6.5.2 中文 BASIC 语言

中文 BASIC 语言在西文 BASIC 语言的字符集和字符串型数据（字符串常量、字符串简单变量和字符串数组）基础上扩充了汉字，在注释语句、赋值语句、输入输出语句、读语句和赋初值语句、报表和数据文件等使用字符串的场合均允许使用汉字，并在字符串比较、字符串运算、字符串函数中均可处理汉字。下面，例举 IBM PC 机上中文 BASIC 语言的一些汉字处理功能。

1. 字符集

中文 BASIC 字符集不仅包括英文字母、数字和特殊字符，而且包括汉字。

2. 字符串型数据

字符串型数据是由汉字和西文字符组成的字符串。

3. 注释语句

REM 后面的注释内容可以含有汉字。例如，

```
100 REM 这是一个含有汉字的注释语句。
```

4. 赋值语句

可以把含有汉字的字符串型表达式的值赋给字符串变量。例

如，

```
10 A$ ="汉字"
20 B$ =A$
```

5. 输入输出语句

INPUT 语句、LINE INPUT 语句和 INPUT $ (n)函数均可把含有汉字的字符串输入给字符串变量。PRINT 语句和 LPRINT 语句可分别用于在屏幕上显示和在打印机打印含有汉字的字符串。例如，

```
10 INPUT "您叫什么名字?", NAME $
20 PRINT "我叫", NAME $
```

由于输入语句中的提示信息允许使用汉字，使得程序执行过程中人机会话界面的友好程度大为改善。

6. 读语句与赋初值语句

READ 语句从 DATA 语句中读取的字符串常量允许含有汉字。例如，

```
10 READ NAME $ , ADDRESS $ , TEL
20 DATA "白冰","大连市北京街 60 号", 31608
```

7. 字符串数组

字符串数组中允许含有汉字。例如，建立一个 500 人的通信录的程序如下：

```
10  REM 建立通信录
20  DIM MAIL $ (499, 2)
30  PRINT "请依次键入姓名、地址和电话号码: "
40  FOR I=0 TO 499
50  INPUT MAIL $ (I,0), MAIL $ (I,1), MAIL $ (I, 2)
60  NEXT  I
70  END
```

8. 字符串比较

中文 BASIC 允许含有汉字的字符串之间相互比较。例如，

在通信录中查找某人地址的程序如下：

```
110  REM 从通信录中查找某人的地址
120  INPUT "姓名"; NAME $
130  FOR I = 0 TO 499
140  IF NAME $ = MAIL $ (I, 0)  THEN 180
150  NEXT  I
160  PRINT "未找到!"
170  GOTO 120
180  PRINT : PRINT "= = = = 通信录 = = = = "
190  PRINT NAME $ , MAIL $ (I,1), MAIL $ (I,2)
200  END
```

9. 字符串连接

字符串连接可用于把多个汉字串合并为一个汉字串。例如，

```
10  A $ = "中国"
20  B $ = "科学院"
30  PRINT  A $ + B $ + "计算技术研究所"
```

10. 字符串函数

BASIC 中有许多字符串函数可用于处理汉字串。假设 X $ 和 Y $ 均为汉字串，则下列函数的意义如下：

LEN(X $)　汉字串 X $ 的字符串长度为它所含汉字个数的 2 倍；

LEFT $ (X $, n)　取汉字串 X $ 中最左边的 n / 2 个汉字，n 必须是偶数；

RIGNT $ (X $, n)　取汉字串 X $ 中最右边的 n / 2 个汉字，n 必须是偶数；

MID $ (X $, n, m)　取汉字串 X $ 中从第(n+1) / 2 个汉字开始的 m / 2 个汉字，n 必须是奇数，m 必须是偶数；

INSERT(n, X $, Y $)　从第(n+1) / 2 个汉字开始检索汉字串 Y $ 在汉字串 X $ 中首次出现的位置，若存在则返回其序

号，否则返回 0，n 必须是奇数。

例如，

```
10  X$ =″中国科学院″
20  PRINT LEN(X$)
```

它的打印结果是：10。

```
10  A$ =″中国科学院″
20  B$ =LEFT$(A$, 4)
30  PRINT B$
```

它的打印结果是：中国。

```
10  A$ =″中国科学院″
20  B$ =RIGHT$(A$, 6)
30  PRINT B$
```

它的打印结果是：科学院。

```
10  A$ =″中国科学院″
20  B$ =MID$(A$, 5, 4)
30  PRINT B$
```

它的打印结果是：科学。

```
10  A$ =″中国科学院″
20  B$ =″科学″
30  PRINT INSERT(A$, B$)
```

它的打印结果为：5。

字符串函数对含有汉字的字符串的处理类似于对汉字串的处理。

除此之外，其它字符串函数，例如，STRING$(n, m)，ASC(X$)，CHR$(n), VAL(X$)，STR$(X)等，对于汉字串来说，意义和作用就不明显了。

11. 打印报表

中文 BASIC 可通过 CHR$( )函数来利用 ESC 序列，打印出具有多种字形的汉字报表（见 1.3.3 节 1.）。

12. 数据文件的处理

中文 BASIC 的随机文件和顺序文件两种数据文件中允许含有汉字。汉字数据文件的处理与西文数据文件基本相同。此外，数据文件的文件名也可以使用汉字。

6.5.3 中文 FORTRAN 语言

中文 FORTRAN 语言在西文 FORTRAN 语言的字符集和字符型数据（文字型常数、字符型变量、字符型数组）的基础上扩充了汉字，允许把汉字赋值给字符型变量，允许输入输出汉字等等。下面，例举 IBM PC 机上中文 FORTRAN 语言的一些汉字处理功能。

1. 字符集

中文 FORTRAN 字符集不仅包括英文字母、数字和特殊字符，而且包括汉字。

2. 字符型数据

字符型数据是由汉字和西文字符组成的字符串。

3. 注释行

注释行中允许含有汉字。例如，

```
C   这是一个含有汉字的注释行。
```

4. 类型说明语句

类型说明语句可用于说明含有汉字的字符型变量或字符型数组。例如，

```
CHARCTER *2NAME*8, SEX, AGE, ADDRESS*90
```

说明字符型变量 NAME 的长度为 8，可容纳 4 个汉字；字符型变量 SEX 长度为 2，可容纳 1 个汉字：男或女；字符型变量 AGE 长度为 2，可容纳 2 个数字：00—99；字符型变量 ADDRESS 长度为 90，最多可容纳 45 个汉字。

又例如，

```
CHARACTER ADDRESS(100, 3)*30
```

说明 ADDRESS 是一个具有 1000×3 个元素的字符型二维数组，每个数组元素长度为 30，最多可容纳 15 个汉字。也就是说，字符型数组 ADDRESS 可容纳 1000 个地址，每个地址由三行组成，每行可容纳 15 个汉字。

5. 数据初值语句

数据初值语句给字符型变量所赋的文字型常数中允许含有汉字。例如，

```
DATA  NAME /"白冰"/, SEX/ "女"/, AGE/"28"/,
    ADDRESS/"大连市北京街 60 号"/
DATA  NAME, SEX, AGE, ADDRESS/"白冰","女","28",
    "大连市北京街 60 号"/
DATA (ADDRESS (I,1), I=1, TO 1000)/ 1000*"    "/,
        (ADDRESS (I,2), I=1, TO 1000)/ 1000*"      "/,
        (ADDRESS (I,3), I=1, TO 1000)/ 1000*"中国"/
```

6. 赋值语句

赋值语句赋给字符型变量的字符串中允许含有汉字。

```
CHARACTER  NAME*8
NAME="白冰"
```

7. 字符串比较

中文 FORTRAN 语言允许含有汉字的字符串之间相互比较。例如，在逻辑条件语句中，

```
IF  (COUNTRY.EQ."中国")  UNIT="科学院"
```

8. 输入输出语句与格式语句

在中文 FORTRAN 中，可用 READ/WRITE 语句和 FORMAT 语句配合输入输出汉字。例如，

```
CHARACTER  NAME(20)*8, ADDRESS(20)*30
DO 50  I=1, 20
WRITE(*, 10)
10  FORMAT(1X,"请输入姓名:")
```

```
      READ(*,20) NAME(I)
20    FORMAT(A8)
      WRITE(*,30)
30    FORMAT(1X,"请输入地址:")
      READ(*,40) ADDRESS(I)
40    FORMAT(A30)
50    CONTINUE
      END
```

由此可见，若要输入输出汉字变量，必须在 FORMAT 语句中用 A 型格式说明符来说明；若要输出汉字文字，则可将汉字作为文字型常数。

6.5.4 中文 PASCAL 语言

在标准 PASCAL 语言中，字符类型只能处理单个西文字符，字符型变量只能含有单个西文字符。在中文 PASCAL 语言中，由于汉字占两个西文字符的位置，因此，无法用字符型变量来存储汉字，也不能通过赋值语句进行汉字赋值，例如，

```
A:="字"
```

是不允许的。

然而，可在常量说明语句中把汉字串说明为常量，亦可在输入输出语句中用字符型变量来存储汉字。

此外，在有些 PASCAL 语言（例如，Turbo PASCAL）中扩充了字符串类型，从而可在字符串类型数据及其操作中使用汉字。

下面，例举 IBM PC 机上中文 PASCAL 语言的一些汉字处理功能。

1. 常量

可用常量说明语句来定义汉字常量，用于容纳含有汉字的字符串。例如，

```
CONST
    a="汉字常量"
```

2. 输入输出

可使用 PASCAL 的标准过程 READ／READLN 和 WRITE／WRITELN 来输入输出汉字。

在输入输出汉字时，由于标准 PASCAL 规定一个字符型变量只能容纳一个西文字符，因此一个字符型变量不能存储一个汉字，要用两个字符型变量存放一个汉字或和两个字符型变量一起输出一个汉字。例如，

```
VAR
    chr1, chr2: CHAR;
READ (chr1, chr2);
WRITE(chr1, chr2);
```

在输出汉字时，汉字常数和常量与西文字符常数和常量的输出完全一样。例如，

```
WRITE('请输入姓名：');
READ(name1, name2, name3, name4, name5, name6, name7, name8);
```

3. 字符串

在下列使用字符串的场合允许使用汉字：

(1)字符串变量的定义

例如，

```
VAR
    name: STRING[8];
    address: STRING[30];
```

(2)字符串的赋值

例如，

```
name:='白冰'
address:='大连市北京街 60 号'
```

(3)字符串的输入输出

例如，

```
REPEAT
    READLN(name);
    WRITELN(name);
UNTIL name=''
```

(4)字符串的比较

例如，

```
IF name='白冰' THEN
    BEGIN
      ⋮
    END
```

(5)字符串函数和过程

Turbo PASCAL 提供的几个可用于汉字串处理的字符串标准函数和标准过程如下：

CONCAT(S_1, S_2,…, S_N) 返回汉字串 S_1 到 S_n 连接在一起后的结果汉字串；

LENGTH(S_t) 返回汉字串 S_t 的长度，每个汉字长度为 2；

POS(Pat, S_t) 返回汉字串 Pat 在汉字串 S_t 中的位置；

COPY (S_t, Index, Size) 返回汉字串 S_t 中从第(Index+1)／2 个汉字开始的 Size／2 个汉字，Index 必须为奇数；

DELETE(S_t, Index, Size) 从汉字串 S_t中删除从第(Index+1)／2 个汉字开始的 Size／2 个汉字，Index 必须为奇数；

INSERT(S_{t1}, S_{t2}, Index) 在汉字串S_{t1}的第Index／2个汉字之后插入汉字串 S_{t2}，Index 必须为偶数。

例如，

```
CONCAT('中国','科学院')
```

的结果值为：'中国科学院'。

LENGTH('中国科学院')

的结果值为：10。

POS ('科学', '中国科学院')

的结果值为：5。

COPY ('中国科学院', 5, 4)

的结果值为：'科学'。

```
VAR  S='计算技术研究所'
DELETE (S, 5, 8)
```

S 的结果值为：'计算所'。

```
VAR  S='计算所'
INSERT(S, '技术研究', 4)
```

S 的结果值为：'计算技术研究所'。

字符串标准函数和标准过程对含有汉字的字符串的处理类似于对汉字串的处理。

(6)含有字符串分量的记录

例如，

```
TYPE
  perstype=
    RECORD
      name: STRING [8];
      address: STRING [30]
    END;
VAR person: perstype;
BEGIN
    WITH person DO BEGIN
      name:= '白冰';
      address:= '大连市北京街 60 号';
      WRITELN(name, address)
```

```
    END
END;
```

6.5.5 中文 dBASE 数据库管理系统

中文 dBASE 数据库管理系统在西文 dBASE 的字符集、名字和字符型数据基础上扩充了汉字，字符串的连接、比较、排序、查找、输入、显示和打印等操作均适用于汉字处理，并在全屏幕操作、提示信息、注解及其它注释、文件处理等方面均允许含有汉字。下面，例举 IBM PC 机上中文 DOS 支持的中文 dBASE Ⅳ的一些汉字处理功能。

1. 字符集

在英文 dBASE 中，字符是指 IBM 扩充字符集中的字符(码值为十进制 0–255)。ASCII 字符是 IBM 扩充字符集的子集(码值为十进制 0–127)。在中文 dBASE 中，由于汉字占用了码值为 128–255 的位置，因此字符是指汉字和 ASCII 字符。

2. 名字

变量名（字段名或内存变量名）、过程名是以汉字或英文字母开头的汉字、英文字母、数字和下划线组成的字符串。

文件名（包括模块名）和扩展名中的字符包括汉字、英文字母、数字及特殊字符。

变量名最长 10 个字符，过程名和文件名最长 8 个字符，扩展名最长 3 个字符。在计算字符个数时，一个汉字长度为 2。

3. 字符型数据

字符型数据是由汉字和 ASCII 字符组成的字符串。

字符型常量是用单引号(′ ′)或双引号(″ ″)或方括号([ ])括起的由汉字和可打印的 ASCII 字符(码值为十进制 32–127)组成的字符串。例如，′您好!′，[name 是一个变量名。]等均为字符串。

字符型变量（字段和内存变量）用于存储汉字和可打印的 ASCII 字符。

备注型数据只用在数据库的备注字段，用于存放可变长度的大块字符串信息，其中的字符串是由汉字和可打印的 ASCII 字符组成的。

4. 字符串的连接

字符串的连接运算符+和–均适用于汉字串的连接以及汉字串与 ASCII 字符串之间的连接。例如，

```
· USE  Names
· ? name+address
柳絮影  北京市 2704 信箱
· ? name–address
柳絮影北京市 2704 信箱
```

5. 字符串的比较

用于字符串比较的关系运算＜，＞，＝，＜＞，＃，＜＝，＞＝，$ 均适用于汉字串的比较以及含有汉字的字符串之间的比较。例如，

张(Zhang)的码值大于柳(liu)的码值：

```
· ? "张弥臻"＞"柳絮影"
.T.
```

汉字的码值大于 ASCII 字符的码值：

```
. ? "Zhang"＞"张"
.F.
```

6. 与字符有关的函数

由于字符包括汉字和 ASCII 字符，因此与字符有关的函数均能处理汉字。但由于一个汉字相当于两个 ASCII 字符，因此在处理汉字时要格外小心。

下面，例举一些与字符有关的函数：

(1)求字符串的长度

```
. ?   LEN  ("北京是中国的首都。")
```

在计算字符的个数时，汉字的长度为2。

(2)取字符串的一部分

分别从字符串"北京 上海 天津"的中间、左端或右端取字符串:

```
. ? SUBSTR("北京 上海 天津", 6, 4)
上海
. ? LEFT(" 北京 上海 天津", 4)
北京
. ? RIGHT(" 北京 上海 天津", 4)
天津
```

由于一个汉字相当于两个ASCII字符，因此截取汉字时要格外小心，不要截取半个汉字，否则会产生不堪设想的后果。比如，下列函数是不允许的:

```
SUBSTR("北京 上海 天津", 6, 3)
LEFT("北京 上海 天津", 3)
RIGHT("北京 上海 天津", 3)
```

(3)检索子字符串的位置

从一个含有汉字的字符串中查找另一个含有汉字的字符串，并确定位置:

```
. ? AT ("上海", "北京 上海 天津")
        6
```

由于每个汉字占两个位置，因此子字符串"上海"起始于字符串"北京 上海 天津"的第6个位置。

(4)删除字符串首部或尾部的空格

要列出数据库Names中字段name和address的内容:

```
. USE Name
. ? name, address
柳絮影  北京市2704信箱
```

姓名和地址之间是用name字段的尾部空格隔开的。为了消去

name 字段尾部的空格：

. ? TRIM (naem), address

柳絮影 北京市 2704 信箱

(5)重复字符串

重复汉字字符集中的制表符用以画线和制表：

. ? REPLICATE ('—', 20)

---

(6)修改字符串

用一个含有汉字的字符串替换另一个含有汉字的字符串的一部分：

. ? STUFF("北京市海淀区中关村路 25 号", 13, 6, "华夏")

北京市海淀区华夏路 25 号

由于一个汉字占两个位置，因此在替换汉字时要格外小心，不要替换半个汉字，否则会产生意想不到的结果。比如，下列函数是不允许的：

STUFF ("北京市海淀区中关村路 25 号", 14, 6, "华夏")

STUFF ("北京市海淀区中关村路 25 号", 13, 5, "华夏")

(7)字符与码值的转换

ASC() 函数返回字符串最左边 ASCII 字符的 ASCII 码值或汉字的第一个码值（汉字"牌"的十进制码值为 197 和 198）：

. ? ASC ("牌")

197

由于一个汉字具有两个码值，因此可以用两个 CHR() 函数相连接表示一个汉字：

. ? CHR (197)+CHR(198)

牌

7. 分类与索引

在使用分类命令 SORT 和索引命令 INDEX 根据关键字段按字符序对数据库中记录排序时，由于字符型字段中的字符串是

由汉字和 ASCII 字符组成的，因此要按照汉字和 ASCII 码值的大小排序，ASCII 字符排在汉字前面。下面，给出三个汉字排序的例子，汉字是按 GB2312 规定的次序排列的。

例 1.按字段 item 和 date 的升序对数据库文件 Orders 排序:

```
. USE Orders
. SORT ON item, date TO Itemdate
  100% Sorted       7 Records sorted
. USE Itemdate
. LIST item, date
Record# item                    date
     1  PC / XT                 87.02.07
     2  dBASE Ⅲ PLUS            87.01.10
     3  dBASE Ⅲ PLUS            87.03.24
     4  联想式汉卡              87.01.25
     5  联想式汉卡              87.04.05
     6  联想式汉卡              87.05.11
     7  图文编辑系统            87.04.23
```

其中，item 字段为大排序，date 字段为小排序。

例 2.按字段 name 的升序和字段 amount 的降序对数据库文件 Orders 排序:

```
. USE Orders
. SORT ON name, amount / D TO Nameamt
  100% Sorted                   7 Records sorted
. USE Nameamt
. LIST name, amount
Record#  name                    amount
     1   韩莉梅                  80000.00
     2   柳絮影                  50000.00
```

3	柳絮影	26240.00
4	马丁	32800.00
5	欧阳向南	6560.00
6	欧阳向南	1500.00
7	张弥臻	500.00

其中，name 字段为大排序，amount 字段为小排序。

例 3.按由字符型字段连接日期型字段形成的关键表达式建立索引文件:

```
. SET CENTURY ON
. USE Orders
. INDEX ON item+STR (YEAR (date), 4)+STR(MONTH(date), 2);
+STR(DAY(date), 2) TO  Iymd
   100% indexed                    7 Records indexed
. LIST item, date
Record#   item              date
      3   PC / XT           1987.02.07
      1   dBASE Ⅲ PLUS      1987.01.10
      4   dBASE Ⅲ PLUS      1987.03.24
      2   联想式汉卡        1987.01.25
      5   联想式汉卡        1987.04.05
      7   联想式汉卡        1987.05.11
      6   图文编辑系统      1987.04.23
. SET CENTURY OFF
```

8. 查找记录

查找记录命令可用于查找数据库中字段与含有汉字的字符串相匹配的记录。

(1)查找满足条件的记录

例如，在数据库文件 Names 中连续查找 address 字段中含有“北京”或含有“上海”的所有记录，直到超出指定的范围:

. USE Names

. LOCATE FOR address＝″北京″ .OR. address＝″上海″

Record＝　　1

. CONTINUE

Record＝　　2

. CONTINUE

Record＝　　5

. CONTINUE

End of LOCATE scope

(2)在索引的数据库文件中查找记录

例 1，在按字段 address 索引的数据库文件 Names 中查找 address 字段与字符串“上海”匹配的第一个记录：

. USE Names INDEX Mailaddr

. FIND 上海

. ? address

上海市九州路 100 号

例 2，分别用 FIND 和 SEEK 命令在数据库文件 Names 中查找并显示与内存变量 mname 中的汉字串相匹配的第一个记录：

. mname ＝″张弥臻″

张弥臻

. USE Names INDEX Mailname

. SEEK mname

. DISPLAY name

Record#　name

　　3　张弥臻

. FIND &mname

. DISPLAY name

Record#　name

9. 输入

从键盘输入的信息，凡是可以输入字符串的地方原则上均可输入汉字。例如，向数据库文件的字符字段或备注字段输入汉字，使用@…SAY…GET 命令或者 ACCEPT，INPUT，WAIT 命令按提示或询问输入汉字。

10. 显示与打印

凡是能显示和打印字符串的地方原则上均可显示和打印汉字。例如，在菜单、窗口、提示信息中显示汉字，在报表和标签中显示或打印汉字。

汉字显示和打印应充分考虑汉字的特点，例如，dBASE 程序可通过 CHR() 函数使用 ESC 序列，亦可通过 RUN 命令调用中文打印公用程序，来进行汉字多字体打印，并可打印出符合中国人习惯的带表格线的汉字报表（见 1.3.3 节）。

11. 全屏幕操作

在全屏幕操作状态下，凡是能够输入、编辑和显示字符串的地方原则上均能输入、编辑和显示汉字。例如，当执行 APPEND，ASSIST，BROWSE，CHANGE，CREATE，EDIT，INSERT，MODIFY，READ，SET 等全屏幕操作命令，按屏幕出现的数据格式输入和编辑字符型数据时，允许字符串中含有汉字。

12. 提示信息

在用于显示字符串提示信息的命令中，均可使用汉字。例如，在下列命令的字符串提示信息中均允许含有汉字:

ACCEPT…TO…

INPUT…TO…

WAIT…

@…SAY…GET…VALID…ERROR…MESSAGE…

SET MESSAGE TO…

DEFINE POPUP…MESSAGE…

DEFINE BAR…MESSAGE…

DEFINE MENU…MESSAGE…

DEFINE PAD…MESSAGE…

13. 注解及其它注释

注解及其它注释中均允许含有汉字。例如，利用 NOTE，*，&&命令在程序中加的注解以及在控制结构的结尾行后面加的注释信息中均可使用汉字。

14. 文件处理

在 dBASE 文件中，凡是能处理字符串的地方均可处理汉字。例如，在数据库文件的字符型数据中允许含有汉字。在命令文件、过程文件等文本文件中可处理汉字。

第七章　中文通用应用软件

中文通用应用软件与西文通用应用软件的主要区别是：中文通用应用软件具有汉字处理功能。除此之外，它应保持西文通用应用软件的原有全部功能。

基于这种考虑，本章一方面介绍微型机上常用的几类通用应用软件：

(1) 字处理软件与编辑程序；
(2) 数据表软件；
(3) CAD／CAM／CAI 与图形软件；
(4) 组合软件与软件族；
(5) 统计软件；
(6) 工具软件。

另一方面，概括和总结中文通用应用软件的特点，讨论中文字处理软件和中文绘图软件这两种典型的中文通用应用软件的汉字处理功能。

7.1　字处理软件

字处理软件（Word Processor）亦称为文字处理软件、文字编辑软件、文书处理软件等。它是电脑的最基本最必要的通用应用软件，是办公自动化的必备工具。

字处理软件在文字修改、存储方面提供了打字机无法提供的功能。它的最大特点在于它在文字编辑方面提供了极大的灵活性，使用户可以随心所欲地修改文本中的文字，并得到最佳的编排格式。目前，用电脑代替传统的纸和笔来处理文件的书写、编

辑和存储，已成为办公自动化的必然趋势。

1. 字处理软件与编辑程序

字处理软件是具有文字输入、输出、编辑和存储等文字处理功能的软件。它常有两种编辑方式：一种是文书编辑方式，用于编辑文章、信件、公文、报告、笔记、表格等；另一种是非文书编辑方式，用于编辑程序文件和数据文件。由于编辑方式不同，因此处理文字的命令和功能也有所不同。

最好不要使用文书编辑方式来编辑程序文件。这是因为文书编辑方式所编辑的文本文件中往往含有格式码，例如，特定的页格式、连字符和黑体字等。这些格式码对于程序语言来说一般是不能识别的。例如，dBASE 无法执行用 WORDSTAR 的文书编辑方式编辑的程序，在翻译或运行过程中会出现莫名其妙的结果。

除单独的字处理软件外，很多软件常常带有自己的编辑程序。例如，DOS 中的行编辑程序 EDLIN，dBASE 中 MODIFY COMMAND 和备注字段的输入和编辑所使用的内部编辑程序，Turbo PASCAL 中提供的编辑程序等等。编辑程序有两个缺点：

(1) 由于编辑软件通常是某一软件的一部分，因此，一般说来，它的编辑功能较弱；

(2) 要使用编辑程序，必须首先进入包含它的软件。

2. 微型机上常用的字处理软件

目前，微型机上常用的字处理软件有：Wordstar，Personal Editor,Word,Word Perfect，Display Writer，PFS: Professional Write 和 File，Norton Editor，MultiMate,NewsROOM，Filing Assistant，Reporting Assistant，Writing Assistant 等等。

下面，简要介绍微型机上常用的几种字处理软件的特点。

(1) Wordstar

Wordstar 是 MicroPro 公司 1979 年推出的，目前版本已升级到 Version 5.5。

Wordstar 提供了多级菜单提示，使用户借助菜单可方便地使用它。

Wordstar 提供了两种编辑方式：一种是文书文件编辑方式，用于编辑文章、书信、公文、报告、笔记、表格等文书文件；另一种是非文书文件编辑方式，用于编辑程序文件和数据文件等非文书文件。

Wordstar 提供了全屏幕编辑功能，给出了用于设计屏幕编辑格式的一整套格式化命令。特别是提供了功能强大的字块操作，用户可标记字块，进行字块的复制、移动、删除等操作。Wordstar 还提供了查找字符串、替换字符串等功能。

除此之外，Wordstar 还允许运行 DOS 命令或用户程序、索引文书文件、构造目录、改变驱动器或目录、进行文件命名、复制、删除、保护、打印等操作。

Wordstar 是一种广泛应用的字处理软件。但是，它也有如下缺点：

①功能键定义复杂，而编辑时真正用到的功能键并不多；

②速度太慢，这是由于原始的 Wordstar 是针对小内存设计的，读盘过于频繁所致；

③光标移动不自由，不能体现全屏幕编辑的特色；

④对于中文处理，在西文 Wordstar 基础上汉化的中文 Wordstar 给人的感觉是：该用的功能缺乏，不该用的功能挺多，十分丰富的打印功能几乎用不上，而复杂的功能键令人眼花缭乱。

(2) PE

PE (Personal Editor) 是 IBM 公司的产品。PE2 是 PE 的新版本。

PE 是一个全屏幕编辑软件。它具有较强的编辑功能，在方便性和速度方面占有优势。它的主要用途是文书处理，也可用于编辑程序文件和数据文件。它的主要特点如下：

①操作方便。光标可在整个屏幕或整个被编辑的文件上随意移动，光标移至哪里，就可以在哪里进行编辑。用户可直观地从屏幕上看到文件的操作结果。它对绘制表格也提供了方便。

②可同时编辑 20 个文件。这些文件可以相互复制，移动，交叉进行修改，在各文件之间相互剪接和调用。

③屏幕上可以开窗口。允许在一个屏幕上最多同时出现四个窗口，在不同的窗口显示不同的文件或同一文件的不同部分，使用户可对同时编辑的几个文件相互参照、相互对比，为同时编辑多个文件提供了方便。

④提供了三种设定标记区的方法，即字标记、行标记、块标记，可对标记区进行移动、复制、清除、覆盖、打印、查找字符串、删除等操作，而且可对标记区进行堆栈操作，把原有的标记区保存起来，重新设定当前标记区，在需要的时候恢复原有标记区。

⑤可保存最近十次被删除的内容，因此可把误删的内容恢复出来。

⑥可以编辑大型文件。当被编辑的文件超出内存容量时，PE 会自动把溢出部分存到磁盘上，使编辑工作继续下去。

⑦所有命令都可直接在命令行上使用，也可以由宏定义功能引用。利用宏定义，用户可根据自己的需要定义功能键，随意修改和扩充编辑系统的功能，使它更加适合自己的要求。

⑧提供了调用 DOS 命令的手段。在编辑过程中，可随时转入 DOS 环境，执行 DOS 下可执行的任何命令和操作，并可用 EXIT 命令返回编辑现场。这种功能适合于程序调试，在修改源程后去执行，执行发现错误后再返回 PE 修改源程序。

(3) Word

Word 是 Microsoft 公司 1984 年推出的。它与 Wordstar 的功能相似，也提供了多级菜单提示，提供了文书文件和非文书文件两种编辑方式，提供了块操作、查找和替换字符串以及文件的

命名、删除、复制等功能。

Word 是具有 WYSIWYG 特色的字处理软件之一。WYSIWYG 是 What-You-See-Is-What-You-Get 的缩写，意思是说，你所看到的就是你所得到的（即见即得）。

Word 能在屏幕上显示出粗体字、斜体字等字形。由于这些特殊的字形都是在图形方式下处理的，因此原版本速度较慢。在 Word 4.0 版本中，这个缺点已不复存在。Word 4.0 比原版本增加了许多特色，例如，宏指令，用户可利用任何英文字母的组合、控制键或功能键的组合来设定所需要的宏指令。这个功能不但增加了指令的灵活性，也使输出文件的格式更符合用户的要求。此外，它还具有读数据表文件、在文字的周围画边框线等功能。

Word 的最新版本 Word 5.0 在 Word 4.0 基础上新增加的功能大致如下：输入 PIC、HPG、PCX 和 BIT 图形文件与文字结合，并作缩放处理，再利用 PRINT RREVIEW 可先在屏幕上预演印出的样子；文字可以围在框形图案的四周，图形可作为文字衬底之用，也能放置于文件的任何一个位置，线条和方框有多种花样，也可将图形位置命令用于表格上，并把表格加上背景；用户可以交替编辑多栏，并可自动编页码；书签（Book mark）可以将一长段文章随意移动、附接和参考（crossreference）；增加评注（annotation）功能，用户可将某处文字用上标数字标出，此时 Word 5.0 会跳到文件的最末页，并打印出所标的文字及上标数字，自动附上标示的日期与时间，留待用户加评注，要重返文件中标示注解处，只要选 JUMP ANNOTATION 输入评注名称即可。与 Windows split footnote 合用，可以一前一后，分别将评注和文字卷动，提供 MACRO 将不同文件内的评注集合成一篇文章；可支持 OS／2 和 EMS；书签、评注及其它的文件管理方式，在局部网环境下尤具威力。

(4) Word Perfect

Word Perfect 是 Word Perfect 公司推出的。它是 1987 年全

球销售量最多的字处理软件，销售量达65万多套。

Word Perfect是一全屏幕编辑软件。除了具有一般字处理软件的屏幕编辑功能（例如，插入、删除、修改、复制等）和文件处理功能（例如，文件的调入、编辑、存储、打印等）外，它还具有一些独特的功能，例如，用户可方便地选择一些制表符（比如，单划线、双划线或用户自定义的画线）来画边框线或制表，可同时编辑多个文件，可实现反向书写，可查看磁盘目录和文件内容，可进行文件的复制、换名、删除和打印等。

在Word Perfect 5.0版本中，用户可将图形轻易地插进文章的任何位置，并修改图形的大小、方向等。它不但可提供激光印刷机中的各种字形，甚至可支持彩色印刷。

(5) PFS: Write和File

Write和File是Software Publishing公司1983年推出的PFS系统中的两个字处理软件。

PFS: Write是文本文件编辑软件，可对文件进行屏幕页式格式化编辑，进行插入、删除、替换、移动、块复制、标题居中、加重和加下划线等操作，可进行页式格式化打印。值得一提的是，PFS: Write通过屏幕页式格式化编辑，使用户在屏幕上看到的与打印机上打印出的格式一致，这是WYSIWYG观念的萌芽期。

PFS: File是数据文件编辑软件。它使数据文件的编辑就象在一张绘图纸上作图一样，可随意增加、编辑、查找、拷贝记录，可格式化打印报表和标签。

PFS: Write和File强调易学性，用户在较短的时间内不但可学会基本功能，而且较深的功能操作也可很快掌握。

(6) DisplayWriter

DisplayWriter是IBM公司针对专业人员设计的一种字处理软件。DisplayWriter 4提供了一种名为文件夹（Paper Clip）的功能，使用户在检索出原先存储的文件后，光标会自动跳回用户

离开该文件时的位置。

3. 字处理软件的发展方向

字处理软件正朝着 WYSIWYG 的方向发展，这是字处理软件发展的必然趋势。

随着硬件功能越来越强，各种高质量的图形显示和打印设备(例如，激光印刷机）层出不穷；同时，随着软件技术在图形和窗口环境方面的发展，随着用户越来越高的要求，字处理软件逐步向着图文并茂的方向发展。未来的字处理软件很可能会与桌上排版系统合二为一。

针对特定行业设计专业性字处理软件，也是字处理软件的发展趋势之一。每个行业对字处理都有自己特定的要求，例如，科学和工程专业常常需要用到特殊的专业符号、方程式、化学式等，法律、保险、印刷等行业也都有各自的专业需求，这些是一般的通用型字处理软件所不能提供的。例如，Manuscript 是 Lotus 公司针对技术性文件所设计的工程与科学用字处理软件，问世后便受到相当的重视。

另外，字处理软件有与其它软件组合的趋势。一种组合方式是，字处理软件与其它软件（例如，数据库管理软件、数据表软件、图形软件等）组成一个软件族，相互配合使用，同时可进行文件格式转换。另一种组合方式是，形成含有字处理功能的多功能的组合软件，例如，Framework，Symphony 等。

7.2 数据表软件

数据表软件（Worksheet Software）亦称为表处理软件、电子表格软件、试算表软件、工作表软件等。它代替了原来使用笔、纸和计算器的传统计算统计方法。它象使用笔、纸和计算器那样，能够以自己的双眼看着工作的进程，但却不必象使用笔、纸和计算器那样将计算结果逐一记录下来。它是办公自动化、企

业管理、商业数据处理等领域数据分析的辅助工具，亦可用作决策分析的辅助工具。

1. 数据表

数据表亦称作工作表、试算表、电子表格。数据表常常呈现为二维表形式，有的数据表软件还具有三维表形式。

屏幕只是数据表的一个可见窗口，屏幕上显示的画面只是数据表的一部分，通过数据表的上下左右移动，可从窗口观察到数据表的各个部分。

数据表中的每一行和每一列的交叉点为一个数据单元（亦称工作单元）。每一数据单元可用行号和列号命名，亦可用标识符命名。每一列数据单元的宽度相同（当然，特别设置的例外），用字符个数表示数据单元的宽度。

命令一般是以菜单的形式出现在屏幕上部或下部。用户通过选择和执行命令来对数据表进行操作。

2. 数据表软件的主要功能

数据表软件主要有以下几项功能:

(1) 数据单元、行、列的输入、修改、插入、删除、拷贝、查找、排序、统计、锁定、外联和打印等功能。

(2) 窗口功能。把屏幕分成若干个窗口，可同时观察数据表的不同部分，便于相互对照。

(3) 数据表文件与其它软件数据文件（数据库文件、图形文件等）之间相互转换的功能。

(4) 有的数据表软件还带有程序语言，用于对某个键编写程序，用键盘宏命令来代替一串命令,按一个键便可完成一系列工作。

(5) 有的数据表软件具有三维表功能。单表文件(single-sheet files) 只含有一个数据表。对于少量的数据，单个数据表是足够的。但对于大量的形形色色的数据，通常最好把数据分为几个数据表，放入多表文件（Multiple-sheet files）中。例如，在几个商店想建立一个文件，使它包含每个商店的收益情

况以及综合数据。对于单表文件，必须键入所有的收益情况。并在同一数据表中综合不同区域中的数据，一次只能设置一个数据区域的格式，而且从一个区域移到另一个区域按键需要花费很多时间。然而，对于多表文件，可把各个商店的收益情况放于单个数据表中，而用另一个数据表来存放综合数据。此外，一次可以同时设置所有数据区域的格式，而且从一个区域移到另一个区域，只需使用单个键即可。可把一个数据表中的数据移动或拷贝到另一数据表中。

(6) 数据表软件网络版本提供网络功能，使多个用户共享数据表文件及其操作。

3. 微型机上常用的数据表软件

微型机上常用的数据表软件有：Lotus1−2−3，Multiplan，Supercalc，Vicicalc，QA，VF 等。

Lotus 1−2−3 是 Lotus 公司推出的以数据表为主体的组合软件。除数据表处理功能外，它还具有数据库管理和统计图形功能。Lotus 1−2−3 的命令采用菜单驱动，所有命令均按树状结构组织，呈现在屏幕顶部。数据的组织形式采用二维数据表的形式。Lotus 1−2−3 在数据处理方面提供了若干个常用的科学计算函数、统计函数、日期函数、逻辑运算等函数；在数据管理方面提供了数据排序、查询、连接等功能；在图形处理方面可生成条形图、圆饼图、折线图、叠积图、坐标图等常见统计图形；在程序自动化方面提供了键盘宏命令，还提供了 Lotus 1−2−3 数据表文件与其它软件数据文件之间的相互转换。Lotus 1−2−3 3.0 版本是 Lotus 1−2−3 的最新版本，它增加了三维表功能、EMS 功能和网络功能等。

7.3 CAD / CAM / CAI 与图形软件

1. CAD / CAM / CAI

CAD 是 Computer-Aided Design（计算机辅助设计）的缩写。它用计算机来帮助设计人员进行设计。例如，在计算机的设计过程中，使用 CAD 技术进行体系模拟、逻辑模拟、插件划分、自动布线等，能提高设计工作的自动化程度，节省人力和时间。

CAM 是 Computer-Aided Manufacturing（计算机辅助制造或计算机辅助生产）的缩写。它用计算机来进行生产设备的管理、控制和操作的过程。例如，在产品制造过程中，应用计算机来控制机器的运行，处理产品制造过程中所需要的数据，控制和处理材料的流动以及对产品进行测试和检验等。CAM 能提高产品质量、降低成本，缩短生产周期以及改善制造人员的工作条件。

CAM 与 CAD 有着密切的关系。CAD 的输出结果常常作为 CAM 的输入。CAD 偏重于设计过程，而 CAM 则偏重于产品的生产过程。

CAD／CAM 经近二十年的发展，今天已成为计算机的重要应用领域，同时也已成为机械、电子、建筑等生产行业的一项重要新技术。它的出现，使传统的人工设计和制造方法转变为自动或半自动的方式，使传统的生产技术产生一个重大变革。

微型机的出现使 CAD／CAM 的普及达到一个新境界。特别是，近年来随着以 Inetl 80386 为 CPU 的微型机的出现，在系统的速度和容量上有了一个大的提高，达到了以往小型机的水平，从而使得微型机上的 CAD／CAM 系统已能代替以往价格昂贵的小型机 CAD／CAM 系统。由于这类系统价格低，使用方便，能够取得较好的经济效益，以工作站网络为基础的中高档 CAD／CAM 系统和以微型机为基础的低档 CAD 系统已成为当今世界上 CAD／CAM 系统的两个发展方向。

CAI 是 Computer-Aided Instruction（计算机辅助教学）的缩写。它用计算机帮助教学，学生学习时和计算机处于对话的方

式，计算机能指出学生在学习过程中的错误，并按照学生的回答，来选择下一个课题或进入下一个学习阶段，使每个学生按其学习能力循序渐进。

近年来，产生了许多 CAD／CAM／CAI 软件，这些软件是适应 CAD／CAM／CAI 的发展而产生的,并对 CAD／CAM／CAI 的发展起着推动作用。随着 CAD／CAM／CAI 的发展，A 不再表示 Aided 或 Assisted（辅助）的含义，而是表示 Automatic（自动）的含义。

2. 图形软件

计算机制图是计算机辅助设计中的重要一环。例如，在机械设计中，绘制零件图和轴测图；在建筑设计中，绘制施工图、结构图和透视图；在地形测量方面，绘制地形图；在气象、海洋方面，绘制天气图、潮流潮汐图；在经济统计方面，绘制各种统计图表；在电子工业方面，绘制集成电路图；在军事方面，绘制军事地图和图表等等。

图形软件是计算机制图的重要工具。按用户界面，可把图形软件分为图形语言和图形通用应用软件两大类。按产生的图形的类型和用途，又可把图形软件分为一般图形软件和商用图形软件两大类。下面，简要介绍这四大类图形软件。

(1) 图形语言

图形语言供计算机辅助设计人员编写有关图形处理的应用程序，重点放在处理图形数据结构上。利用图形语言进行程序设计的基本过程是：首先，用数学方法建立产生图形的数学模型；然后，根据数学模型编写绘图程序并输入计算机，最后，计算机执行程序产生控制信息，控制绘图设备制出图形。

构造图形语言有两种途径：一种途径是在原有程序语言基础上加一组处理图形的标准过程，例如，图形 FORTRAN 语言就是在原 FORTRAN 语言基础扩充而成的；另一种途径是构造具有图形处理功能的新语言，例如，GIS，APL，MASP，

LEAP，AED 等。

(2) 图形通用应用软件

图形通用应用软件是一种人机对话式的图形处理软件。计算机辅助设计人员可利用光笔、鼠标器、键盘等在屏幕上直接绘制图形，就象在画板上画图画一样。通过反复修改和检查，直到设计人员满意为止。设计完毕后，计算机通过绘图设备绘制出设计图纸，通过打印机打印出有关技术资料。例如，AUTOCAD 就是一个图形通用应用软件。

(3) 一般图形软件

一般图形软件用于计算机辅助设计的各个领域。微型机上常用的 CAD 与图形软件有 AUTOCAD，CADKEY，CAD，VCAD，PRODESIGNⅡ,EESYSTEM，New ROOM,SHOW PARENTER 等。

AUTOCAD 是美国 Autodesk 公司推出的用在微型机上的计算机辅助绘图与设计软件包。它是一个功能很强的平面绘图辅助工具，根据用户发出的命令可迅速准确地绘出所需要的图形，具有容易校正绘图误差以及做较大修改而无需重新绘制全图的特点。

它适用于各种设计人员，例如，工艺美术工作者、机械师、建筑师、服装设计师等。它能完成下列绘图与设计功能：绘制各种艺术画面，绘制流程图和组织结构图，绘制电器、电子、化学、城建及机械零件图，绘制复杂的数学函数图形，制作精细艺术品等等。总之，任何用手画的图形都能借助 AUTOCAD 绘制得更快更好，而且还可以绘制手工画图所难以完成的图形，例如，集成电路印刷线路板的设计。

AUTOCAD 提供了一组用于构造图形的图素：点、线、弧、圆、框、文字等。首先，通过输入命令告诉 AUTOCAD 要绘制哪一图素。命令可从键盘键入，可用光笔等设备从屏幕菜单上选择，也可根据数字化图形输入板上的菜单或多个按钮的定标设备的按钮输入，还可通过定做命令文件使其自动执行。然后

回答显示屏幕上的提示，对所选图素提供参数，这些参数通常包括图素在图形中的位置坐标、大小、角度等数值。给出参数后，图素生成并显示在屏幕上。由此可见，AUTOCAD 绘制图形的过程就象在画板上图画一样，通过光笔、鼠标器等设备可直接在屏幕上绘制图形。

AUTOCAD 在屏幕上提供了各种操作命令的菜单供选择使用。图素可被擦除、移动、旋转或拷贝以生成重复的图形；可改变图形在显示器上的视图，例如，放大、缩小等；给出有关图形的线段长度、面积、数据等信息；提供绘图辅助手段，例如，图素的精确定位等；图形绘制好后以图形文件的形式存入磁盘，留作以后调用。当需要某个图形时，通过选择菜单自动调入图形文件，把图形显示在屏幕上或用绘图设备绘制在图纸上。这种会话式操作，既易学易用，又容易修改剪裁为用户定义的操作形式。

同时，AUTOCAD 还具有立体绘图、视角旋转、零件属性分类、统计以及 AutoLISP 程序设计语言等功能，可成为全三向(3D) 功能的绘图设计用软件。

AUTOCAD 的最新版本第 10 版在第 9 版基础上新增加的功能如下：用户自定义相对坐标 (user coordinates system)，动态观察 (dynamic view)，多重窗口 (multi-viewport)，所有绘图和编辑命令全部改为含 3D 功能，动态剪辑 (clipping)，扩展的 AutoLisp 等。

(4) 商用图形软件

商用图形软件是用于商业领域中的图形软件。它主要指统计图形软件。微型机上常用的商用图形软件有 MASTER GRAPHICS，dGRAPH，GRAPH Assistant，4PT，FAST GRAPH 等。下面，通过简要介绍 MASTER GRAPHICS 来说明商用图形软件的功能。

MASTER GRAPHICS 是 Ashton-Tate 公司推出的商用图形软件族。该软件族包括以下四个软件：

①Chart–Master

它用于把用户自行设计的数值数据或 Lotus 1–2–3 等通用数据表软件中的数据表和 dBASE 等数据库管理系统中的数据库的数值数据转换为统计图形。这些统计图形包括：簇式条形图、栈式条形图、散式图、折线图、圆饼图、区域图等。它的主要功能是：建立、编辑、检索、存储、删除图形，在多种分辨率的显示器上显示图形，在打印机上打印或在绘图机上绘制图形。

②Sign–Master

它用于建立高质量的美观的美术字、图表和表格。它提供了七种字体（如图 7.1 所示）、八种颜色的字符，并可加下划线，进行版面调整，变成斜体字等，从而可把普通字体输入的演示、协议、报表、广告及其它商用文件转换为美观的字体，在打印机或绘图机上输出。

Font Name	Sample
Standard	Standard
Bold	Bold
Roman	Roman
Bold Roman	Bold Roman
Script	Script
Swiss	Swiss
Bold Swiss	Bold Swiss
Symbol	★☼✂☤✆☎

图 7.1　Sign–Master 的字体

③Diagram-Master

它是用于绘制高质量商用图形的计算机绘图工具软件。它提供了类似绘图板的功能，用户可在绘图板上选择形状（方形、圆形、椭圆形、三角形），定位，放大和缩小，键入文字，绘制直线、弧和文字。它的绘图方法具有灵活性，使用方便，并可把各种图形作为图形库保存在磁盘中。

④Map-Master

它用于根据提供的统计数据绘制地图，例如，人口密度地图、销售状况地图、市场分布地图、利润统计地图等等。

3. 统计图形软件

统计图形软件是一类商用图形软件。统计图形亦称为统计图表，是数值数据的图形表示。统计图形软件用统计图形来表示数据表或数据库中的信息，更为直观，更为容易理解；而且，计算机绘图比手工绘图更为准确。

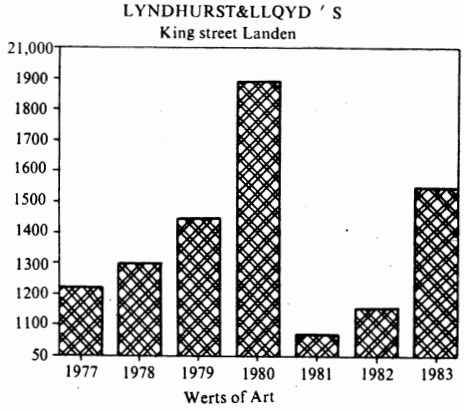
常用的统计图形或图表有以下几种:

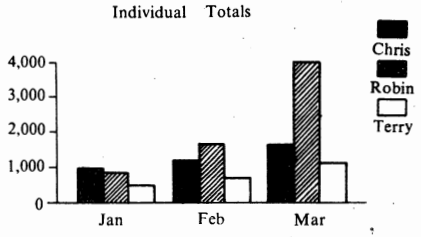
(1) 条形图 (Bar graphs / charts)

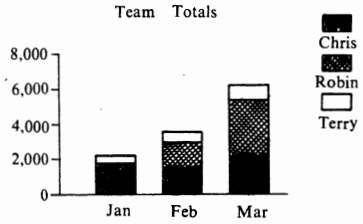
条形图又称直方图。它利用不同长度、颜色、图案的竖直或水平条形（长方形）来显示不同数值的大小，适于少量数值的比较。

对于具有多个数值的数值组的比较，可采用簇式条形图(Clustred bar graphs / charts) 或栈式条形图 (Stacked bar graphs / charts)。在一个簇式条形图中，有几个条形并列聚集(挨在) 一起。在一个栈式条形图中，一个条形堆放在另一个条形的顶上。由此，不仅每组数值之间可以相互比较，而且每组数值中的各个数值之间也可以相互比较。栈式条形图亦称叠积图。

图 7.2 依次给出简单条形图、簇式条形图和栈式条形图的例子。







Chris	$ 1,000.00	$ 1,200.00	$ 1,800.00
Robin	$ 900.00	$ 1,800.00	$ 4,000.00
Terry	$ 500.00	$ 799.00	$ 1,080.00

图 7.2 条形图

(2) 点线图(Line and points graphs / charts)

点线图用点或折线或点与线的配合来表示一组或几组值，适于大量数值的比较，也适于显示数值的进程，例如，价格的升降。对于多组数值的显示，可采用不同颜色或不同形状的多条折线或多组点来表示。

点线图包括以下几种:

①点线图 (Line and points graphs / charts)

点线图显示点和连接它们的直线。

②折线图 (Line graphs / charts)

折线图只显示线，数据点不被标记。当只想综观图象而对实际数值不感兴趣时，适于采用这种图形。

③点图 (points graphs / charts)

点图不仅可以象折线图一样进行多组数值的比较 (采用不同的标记)，也可表示零乱分布的数值，这些点一般是无法连线的。因此，有时把它称为散式图 (Scatter graphs / charts)。

④XY 图 (XY graphs / charts)

XY 图亦称坐标图。它适于表示两组数据的对应关系或函数关系。例如，分别用 X 轴和 Y 轴表示数据表的行和列，或表示数据库中的两个相关字段。

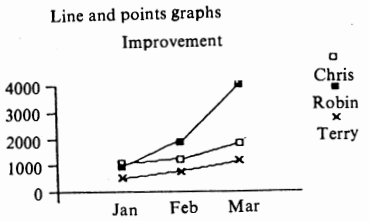
XY 图与一般点线图的主要区别在于: 一般点线图的 X 轴不具有数值意义，不表示数值的大小; 而 XY 图的 X 轴也是一个数轴。

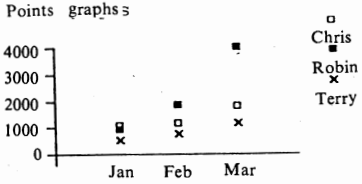
⑤区域图 (Area graphs / charts)

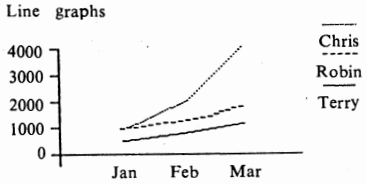
区域图亦称平面图 (Surface graph / charts)。它实际上是折线图的变种。它在折线和水平轴之间设置阴影。

图 7.3 分别给出点线图、点图、折线图、XY 图和区域图的例子。

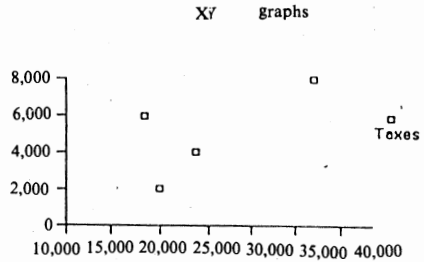
Line and points graphs







	Jan	Feb	Mar
Chris	$ 1,000.00	$ 1,200.00	$ 1,800.00
Robin	$ 900.00	$ 1,800.00	$ 4,000.00
Terry	$ 500.00	$ 799.00	$ 1,080.00



Income	20,000	24,000	19,000	37,000
Taxes	2,000	4,000	6,000	8,000

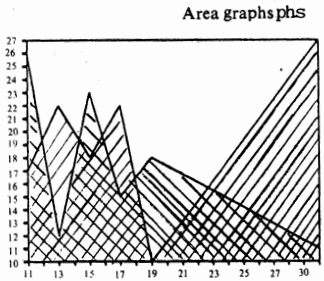
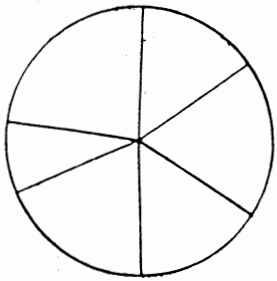


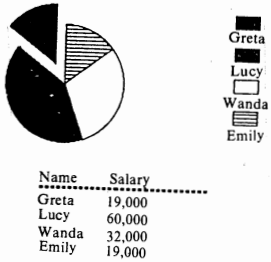
图 7.3 点线图

(3) 圆饼图(pie graphs / charts)

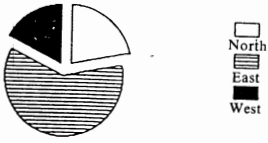
圆饼图亦称扇形图。它把一个圆分成若干个不同颜色或不同图案的扇形，用以表示不同数值之间的关系，适于比较一个整体之中各个部分之间的比例关系。为了强调某一个扇形，可把它与圆的剩余部分分离开。

图 7.4 给出三个圆饼图的例子。





Name	Salary
Greta	19,000
Lucy	60,000
Wanda	32,000
Emily	19,000



Region	Total Sales
North	90,000
East	225,000
West	65,000

图 7.4 圆饼图

(4) 高／低／收／开图 (High–low–close–open graphs／charts)

高／低／收／开图用于按时间追踪股票或商品行情，有时亦称为股票图。对于每个观察日期，该图形给出一条带有一个、两个、三个或四个值的垂线，这四个值分别是最高价、最低价、收盘价和开盘价。一般用　表示收盘价，　表示开盘价，垂线的上顶点表示最高价，下顶点表示最低价。

这种图不仅可用于观察和追踪股票市场的趋势，而且可用于观察和追踪具有一个、两个、三个或四个值的任何数值。

这种图也可看作是特殊的点线图。

图 7.5 分别给出高／低／收图和高／低／收／开图的例子。

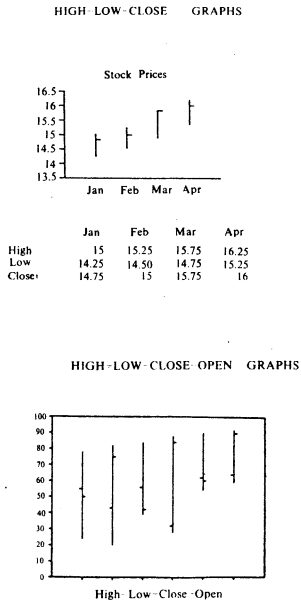


图 7.5　高／低／收／开图

(5) 混合图(Mixed graphs / charts)

混合图是上述几种图形的混合图形。例如，簇式条形图或栈式条形图与折线图的混合图。

7.4 组合软件与软件族

1. 组合软件

组合软件是具有多种功能的软件，例如，数据库管理功能、字处理功能、数据表功能、图形功能、通信功能等。它汇集多种功能于一个软件之中，将多种功能有机地结合在一起。组合软件亦称为套装软件、集成软件、整合软件等。

微型计算机上常用的组合软件有 Symphony，Framework，Knowledgeman，Lotus 1-2-3，ABLE-ONE，GEM Desktop 等。下面，简要介绍 Symphony 和 Framework 这两种组合软件。

(1) Symphony

Symphony（交响乐）是 Lotus 公司推出的组合软件，亦称为 Lotus 1-2-3-4-5，它在 Lotus 1-2-3 功能基础上作了大幅度提高。

Symphony 具有分析数据表、建立和编辑文本、产生商用图形、建立和管理数据库、与其它计算机通信等功能。

由于上述功能彼此不同，Symphony 为每项功能各自提供一种窗口，以形成不同的工作环境。这五种窗口及其对应的工作环境如下：

SHEET　建立和分析数据表；

DOC　字处理；

GRAPH　显示和保存商用图形；

FORM　管理数据库；

COMM　与其它计算机通信。

Symphony 在不同的窗口环境中提供了各自的操作命令，同时还提供了适用于所有环境的公用操作命令。

(2) Framework

Framework（框架）是 Ashton-Tate 公司推出的组合软件。FrameworkⅢ是 Framework 的最新版本。

Framework 具有字处理、数据表、数据库管理、图形、远程通信等功能。各功能简述如下：

字处理：具有较强的文字编辑功能，可通过窗口编辑和放大进行全屏幕编辑；

数据表：定做数据表，可与数据库和图形相互转换；

数据库管理：建立数据库，键入数据，自动计算字段值、显示记录，通过过滤器查询记录等，由于是 dBASE 厂家推出的，因此隐含 dBASE 的功能；

图形功能：能根据数据表或数据库中的数据绘制、显示和打印图形，能迭加和组合图形；

远程通信：能利用电话、调制解调器等硬件设备进行远程通信。

Framework 中含有一个功能很强的公式语言 FRED。它能够把 Framrwork 的命令以文件的形式存储起来，以便迅速方便地运行 Framework。

2. 软件族

组合软件功能全，有一定的优点。然而，由于组合软件是一种“无所不包”的大型通用软件，企图兼收并蓄各种功能，“凡是你想要的，我这里都有”，致使软件太大，使用太复杂，很难广泛推广使用。这犹如程序语言的发展，ALGOL 60 发展为“公共汽车”语言 ALGOL 68 和 PL／1，就很少有人问津，而发展为结构程序设计语言 PASCAL，则促进了程序语言的发展。同样，通用应用软件的发展，也应当朝着小而精的方向发展，关键在于用户界面友好，而不应朝着贪大求全的方向发展，否则会适得其

反。软件族是解决上述问题的一种尝试。

软件族是由若干个具有不同功能的离散的软件组成的。软件族中的每个软件都是一个独立的软件，可单独使用，亦可配合使用。例如，PFS 软件族包括 Write、File、Graph 等软件；IBM 助手软件族包括：Filing Assistant，Reporting Assistant，Writing Assistant，Graphing Assistant 等软件。

目前，计算机应用出现了多种软件配合使用的倾向。各种软件取长补短，各自发挥优势，相互配合解决一个问题。广义地说，相互配合使用的软件也构成了一个软件族。例如，用 FORTRAN 语言进行数值计算，而用 BASIC 语言作图。软件族中各软件之间的联系和沟通方式有两种：程序连接和数据连接。对于不同高级语言的程序连接有两种方法：一种是不同语言的目标模块的连接；另一种是在一种高级语言程序中调用另一种高级语言的目标模块（机器语言程序）。但是，很多高级语言文本未提供这两种功能，即使提供了这两种功能，限制也很多，而且使用很困难。因此，对于软件的配合使用，建议尽量不采用程序连接方式，最好采用数据连接方式。数据连接是以数据文件作为桥梁来实现的。如果两种软件的数据文件中的数据格式恰好相同，例如，BASIC 语言与 FORTRAN 语言的顺序文件中的数据格式恰好相同，则可直接把一种软件输出的数据文件作为另一种软件输入的数据文件。如果两种软件的数据文件中的数据格式不同，则需要先进行文件转换，把一种软件输出的数据文件转换为另一种软件的数据文件，才能作为另一种软件的输入文件。

3. 文件转换

文件转换一般是指文件之间的相互转换。文件转换程序是实现文件自动转换的专用程序，有时也称为文件转换软件、文件转换系统、文件转换工具等等。文件转换程序有的是单独的软件，例如，内码转换程序；有的包含在某一软件之中，例如，dBASE 提供了数据库文件与高级语言数据文件之间的转换、与

数据表软件（Lotus 1-2-3，Multiplan，Visicalc 等）数据表文件之间的转换，与字处理软件（Wordstar，PE 等）文本文件之间的转换、与 PFS 文件之间的转换等等。

然而，目前的文件转换程序，特别是微型机上的文件转换程序，大多仅仅是文件转换的辅助工具，文件转换尚不能完全自动进行，还往往限于文件中符号的转换，或文件内涵语言语法上的转换。

大家知道，程序语言包含语法、语义、语用三大要素。为了实现文件转换的自动化，首先必须实现文件内涵语言的形式化描述。多年来，形式语言理论的发展为语法形式化描述奠定了基础，可是语义形式化描述却相当困难。尽管近年来国际上出现了许多形式语义的描述方法，但现存的形式语义描述方法大多只停留在对语义的精确的、严格的、无歧义的描述上，却很少付诸于程序和数据的自动转换上。鉴于语义转换的困难性（且不说语用问题），目前的文件转换很难实现完全的自动化，所以，现存的文件转换程序大多限于文件中符号的转换，或文件内涵语言词法上的转换，而且往往限于相近语言之间的文件转换。因此，现存的文件转换程序大多只能自动地完成大量的转换工作，而另一部分转换工作仍需人工去完成。但它毕竟能帮助人们自动完成大量的转换工作，甚至在一定的条件下，有些文件几乎可以全部自动转换，例如，大多数数据文件的转换。我们应当充分利用这些文件转换程序。可以相信，随着语义形式化研究的不断深入，真正的文件自动转换工具必将会诞生。

7.5 统计软件

统计软件是用于社会科学或自然科学的统计分析工具。

本节简要介绍三个统计软件包 SPSS，STATISTICS 和 STATPAK，用以说明统计软件的功能。

1. SPSS

SPSS（Statistical Package for Social Sciences）是专供统计数据分析处理用的一套社会科学统计软件包，是社会科学（包括心理学、社会学、政治学、人文地理学、企业管理等）、企业界及政府机构的专业人员的统计分析工具。

SPSS 提供了从基本到高深的统计分析程序，可对各种数据进行转换和分析。它所产生的数据报表可根据需要加入各种标题，再配合上精致的统计图形。

早期的 SPSS 只能在大型机上使用，随着微型机的发展，已出现 IBM PC 及其兼容机上可以使用的版本 SPSS／PC+。

SPSS 通过以下四个阶段来产生 SPSS 的分析结果：

（1）收集数据并加以编码

通常为了了解社会或心理学上的现象，社会学家需要用一些信息或数据来加以解释。这些信息除了参考已经出版的刊物之外，还必须通过已有的研究推导出一些新数据，并将这些数据加以收集和整理，以便产生所需要的信息，或者得到能够解决特定问题的答案。

社会学家最关注的基本元素就是变量，它包括各种形式的信息项目，例如，性别、年龄、婚姻状况、收入、宗教信仰等等。

在社会科学中最普遍的研究方式便是实验。例如，我们可把一串单词的长度看成一个变量，并对这个变量进行某些处理，而产生另一个变量，比如，这个变量代表学习这串单词所需花费的时间。

社会科学的另一个研究方法是调查，向有关人员收集数据。调查的方式是交谈或问答。有关被调查人员情况（例如，性别、年龄等）的数据可以看成是一些变量，针对这些情况项目变量所提供的答案数据又可视为另一些变量。

交谈有两种形式：结构性的和非结构性的。问答是结构性

的。结构性交谈和问答是一种标准测试，是按照预先定好的问题进行的，主要好处是可以使研究人员对同一问题的答案进行比较分析。非结构性交谈的主要好处是可以得到较多关于回答者的数据，便于对比较感兴趣的问题做深入了解。

无论是交谈或问答，均有两种问题形式：一种是开放性问题，允许回答者以自由开放方式回答，例如，您对电脑发展状况的看法如何？回答者可用自己的想法来回答问题，因此可产生各式各样的答案。然而，这种问题不容易做分析，即比较不同回答者对相同问题所提供的答案是困难的。另一种是封闭性问题，只允许回答者在固定的答案范围内回答问题，例如，被调查人员的情况，对性别、年龄、婚姻状况、收入、宗教信仰等问题的回答。回答者必须在限定的答案范围内回答，因此进行比较答案的工作就容易多了。

在 SPSS 中所用到的数据必须以相同的方式编码。对每个回答者而言，必须确定每个变量值（例如，对一个问题所回答的答案）。以性别这个变量为例，它只有两个可能并有效的答案（男性或女性），因此在对数据进行编码时，便可用 1 代表男性，用 2 代表女性。对于年龄这个变量来说，它的答案就是正整数值，因此可取作它的编码。由此可见，封闭性问题的答案容易编码，而且所产生的编码数值适合于 SPSS 的分析。而开放性问题尽管可以设法找到一些合适的编码方式来对每个答案编码，但在实用上毕竟还是相当困难的。

编码系统把通过调查得到的许多回答者提供的原始数据（例如，被调查人员的性别、年龄等）转换成用数字串表示的编码值。把这些以编码形式表示的数据汇集起来，便形成了数据集。

(2) 输入并存储数据

当收集数据并编码后，便可把这些数据以数据集的形式输入电脑，这些数据集如同下列矩阵形式：

019865976597659

029867596576586

035475486547654

047865967654445

在这个例子中，有四行数据，每行分别代表一个回答者所提供的数据，每行含有 15 个数字，一个或几个数字分别代表回答者对某一特定问题回答的答案，而整个矩阵便构成了数据集。

数据集可通过编辑程序来建立，并可保存到磁盘上，以备后用。

(3) 对数据加注解和标记

在数据收集、编码并以矩阵形式输入电脑之后，还要解释或描述这些数据，以使电脑能在数据集内把变量识别出来。

首先，要把每个个体所提供的一串数字切分为几个部分。例如，在上述第一行中的一串数字可切分为下列 10 个数字串:

01，9，8，65，9，76，5，9，7，659

可见，这 15 个数字代表 10 个变量的编码值。按照这种格式，可把上述数据集的格式安排如下:

01	9	8	65	9	76	5	9	7	659
02	9	8	67	5	96	5	7	6	586
03	5	4	75	4	86	5	4	7	654
04	7	8	65	9	67	6	5	4	445
V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}

在描述数据格式时，必须给每个变量各赋予一个名称，例如，V_1—V_{10} 分别表示它是第几个问题的编码答案。变量的名称本身最好代表明显的意义，例如，以 sex 代表性别，以 age 代表年龄等等。

除变量名称外，还可对每个变量加标记，用来对分析结果做一番注解。这些分析结果是由系统对数据集进行分析后得到的显示输出或打印输出。输出数据由较完整的标记加以说明，可使我

们容易了解统计结果或表格所代表的意义。

此外，也可以对变量的值加以标记。例如，从性别变量 sex 来看，它的编码值 1 的标记为'male'，编码值 2 的标记为'famale'。其它常用的数值标记有'agree'，'disagree'；'yes'，'No'；'always'，'often'，'sometimes'，'never'等等。数值标记和变量标记一样，仅仅用于输出结果时加注解，增加输出数据的可读性。

格式描述、变量名称、变量标记和数值标记都是用来描述数据集的某些属性，它与数据集一起输入并存储起来，供分析数据用。

(4) 使用加注解的数据

SPSS 通过一些程序命令，例如，操作命令、数据定义和转换命令、过程命令，来提供统计方法，进行统计分析和检定，并产生摘要和表格。

例如，下面给出学生基本数据表及其相应的 SPSS 程序。

姓　名	性别	年龄	身高	体重	中文	英文	数学	班别
Alfered	1	14	69	112	78	82	78	1
Alice	2	13	56	84	65	79	82	1
Berndette	2	13	65	98	86	95	57	1
Barbara	2	14	62	102	95	98	69	2
Henry	1	14	63	102	92	91	78	2
James	1	12	57	83	83	86	83	2
Jane	2	12	59	84	69	87	89	3
Janet	1	15	62	112	78	83	85	3
Jaffrey	1	13	63	84	68	79	92	3
John	1	12	59	99	86	68	94	1
Joyce	2	11	51	50	78	92	89	2
Judy	2	14	64	90	84	78	79	3
Louise	2	12	56	77	79	82	74	1
Mary	2	15	66	112	84	86	78	2

Philip	1	16	72	150	78	84	86	3
Robert	1	12	64	128	83	85	88	1
Thomas	1	11	57	85	88	79	83	2
William	1	15	66	112	86	73	89	3
Alice	2	14	70	107	89	78	92	1
Philip	1	13	57	122	98	94	93	3
John	1	12	59	122	98	94	93	3
James	1	11	62	146	90	88	89	1
Henry	1	12	63	132	96	89	90	2
Thomas	1	15	56	85	89	94	96	3
Judy	2	13	66	90	93	90	98	1
Marry	2	14	68	96	78	89	92	2
Jane	2	12	69	100	88	90	87	3
Alice	2	15	63	88	87	94	92	1
Ross	2	14	70	85	92	85	87	2

set disk = on.

set listing 'b:sp_dat.lis'.

set printer off.

Data list / Name 1–10(A) Sex 12 Age 14–15 Height 17–18

Weight 20–22 China 24–25 Eng 27–28

Math 30–31 Class 33.

begin data.

Alfered	1	14	69	112	78	82	78	1
Alice	2	13	56	84	65	79	82	1
Berndette	2	13	65	98	86	95	57	1
Barbara	2	14	62	102	95	98	69	2
Henry	1	14	63	102	92	91	78	2
James	1	12	57	83	83	86	83	2
Jane	2	12	59	84	69	87	89	3

Janet	1	15	62	112	78	83	85	3
Jaffrey	1	13	63	84	68	79	92	3
John	1	12	59	99	86	68	94	1
Joyce	2	11	51	50	78	92	89	2
Judy	2	14	64	90	84	78	79	3
Louise	2	12	56	77	79	82	74	1
Mary	2	15	66	112	84	86	78	2
Philip	1	16	72	150	78	84	86	3
Robert	1	12	64	128	83	85	88	1
Thomas	1	11	57	85	88	79	83	2
William	1	15	66	112	86	73	89	3
Alice	2	14	70	107	89	78	92	1
Philip	1	13	57	122	98	94	93	3
John	1	12	59	122	98	94	93	3
James	1	11	62	146	90	88	89	1
Henry	1	12	63	132	96	89	90	2
Thomas	1	15	56	85	89	94	96	3
Judy	2	13	66	90	93	90	98	1
Marry	2	14	68	96	78	89	92	2
Jane	2	12	69	100	88	90	87	3
Alice	2	15	63	88	87	94	92	1
Ross	2	14	70	85	92	85	87	2

end data.

Variable labels name 'Student name'/
sex 'Student sex'/
age 'Student age'/
height 'Student height'/
weight 'Student weight'/
china 'Chinese test scores'/
eng 'English test scores'/
math 'Mathematic test scores'.

```
Value labels sex 1 'Male' 2 'Female' /
                class 1 'A class' 2 'B class' 3 'C class'.

* COMPUTE weight-kilogram, total average.
Compute Wtkilo = weight * 0.45.
Compute Totavg = (eng+china+math) / 3

* DESCRIPTIVE statistics (mean, standerd deviation, range).
Descriptives variable = age, height weight to math, wtkilo, totavg
        / statistics = 1 ,5 ,9
        / options = 5.

* COMPUTE frequencies table.
Frequencies variables = age height weight
        / statistics = mean skewness kurtoses minmum maximum variance.

Frequencies variables = sex to math totavg
        / format = limit(10)
        / statistics = default median.

Frequencies variables = age / barchart.

Frequencies variables = weight
        / histogram min(40) max (150) increment(10).

* COMPUTE cross table
Crosstabs sex by class.
```

```
Crosstabs tables=sex age by class.

Crosstabs tables=sex by class
        /optious=3  4  5
        /statistics=1  2  6  7  11.

* COMPUTET Test statistic.
T-test grpups=sex (1,2)
        /variables=age height weight.

finish.
```

2. STATISTICS

STATISTICS是一统计学软件包。它具有下列统计功能：排列组合、简单统计、偏斜度、峭度、偏差第一种分析、偏差第二种分析、线性回归、指数曲线拟合、对数曲线拟合、幂曲线拟合、多重线性回归、正态分布、F-分布、二元正态分布、对数正态分布、几何分布、泊松分布、二项分布、WEIEULL 分布、SPEARMANS 秩、列联表等。

该软件程序库所包括的统计学范围较为全面，是数理统计学工作者的一个好助手。该软件采用菜单显示所有统计函数，用户可根据菜单选择自己所需要的统计计算公式名称，然后按系统提示信息完成计算并输出结果。

3. STATPAK

STATPAK 是一统计学软件包。它的程序库包括八十多个程序，主要完成下列功能：概率计算、单变量分析、离散型分布函数、连续型分布函数、回归和相关、均值检验、观测值和列联表、非参量性统计、方差分析、时间序列等。

该软件的统计学功能较强，分类清晰，是统计学工作者的一

个较好的辅助工具。该软件采用菜单提示显示所有统计函数，用户可根据菜单选择自己所需要的统计计算公式的名称，然后按系统提示信息完成计算并输出结果。

7.6 工具软件

工具软件主要用于增强操作系统的功能。例如，具有计算器、日历、时间安排、磁盘和文件管理、调试、万能拷贝、加密解密、磁盘格式化、硬件诊断和测试等功能。专用工具软件往往只有一项功能，而综合性工具软件则具有多项功能。

微型计算机上常用的工具软件有 PCTOOLS，SIDEKICK，KEYWORKS，1DIR，Norton Utilities，Norton Commander，TIME LINE，PRINTSHOP，PRINT MASTER，Disk Explorer，Inside，Turbo Debug，Copywrite，Copy 等等。下面，仅就 PCTOOLS 来说明工具软件的功能。

PCTOOLS 是一个功能较强、操作简便的磁盘管理工具软件。PCTOOLS 5.1 版本是 PC TOOLS 的最新版本，亦称为 PC TOOLS 大全（Deluxe）。它包含以下几个部分：

1.PC SHELL

PC SHELL 是一个功能很强的公用程序。它具有一个使用方便的窗口环境，能够取代 DOS 的一些内部命令和外部命令，例如，拷贝、移动、删除、比较等功能。此外，它还具有以下功能：

(1) 在 PC SHELL 中使用鼠标器；
(2) 在 PC SHELL 中执行其它的应用程序；
(3) 恢复被误删的文件；
(4) 修剪、合并和重组子目录；
(5) 常驻内存程序的规划；
(6) 系统及磁盘数据的格式化；

(7) 文件和磁盘的编辑。

2.PC TOOLS DESKTOP

该桌上系统能将你桌上杂乱无章的笔记、计算器、备忘录、约会记事、电话薄、名片、顾客名单等有条不紊地整理到电脑中。

PC TOOLS DESKTOP 提供了下列九项功能:

(1) 笔记板 (Notepads)

具有文字移动、查找与替换、拼写检查、读取 ASCII 文件或 WORDSTAR 文件的字处理功能。

(2) 要点记事 (Outlines)

可随时在任何软件层次上使用完整的编辑功能来记事。

(3) 数据库 (Databases)

可利用多种功能来管理或整理数据，亦可使用与 dBASE 兼容的数据文件，而且含有一个自动电话拨号器 (phone dialer)。

(4) 约会安排器 (Appointment Scheduler)

建立约会时间安排表与执行表。根据需要可显示一天或一个星期的约会表，检查是否有冲突，而且可设定一个闹钟 (alarm) 来提醒你，并可显示笔记板列出约会记录、根据数据库的记录自动拨电话、使用电传来传输文件或在预定时间内执行某一程序。

(5) 电传 (Telecommunications)

传送或接收数据文件和程序文件。该功能可在后台执行，因此在传输文件时不会影响当前的工作。

(6) 键盘宏编辑器 (Macro Editor)

可随时记录和保存一串键盘输入，而使用单个键来代替。

(7) 剪贴板 (Clipboard)

通过一个临时存储空间在 PC TOOLS DESKTOP 和其它程序之间拷贝或修剪文字。

(8) 计算器 (Calculators)

适用于数学运算、财务和程序设计的计算器。

(9) 公用程序 (utilities)

具有选择 PC TOOLS DESKTOP 的热键。显示 ASCII 字符表、改变屏幕颜色、从内存中移掉 PC TOOLS DESKTOP 等功能。

3. PC BACKUP

该公用程序提供一个快速而又容易使用的备份数据的手段。它适用于各种不同的磁盘驱动器和磁盘格式。文件备份前可先将其压缩。

4. COMPRESS

该公用程序可提高磁盘驱动器的使用效率。它检查磁盘中数据的存放方式以及是否有空隙存在，然后改进存放方式。可将一子目录移至磁盘驱动器的前部或将所有子目录重新排序，使存取速度加快；而且可检查磁盘驱动器是否发生错误，并将出错的文件移出。

5.PC CACHE

将使用频度高的数据放在内存中，用以提高磁盘驱动器的存取效率，即减少电脑等待存取数据的时间。

6.PC FORMAT

该公用程序可替代 DOS 的 FORMAT.COM。它将磁盘格式化，所采取的格式能为 REBUILD 所恢复。

7.MIRROR / REBUILD

对于在意外情况下清除或格式化磁盘提供保护作用。

MIRROR 复制一份文件分配表和根目录，存放于隐藏文件中。如果意外将磁盘格式化，便可快速地将数据恢复。

由于在文件被删掉之前，MIRROR 已将删除动作的内部过程及其存放位置记录下来，因此当反悔时，便可根据 MIRROR 所记录的状态将数据恢复。

8.PC SECURE

它是程序和数据安全保密的一个强有力的工具。它可将文件中的数据加密、解密、压缩和隐藏。它使用 DEC 加密系统，将数据随机性处理，因此很难解密。文件压缩时，能使文件大小节省 25%～60%。

7.7 中文通用应用软件

中文通用应用软件一般是在西文通用应用较件基础上增加汉字处理功能而成的。不同种类的中文通用软件对汉字的使用和处理方法各不相同。

对于中文字处理软件，应具有中英文两种文字处理功能。用户可用文书编辑方式来编辑中英文混合的文章、信件、公文、报告、笔记、表格等，亦可用非文书编辑方式来编辑中英文兼容的程序文件和数据文件。

对于中文数据表软件，凡是能够处理文字数据的场合都应能处理汉字。例如，文字的输入、修改、删除、移动、复制、排序、打印功能亦适用于汉字处理。又例如，数据单元亦可用汉字命名。

对于中文 CAD／CAM 和图形软件，加到图形中的标题或文字解释应允许含有汉字。

对于中文 CAI 软件，学生与计算机的对话应允许使用汉字和显示汉字，计算机能用中文指出学生在学习过程中的错误。

对于中文组合软件，它的各个组成部分都应包含汉字处理功能。

对于中文软件族，它所包括的各个软件都应含有汉字处理功能。

对于中文统计软件，例如，SPSS，对数据加的注解和标记亦可含有汉字，打印的表格也应允许包含汉字。

对于中文工具软件，例如，PCTOOLS，PC SHELL 中的

浏览／编辑文件命令（View／Edit）和在文件中寻找字符串命令（Find）应适用于汉字处理；PC TOOLS DESKTOP 中的笔记板、要点记事、数据库、约会安排器、电传、剪贴板等均应能处理汉字。有些中文工具软件是专门为处理汉字而开发的，而不是在西文工具软件基础上汉化而成的，例如，软件汉化专用工具，汉字多字体打印程序，汉字排序程序，汉字内码转换程序（比如，简体字与繁体字转换程序，国标码与 IBM 5550 内码转换程序，台湾各内码之间的转换程序等）。

总之，中文通用应用软件与西文通用软件的主要区别是：

(1) 凡是在西文通用软件中使用和处理西文字母的场合，在中文通用应用软件中都能使用和处理汉字，例如，名字、文字数据、文本文件等场合。注意，由于一个汉字视为两个字符，因此在某些场合下，汉字处理功能要受一定的限制。

(2) 中文通用应用软件要充分考虑汉字的特点和中国人的习惯。例如，在中文字处理软件中，可通过 ESC 序列实现汉字字形变换等汉字打印功能，亦可通过规定键输入制表符实现中国人制表的要求。又例如，可用汉字多字体打印程序解释执行在汉字文本文件中插入的打印格式命令，用以实现汉字的多字体打印。

(3) 中文通用应用软件面向用户的界面应当汉化，也就是说，菜单及其它提示信息应当是中文。

本节仅就中文字处理软件 WORDSTAR 和中文计算机辅助绘图软件 AUTOCAD 的汉字处理功能，来说明中文通用应用软件的特点。

1. 中文 WORDSTAR

中文 WORDSTAR 是一个具有中英文两种文字处理功能的并具有中文菜单及其它中文提示信息的全屏幕编辑软件。用户可用文书编辑方式编辑中英文混合的文章、信件、公文、报告、笔记、表格等，也可用非文书编辑方式编辑中英文兼容的程序文件和数据文件。

考虑汉字文字编辑的特点（汉字占两个西文字符的位置），在使用中文 WORSTAR 时要注意以下几点：

(1) 光标移动键经过西文字符移动一个字符位置；经过汉字时移动一个汉字位置，相当于两个西文字符位置。

(2) 当插入汉字时，若插入的位置后面是汉字，则光标必须位于汉字的左半部；否则光标后面的汉字将错位半个汉字，会出现混乱现象。

(3) 当删除汉字时，光标一定要放在汉字的左半部；若将光标放在后半部，则将造成后面的汉字混乱。

(4) 当替换汉字时，光标必须位于汉字的左半部；否则会出现混乱现象。

(5) 当连续键入文字时，中文 WORDSTAR 会自动调整输入换行，西文以单词为单位换行，中文以单个汉字为单位换行，因此，每行右端不应出现半个汉字的禁则现象。

(6) 在中文 WORDSTAR 中可以使用中文操作系统中的打印控制命令。例如，在汉字文本文件中插入 Esc 序列等控制字符，当打印时遇到控制字符时，便解释执行它所代表的汉字打印功能（比如，字形变换等）（见 1.3.3 节 1.）。

2. 中文 AUTOCAD

由于计算机辅助绘图与设计软件包 AUTOCAD 是从国外引进的，因此它不能直接在绘制的图形中标注或书写汉字。然而，利用它的原有功能，可以在西文 AUTOCAD 基础上增加一个汉字处理外壳，用以构成中文 AUTOCAD。基本方法是：利用 AUTOCAD 本身提供的功能和命令，预先编制汉字图形库，当要在绘制的图形中加汉字（例如，标题、文字解释等）时，再从汉字图形库中调用汉字。

中文 AUTOCAD 的汉字图形库与一般中文操作系统中的汉字字库有着本质的区别。中文操作系统中的汉字字库一般为点阵字库；而中文 AUTOCAD 中的汉字图形库中存储的是一组向量

的起点、终点、方向、长度、以及抬、落笔等信息。这种汉字字形可以任意放大、缩小和旋转，而不会产生锯齿状。简言之，汉字图形库实质上就是一种图形库。

AUTOCAD 提供了两种建立用户图形库的方法：一种是将图形做成“块”，另一种是将图形做成“形”。块可用于图形复杂、信息量多的场合，形可用于图形简单、信息量少的场合，故后种方式较适合制作汉字图形库。制作的基本原理是：将每个汉字都做成一个“形体”，按规定把它们编辑成形体文件，存放在磁盘上，调用时按预先给每个汉字赋予的形体名称，用键盘或数字化图形输入板等输入方式将其从汉字图形库中调入图形。

AUTOCAD 规定：形体文件由若干个形体定义组成，每个形体定义均以一个如下形式的标题行开头：

*〈形体编号〉,〈定义字节数〉,〈形体名称〉

后面跟着若干行形体定义字节。

形体编号和形体名称标识所定义的形体。

定义字节数指明描述形体所用的字节个数。

形体定义字节用于按照有关形体的特殊规定描述形体的形状。有两种规定：一种是具有特定方向和特定长度的规定，另一种是些特殊码的规定。两种规定可以结合使用。具有特定方向和特定长度的规定，是在一个字节内对矢量的长度和方向进行编码：在一个字节中高四位表示矢量的长度（长度是图形单位的相对值，基准由调用形体时指定的高度决定），低四位为矢量方向编码。用具有特定方向和特定长度的规定构造一个形体定义虽然十分简单，用的信息量少，但不太灵活，只能画出十六种预定方向的矢量，而长度也只能是单位长的整数倍，即方向和长度只能是特定的；对于任意方向和任意长度的矢量，则显得无能为力。为了弥补上述规定的不足，AUTOCAD 规定了一组特殊码，用以扩充上述规定的功能，例如，借助 X-Y 位移画出非标准的矢量、画任意方向的圆弧、用多对 X-Y 位移来移笔、形体结束、

落笔、抬笔、尺寸控制、位置存入和取出等。尽管用特殊码定义形体比用特定方向和特定长度定义形体所用的字节数多，但却可以画出任意方向和任意长度的矢量。

如果汉字图形库以 AUTOCAD 规定的向量存储方式存放 GB2312 基本集中的一级汉字，那么，可按汉语拼音的第一个字母把各个汉字的形体定义划分为若干个音区，形体名称以汉语拼音的第一个字母开头后跟该汉字形体定义在所在音区的行列号命名。例如，A101 为“啊”字的形体名称。此时，用 LOAD 命令装入形体文件后，便可随时使用 SHAPE 命令把此形体文件中定义的任何形体调用到图形中。

由于中文 AUTOCAD 是在西文 AUTOCAD 外层上加汉字处理外壳形成的，因此不对 AUTOCAD 系统程序做任何修改，而是充分利用西文 AUTOCAD 的外部特性加以扩充，因此，一方面，保证了 AUTOCAD 系统功能不受汉化的影响，做到中西文完全兼容；另一方面，保证了汉字处理外壳不受西文 AUTOCAD 版本升级的影响，适应版本不断更新的需要，当西文 AUTOCAD 版本更新时，只要其中形的结构不变，汉字处理外壳不加修改或微作修改，便可继续使用。

中文 AUTOCAD 的汉字处理外壳也可以使用嵌入 AUTOCAD 内部的 AUTOLISP 程序设计语言编程，在西文 AUTOCAD 的原有命令基础上扩充有关汉字处理功能的新命令。用户使用这些新命令，可以建立汉字生成环境、从键盘向图形输入汉字或以文本方式向图形输入汉字，就象使用其它 AUTOCAD 命令一样。这些新命令与西文 AUTOCAD 的原有命令一起构成中文 AUTOCAD 的命令集。

第八章　中文应用系统

应用系统是适用于特定应用领域的计算机软硬件系统，本书中主要指应用系统中所包含的应用软件系统。应用软件可分为通用应用软件和专用应用软件。无论是通用应用软件，还是专用应用软件，它们的用户界面都是面向最终用户的。

鉴于管理信息系统是一类应用最广泛的应用系统，尤其是中文管理信息系统已广泛应用于我国的各行各业，本章介绍管理信息系统和中文管理信息系统的有关概念。

同其它软件的开发和维护过程一样，中文专用应用软件需要经历分析、设计、编程、测试、维护五个阶段，即使是选购中文通用应用软件，也需要经历分析和设计两个阶段，因此，本章较详细地讨论应用系统开发和维护的五个阶段，特别是软件汉化采用的测试原理和方法。除上述结构化开发生存期法之外，本章还简要介绍另一种软件开发方法——原型化(prototyping)方法。

中文应用系统与西文应用系统的主要区别是：中文应用系统具有汉字处理功能。目前，各种中文应用系统层出不穷。本章只简要介绍几种常用而又典型的中文应用系统：中文管理信息系统、中文桌上排版系统、英汉机器翻译系统。

8.1　管理信息系统

系统是一群个体的组合，通过彼此间的相互作用，达到特定的目标。任何一个工厂、机关、学校、商店、银行、医院等单位都可以看成为一个系统。如果我们用动态的观点分析系统的活动过程，就会发现有物质流和信息流在系统中流动。只考虑信息流

的系统称为信息系统。而对用于管理的信息进行收集、存储、处理和传输的信息系统称为管理信息系统（MIS: Management Information System）。

管理信息系统面向管理工作，为管理人员提供管理所需要的各种信息。虽然它不一定使用计算机，但由于现代管理工作的复杂性，一般都以计算机为基础。因此，管理信息系统是计算机、人和管理对象组成的人机系统。这种人机系统可以人尽其才，物尽其用。对于大量信息的快速处理和重复性的劳动，都让计算机来完成；而对于处理结果的分析、判断、决策等创造性劳动，则由人来完成。

1. 管理信息系统的分类

管理信息系统具有不同的种类。

按行业可分为企业管理信息系统、机关管理信息系统、商店管理信息系统等。

按管理内容可分为计划管理系统、生产管理系统、质量管理系统、技术管理系统、工资管理系统、设备管理系统、物质管理系统、财务管理系统、库存管理系统、合同管理系统、进货管理系统、销售管理系统、货运管理系统、进出口管理系统、商标注册管理系统、人事管理系统、档案管理系统、后勤管理系统等。

按管理工作的层次，可分为高层管理信息系统、中层管理信息系统和低层管理信息系统。

按组织和存储数据的方式，可分为使用文件系统和使用数据库的两种管理信息系统。

按参与决策的程度，可分为数据处理系统、统计分析系统、决策支持系统和专家系统。

按信息的内在属性，可分为军事、行政、社会性、经济性的各类管理信息系统。

按企业的规模，可分为大、中、小型管理信息系统。

按企业的产品种类，可分为机械、电子、化工等类型的管理

信息系统。

按生产过程，可分为连续型和离散型两种管理信息系统。

按处理作业的方式，可分为批处理和实时处理两种管理信息系统。

按各部分的联系方式，可分为集中式和分布式两种管理信息系统。

2. 管理信息系统的层次

管理工作大致可分为以下三个层次:

(1) 高层管理

实施策略性规划程序，拟定计划、预算和目标，决定赖以达到这些目标的策略。

(2) 中层管理

负责管理控制程序，拟定流程与测度标准，确保有效实施策略。

(3) 低层管理

低层管理亦称为操作管理，负责操作管理程序，执行目标，确保有效地实施特定的工作。

管理信息系统可统管各层的管理工作，亦可按其面向的管理工作的层次，分别建立各个层次的管理信息系统。

高层管理信息系统帮助高层管理人员进行市场预测、库存控制、质量分析等工作，由此作出决策，例如，产品的更新、重大革新项目的采用、新市场的开辟等。它以中、低层管理信息系统提供的数据为基础，运用一些数学方法，例如，回归分析、排队论、线性规划、优化理论等，对数据进行处理和分析，向高层管理人员提供倾向性的数据和各种图表。

中、低层管理信息系统帮助中、低层管理人员处理日常业务信息，例如，根据高层管理人员的决策下达的指标，完成这些指标的各种数据。这些表达日常业务信息的大量数据（例如，定购单、发票、支票、统计表、库存表、原材料消耗表、成品表等）

往往是每天、每周、每月都在发生变化，层出不穷。它的主要工作是数据处理，例如，计算、存储、排序、检索、制表等。从功能上讲，它把各个职能部门中各类工作人员（例如，会计、统计、出纳、仓库管理人员等）的工作过程编成程序存于计算机中，由计算机代替这些工作人员的大部分工作。

3. 管理信息系统的实现

以企业管理信息系统（简称为企业管理系统）为例，管理信息系统的实现步骤是：首先，进行系统调查，调查企业管理的内容和管理方式；然后，进行系统分析，建立企业管理模型；最后，进行系统设计，选购软硬件，开发应用程序，最终实现管理信息系统。实现之后，还要进行系统维护。

4. 管理信息系统的信息结构

从管理信息系统的信息结构来看，有的采用文件系统来组织信息，有的则是建立在数据库之上的。

在管理信息系统发展的初级阶段，往往是针对某一方面的管理工作而设计的，大多采用文件系统方式来组织信息。信息的组织可以优化地与应用系统结合起来。然而，文件系统对数据的完整性、安全性和保密性缺乏有效的统一的控制办法。

建立在数据库上的管理信息系统可以对信息进行统一管理，因此，数据可以做到没有冗余性，从而可保证数据的一致性。数据库管理系统提供了对数据的定义、存储、修改、检索等操作，提供了对数据完整性、安全性和保密性的统一控制，使得对数据的应用更为有效。因此，数据库广泛应用于管理信息系统。使用数据库的管理信息系统大致可分为两类：一类是采用大、中型计算机及若干台中、小型计算机构成的计算机系统，它适用于大、中型管理信息系统；另一类是采用若干台微型机在局部网支持下构成的微型机系统，它一般可用于中、小型管理信息系统。

5. 管理信息系统的评价

如何评价管理信息系统是一个复杂的问题。下面，从四个方

面简述管理信息系统的评价标准:

(1) 系统目标

系统是否达到原定目标，能否满足系统目标的要求；有哪些功能已超出原来设计的系统目标；由于系统环境变化或若干基础工作不完善而影响了哪些系统目标的实现；用户对系统使用性能的满意程度，其中包括使用方便性、可靠性、可用性、可维护性；系统操作是否齐全、清楚、实用、方便；等等。

(2) 系统技术性能

信息处理全过程所需时间，信息反馈所需要的时间；系统故障后的恢复时间；联机作业的响应时间；作业处理速度；外存容量利用率；等等。

(3) 系统使用性能

输入数据的正确、完整、统一和可靠性；各级管理部门使用输出数据的有效性；系统文档的完备性，文档是否方便和有效；系统的安全性，包括系统数据的备份，系统遭破坏后的恢复功能，备份管理制度的执行情况等；系统的保密性，保密字设置的有效性，保密制度与执行情况等，系统容错能力；操作的方便、灵活与及时性；等等。

(4) 系统经济性能

系统硬件、软件投资，附属设备、通信设备、机房投资等；系统应用软件开发费；人员培训费用；系统运行费用，包括系统维修费、折旧费、消耗品、管理费及工作人员的工资；等等。

6. 决策支持系统

用计算机支持数据处理、管理和决策，经历了下列几个阶段:

(1) 电子数据处理系统 (EDPS: Electronic Data Processing System)

计算机发展初期，主要用于科学计算。自五十年代起，开始用计算机进行数据处理。数据处理逐渐成为计算机应用的一个重

要领域。EDPS 辅助决策的工作方式是：对各种类型的数据（数值、文字、图象、声音等）进行收集、存储、传送、排序、统计、计算、综合、分组、摘要，并显示或打印报表和图表，输出图象等，以供决策者参考。

（2）管理信息系统（MIS：Management Information System）

到了六十年代，计算机广泛应用于管理工作，MIS 应运而生。MIS 收集、存储、分析管理所需要的各种信息，以供管理人员使用。MIS 辅助决策的工作方式是：运用系统分析和信息处理模型化的方法，将收集的与决策有关的信息（例如，市场、用户、上级、政策等）加以分析，并预测未来环境的改变，包括执行决策后果估计和实际执行所得结果的有关信息，从而有效地为各类管理决策提供所需要的信息。

（3）决策支持系统（DSS：Decision Support System）

DSS 出现于七十年代中期。DSS 是由 MIS 演变而来的。DSS 支持决策的工作方式是：为决策者提供一个分析问题、构造模型和模拟决策过程及其效果的决策环境，根据事先建立的判定原则和模拟模型为各种请求寻找最佳方案。

（4）专家系统（ES：Expert System）

ES 问世较早，但却在八十年代初期才得到普遍的重视。ES 是一种基于知识的计算机程序系统。它使计算机能象人类专家那样解决某方面领域的问题。ES 支持决策的工作方式是：将专家的知识预先存放在计算机中，并赋予计算机应用这些知识的能力，从而使计算机能象专家那样使用知识和推理过程去解决问题。这些问题是相当复杂的，原来解决这些问题往往需要有关专家。ES 使决策者可以借用专家的知识和经验，解决原来只有专家才能解决的复杂问题，制定合理可行的决策。

由此可见，EDPS 利用计算机存储和检索数据快的优点，迅速而自动地完成各种烦琐的事务性数据处理工作，例如，报表统

计、帐目计算、文书处理等。MIS 是 EDPS 的延伸与扩展，不仅进行数据处理，而且把数据转换为信息。它是建立在信息流和数据文件基础上的，主要面向中、低层管理人员，偏重于日常信息管理，通过对信息系统功能的综合和规划来适应管理的需要。DSS 是高级的 MIS，它对 MIS 的功能作了扩充和提高。它支持决策过程中的智能活动、设计活动和选择活动，主要面向高层管理人员或管理决策层。ES 是高级的或聪明的 DSS，是人工智能的热门课题。它吸取了专家的知识和经验，得到了专家的合作。

8.2 应用系统的开发方法

从软件工程的观点看，应用系统的开发同其它软件系统一样，最常用的方法是结构化开发生存期法，简称为结构化方法或生存期法。它把软件生存期分成系统分析、系统设计、编写程序、系统测试和系统维护五个阶段。前四个阶段称为开发期，最后一个阶段称为维护期。本节将分别按这五个阶段讨论结构化开发生存期法。值得注意的是，近年来发展起来的另一种软件开发方法——原型化开发方法。这种方法不过分强调软件开发的阶段性，而注重快速地塑造一种接近用户要求的工作模型，称为原型，在此基础上与用户合作快速地完善这一原型，直至满足用户要求为止。本节将简要介绍原型化开发方法。

8.2.1 系统分析

1. 可行性研究

在系统开发之前，要进行可行性研究，包括经济可行性、技术可行性和社会条件可行性，并形成一份可行性报告。可行性报告的格式不尽相同，但内容大体如下：

(1) 背景情况

介绍历史与现状、国内外水平、市场需求、经济前景等情

况;

(2) 各种选用方案

介绍各种选用方案的系统配置，对各种可能的方案进行比较;

(3) 系统描述

给出系统说明书的一个简写版本，包括总体方案和技术路线、课题分解、关键技术、计划目标和阶段目标;

(4) 经济可行性

对系统经济合理性进行评价，对系统进行价格-利益分析，即分析经济概算和预期的经济效益，对开发价格与从所开发的系统将得到的利益进行比较;

(5) 技术可行性

分析技术冒险的各种因素，包括技术实力、设备条件和已有的工作基础;

(6) 社会条件可行性

分析社会条件、管理体制、人员素质对系统的影响；确定由于系统开发可能引起的侵权、违法以及由此而承担的法律责任;

(7) 其它与系统有关的问题。

2. 系统分析的任务

系统分析亦称为需求分析，它的主要任务是定义用户要求。由于软件人员往往不了解用户的业务，而用户一般不熟悉计算机技术，因此，软件人员在为用户开发应用系统之前，应首先对用户的业务活动进行调查研究、分析和模拟，充分理解用户要求，明确用户要求系统“做什么”。

用户要求是指软件系统必须满足的所有性质和限制。它通常包括：功能要求、性能要求、可靠性要求、安全保密要求，以及开发进度、开发费用、资源限制等。

系统分析一般由系统分析员完成。系统分析员的任务是协调用户和软件开发人员之间的工作，分析和设计一个满足用户要求

的系统。系统分析员一般要求具有能熟练地描述问题、拟定解决问题的算法、选择适当的计算机硬件和软件、对最终运行的系统进行性能分析和评价等能力。

3. 系统说明书

通常以系统说明书的形式来定义用户要求，系统说明书是软件人员和用户双方经过充分讨论后达成的协议或签订的合用，是程序设计的基础、程序测试的标准、程序验收的依据。一方面，系统说明书中的功能描述应尽量准确、无二义性；另一方面，系统说明书应尽量简明易懂，尽量不使用计算机的概念和术语，不考虑计算机的各种指标，使用户和软件人员都能接受它。尽管有一种用形式语言书写功能描述的趋势，但目前还没有真正可供使用的形式语言，而且用户一般很难接受这种形式语言，因此，目前仍采用自然语言和图表来书写系统说明书。

系统说明书的主要内容如下:

(1) 范围

确定系统的作用范围，主要包括功能、性能、可靠性、界面等问题。

简要描述整个系统的功能，只讲系统“做什么”，而不讲怎样实现功能。在可能情况下要作进一步分解，以提供更多的子功能描述。

描述系统的性能特征，包括存储约束、响应时间、与机器相关的特点等。

描述系统界面，即本系统与计算机系统的其它部分（硬件、软件、人等）之间的功能联系。硬件界面包括计算机特性、内外存容量、输入输出设备能力等。软件界面包括操作系统特性、公用程序和支持软件以及它们相互之间的连接特性。

尽管软件可靠性至今尚不能精确地描述，但对于具有特殊性质的软件可以要求特殊考虑，以保证可靠性。

(2) 资源

资源包括人力资源、硬件资源和软件资源。每种资源均从资源的描述、对资源要求的日程表时间以及对资源应用的持续时间三个方面来说明。

人力资源是系统开发的最重要的资源。人力资源描述包括需要的人数、每个人的技术水平以及工作的持续性。

硬件资源除了计算机硬件本身外，还应考虑所需要的特殊测试设备及其它各种硬件支持等。

软件资源包括用于系统开发的各种支持软件，例如，操作系统、编译程序、数据库管理系统、通用应用软件、测试工具等。

(3) 进度

根据系统规定的交付日期，协调可用资源和工作量，合理安排系统开发进度，制定进度表。

(4) 成本

对系统进行成本估算，包括人力、机时、设备、软件、办公费用等。

(5) 数据流分析

用数据流程图和数据词典给出分层数据流分析的文字和图形描述（详见本节 4.）。它们是系统说明书的主要组成部分。

(6) 质量评审要求

规定系统功能和性能的正确确认需求和测试限值。

4. 数据流分析

数据流分析是一种结构化分析（SA: Structured Analysis）方法。它适用于分析大型的数据处理系统，特别是管理信息系统。这个方法通常与系统设计的 SD 方法衔接起来使用。

SA 方法控制复杂性的两个基本手段是：分解和抽象。它把一个大而复杂的系统自顶向下逐层分解为容易理解和表达的子系统。逐层分解也体现了抽象的原则，上一层就是下一层的抽象。这种抽象使人们不致于陷入细节，而是有控制地逐步地了解更多的细节。对于任何复杂的系统，分析工作都可以按照这样的方式

有计划、有步骤、有条不紊地进行。系统规模再大，分析工作的复杂程度也不会随之增大，而只是多分解几层而已。

数据流分析方法采用数据流程图与数据词典结合的方式来描述系统。这种描述方法介于形式语言与自然语言之间，虽不如形式语言精确，但简明易读，所表达的意义也比较明确。

(1) 数据流程图 (DFD: Data Flow Diagram)

数据流程图是一种描述数据流程和加工的图形表示。它描述系统由哪些部分组成以及各部分之间有什么联系。

图 8.1 给出飞机订票系统的数据流程图。

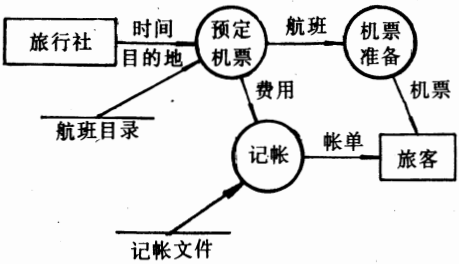


图 8.1 数据流程图

从图中可以看出，数据流程图由以下四种基本符号组成:

①数据流，用箭头表示，它代表数据的逻辑运动方向;

②加工，用圆圈表示，它代表人工处理或计算机处理的广义数据加工;

③数据存储，用横线表示，它代表抽象的逻辑存储，即数据文件;

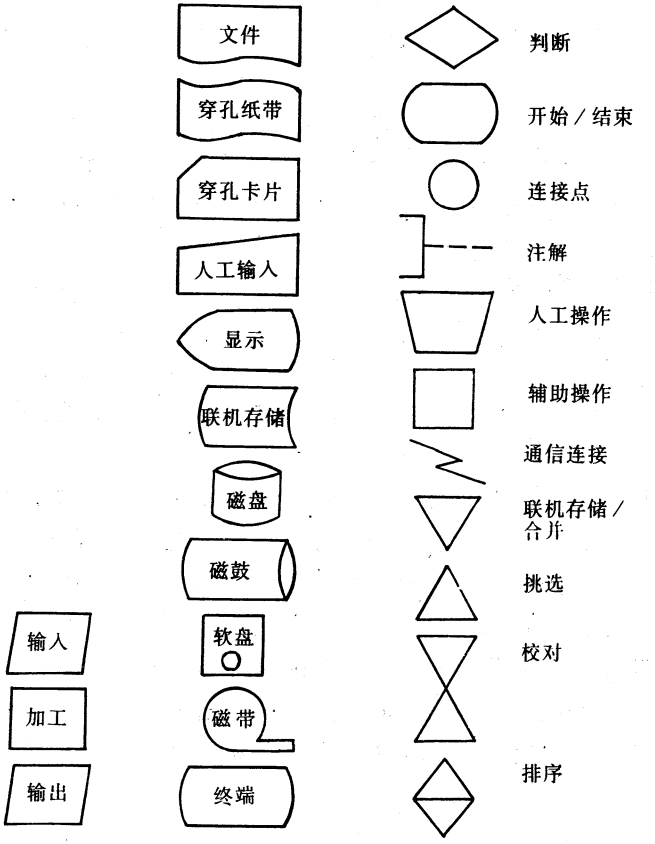
④外部实体，用方框表示，它代表在系统之外的数据来源或信息去向。

数据流程图与程序流程图是不同的。数据流程图是按数据处理过程来描述一个系统；而程序流程图表达的是程序执行的次

序。数据流程图中的箭头表示数据流；而程序流程图中的箭头表示控制流。由于数据处理系统的中心问题是对数据进行变换和传送，系统中业务活动过程恰好就是数据处理过程，因此，在宏观地分析系统的业务概况时，用数据流程图来描述系统的业务活动过程是比较合适的；而程序流程图只适用于描述系统中某个加工的执行细节。

在系统分析中，常采用自顶向下逐层画数据流程图的方法，来描述系统的逐层分解过程。

系统流程图（System Flowchart）是一种常用的数据流程图。它强调数据的输入输出，至于如何将输入数据转换为所希望的输出，则只提供简单的说明。系统流程图中使用的标准符号如图 8.2 所示。图 8.2(a)给出了输入／输出和加工的基本符号。尽管输入／输出的基本符号可用来表示任何类型的媒体或装置，但通常使用其它符号来表示特定的媒体或装置（见图 8.2(b)），用于取代输入／输出的基本符号。此外，图 8.2(c)还给出了其它一些系统流程图符号。其中，人工操作是指为适应人的速度而进行的任何脱机处理；辅助操作是指用来辅助主处理功能，但由一非直接受控于 CPU 的机器执行的操作；脱机存储是指信息存储在脱机存储媒体上，例如，卡片、纸带、磁带、报表等；注解标识被虚线连接到系统流程图上，在方括号中提供注释。虚线可以画在左边，也可以画在右边，方括号也可朝右或朝左。



(a) 输入／输出和加工的基本符号

(b) 表示特定媒体或装置的输入／输出符号

(c) 其它符号

图 8.2 系统流程图标准符号

(2) 数据词典

数据流程图只描述了系统的分解，但没有说明系统中各个数据和加工是什么含义。数据词典就是用来描述数据和加工的。

数据词典要对数据流程图中出现的所有名字（数据流、加工、文件等）进行定义。就象日常使用的词典一样，借助于数据词典可查出某个名字的具体含义。词典中的所有条目应该按一定的次序排列起来，以方便查阅。

数据流分析方法把一个复杂的系统分解成许多个足够简单的不能再被分解的基本加工。为了理解这些基本加工，要为每个基本加工编写详细的“小说明”。

数据流程图中的每一个基本加工都必须有一个小说明，用以给出这个加工的精确描述，而对数据流程图中的其它加工，则可以没有小说明，这是因为任何一个加工最后终能分解成一些基本加工，只要有了基本加工的小说明及其有关描述，就可以理解其它加工。

小说明中应精确描述一个加工“做什么”，即加工逻辑及其它一些与加工有关的信息，如执行条件、优先级、执行频率、出错处理等。加工逻辑是指用户对这个加工的逻辑要求，即该加工的输出数据流和输入数据流之间的逻辑关系。

理想的小说明应该既严格精确又容易被软件人员和用户理解。由于至今尚未研究出描述加工逻辑的形式语言，目前小说明通常仍采用自然语言、结构化自然语言、判定表和判定树来描述。

结构化自然语言是介于自然语言和形式语言之间的一种半形式语言，它是自然语言的一个受某些限制的子集。它虽然没有形式语言那样精确，但具有自然语言简单易懂的优点，又避免了自然语言的一些缺点。结构化自然语言的语法通常可分为内外两层。外层语法描述操作的控制结构，如顺序、选择、循环等，这些控制结构将加工中的各个操作连接起来。内层语法一般没有什

么限制，但描述方式最好带有一定格式。

判定表用来描述一些不易用语言表达清楚或用语言需要很大篇幅才能表达清楚的加工。例如，在飞机订票系统中，“计算折扣量”的加工逻辑是：在旅游旺季 7–9，12 月份，如果订票量超过 20 张，则优惠票价的 15%；20 张以下，则优惠 5%；在旅游淡季 1–6，10，11 月份，如果订票量超过 20 张，则优惠票价的 30%，20 张以下，则优惠 20%。显然，对于上述这样执行的操作取决于一组条件的加工逻辑，用自然语言来描述是不易理解的，而用判定表来描述是比较合适的。上述加工逻辑可用判定表描述，如图 8.3 所示。

判定表由四部分组成。左上部分列出决定这组条件的对象，右上部分列出各种可能的条件组合，左下部分列出所有的操作，右下部分说明在相应的条件组合下，某个操作是否要执行。

判定树是以图形方式描述加工逻辑的，它本质上同判定表是一样的。上述加工逻辑可用判定树描述，如图 8.3 所示。

旅游时间	7–9, 12 月		1–6, 10, 11 月	
订票量	≤20	>20	≤20	>20
折扣量	5%	15%	20%	30%

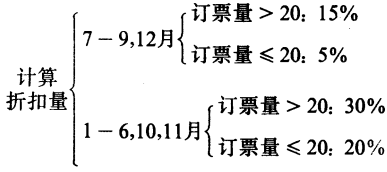


图 8.3　判定表与判定树

8.2.2　系统设计

系统分析集中考虑系统“做什么”，尽量少考虑怎样实现的问

题。而系统设计则应主要考虑“怎样做”才能满足用户的要求。系统设计的主要任务是把用户要求转变为一个具体的设计方案。

系统设计可分为总体设计和详细设计两个阶段。总体设计集中于系统的总体模块结构，而详细设计集中于模块实现的细节。

1. 总体设计

总体设计根据系统说明书中的功能描述，决定系统的模块结构。它主要考虑以下几个问题:

(1) 如何将系统划分成一个个模块;

(2) 模块间传送什么数据;

(3) 模块间的调用关系如何;

(4) 如何评价模块结构的质量。

系统设计的主要工作结果是设计说明书。它主要包括两个部分:

(1) 模块结构图;

(2) 模块功能描述。

模块结构图类似于程序流程图。它可形象地描述系统由哪些模块组成，并说明模块间的调用关系。每个方框表示一个模块，方框中的模块名要尽量反映这个模块的功能。模块结构图不一定是树型，但必须是分层次的。从一个模块指向另一个模块的箭头表示上一层模块中含有对下一层模块的调用。例如，通信录管理系统的模块结构图如图 8.4 所示。

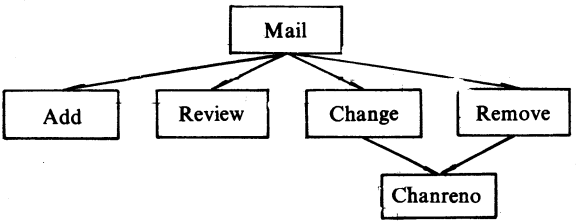


图 8.4 模块结构图

模块的功能描述用来说明每个模块的输入、输出以及这个模

块的功能。

在众多的设计方法中，最常用的是结构化设计（SD: Structured Design）方法和 Jackson 方法。它们适用于软件系统的总体设计。

SD 方法通常与系统分析的 SA 方法衔接起来使用。系统分析的 SA 方法获得用数据流程图和数据词典描述的系统说明书，而 SD 方法则以数据流程图为基础设计模块结构。SD 方法采用自顶向下逐步求精的方式，把一个大而复杂的系统分解为若干个相对独立的单一功能的模块，每个模块最终解决一个具体问题。由于模块之间是相对独立的，因此可对每个模块单独进行设计、编写程序、测试和修改。这种模块化方法既提高了系统的质量(易理解性、易维护性、可靠性等)，又简化了开发工作。SD 方法的独到之处是提出了评价模块结构质量的标准：模块间联系越小，模块内联系越大，则模块的相对独立性就越高。为了提高模块结构的质量，应当尽量减少模块间的联系，增大模块内各成分之间的联系。

Jackson 方法的某些基本思想同 SD 方法是一致的，例如，模块化、自顶向下逐步求精、程序结构与问题结构相对应等。但是，Jackson 方法不是在数据流程图基础上，而是在数据结构基础上建立模块结构，从数据结构导出程序结构。

2. 详细设计

总体设计确定了系统的模块总体结构和接口描述。而详细设计则给出各个模块具体的处理过程描述，具体考虑每一模块内部采用什么算法。根据这些描述，程序设计人员就能很快地写出程序来。当然，如果程序设计人员的素质很好，则可以不经过详细设计阶段，而直接将模块的功能与接口描述交给程序设计人员，至于程序设计人员采用什么算法来实现则完全由他们自己决定，只要编出的程序符合总体设计要求即可。

详细设计通常使用图形、表格、结构化自然语言、自然语言

等方式来描述处理过程。

可用于描述处理过程的图形工具种类很多，例如，程序流程图（程序框图）、N-S 图、IPO 图、Warnier-Orr 图、PAD 图等。程序流程图是最简单而且使用最广泛的一种描述处理过程的图形表示。已经证明，只要有图 8.5 那样的三种基本控制结构，就可构造出所有形式的程序流程图。程序流程图中的方框是处理框，菱形框是判定框，箭头表示控制流。除上述基本符号外，根据需要，程序流程图中还可以引入其它符号，例如，用平行四边形代表输入／输出框。

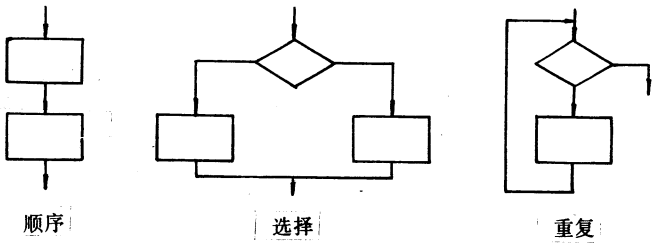


图 8.5　三种基本控制结构

程序流程图的优点是直观、清晰、易懂，便于检查、修改和交流。一方面，流程图是一种程序设计工具，程序设计人员可以通过画流程图构思程序的逻辑结构，作为编写程序的依据；另一方面，可以作为测试和调试程序的参考资料，亦可留作以后维护程序时使用。

程序流程图的缺点是：在某些情况下往往难以设计，有时比设计程序还困难；流程图只能表示程序结构，而不能表示数据结构；流程图是一种非结构设计，与结构程序设计相冲突，因此在结构程序设计方法出现以后，它一般只作为程序设计的辅助工具。

伪码是一种结构化语言，它采用程序语言与自然语言混合的

方式来描述处理过程。伪码使用一种程序语言的语法，而在程序语句或命令中嵌有一种自然语言的句子或词汇。

例如，在 PASCAL 语言程序中嵌有中文或英文的伪码分别如下：

伪码（中文）：

IF　　表的当前项大于表的下一项

THEN　互换这两项

ELSE　用缓冲区第 1 项取代当前项，用缓冲区第 2 项取代下一项

伪码（英文）：

IF　　current item of list > next item of list

THEN　interchange both items

ELSE　replace current item with buffer entry 1,

　　replace next item with buffer entry 2

显然，伪码具有下列优点：

(1) 利用程序语言关键字的固定语法，便于提供结构化、格式化、模块化结构；

(2) 利用自然语言的自由语法，便于灵活描述处理过程；

(3) 利用子程序定义和调用，便于提供各种接口描述模式；

(4) 由于伪码引进了程序语言的语法，实际上已写了一部分程序，因此，从伪码变换到程序是比较容易的，只要将其中的自然语言描述翻译为程序即可。

例如，上述伪码可变换为下列 PASCAL 程序：

```
IF  LIST (I) > LIST (I+1)
THEN BEGIN
        TEMP: = LIST (I);
        LIST (I): = LIST (I+1);
        LIST (I+1): = TEMP
     END
```

```
ELSE BEGIN
        LIST (I):=BUFFER (1);
        LIST (I+1):=BUFFER (2)
    END;
```

3. 设计说明书

设计说明书是系统设计的主要工作结果，是编写程序、系统测试、系统维护的依据。设计说明书的主要内容如下：

(1) 概述

给出系统功能和结构的总体描述。

(2) 模块结构图

分层给出模块结构图（见本节 1.）

(3) 文件结构和全局数据

给出外部文件描述，包括每个外部文件的名称、结构、特性和访问权限的约定；给出系统各模块所共享的全局数据的结构和存取模式。

(4) 模块描述

给出模块结构图中每个模块的功能描述和处理过程描述。

模块功能描述说明每个模块的功能、模块内部的数据组织和文件处理、模块外部界面，包括调用它的模块、被它调用的模块、与它有关的数据流信息、文件和对标准输入／输出设备的读写。

模块功能描述通常用自然语言、结构化自然语言来表达。模块功能描述同样可按词典的方式组织起来：每个模块在词典中各有一个条目，条目中列出这个模块的名字、输入数据、输出数据、功能及其它种种性质和限制等。

模块处理过程描述说明每个模块的处理过程，通常用图表、结构化自然语言、自然语言来表达，最常用的描述工具是程序流程图和伪码。

模块结构图和模块功能描述是总结设计阶段的工作结果；模

块处理过程描述是详细设计阶段的工作结果。若系统分析只有总体设计阶段而无详细设计阶段，则设计说明书中不包含模块处理过程描述。

如果决定购买软件，那么，当系统分析和总体设计完成后，就可以选购软件了。选购的软件必须符合系统说明书和设计说明书的功能描述。如果有适当的软件可供选购，则系统开发的下述步骤可省略，否则必须继续下述步骤。

8.2.3 编写程序

编写程序的任务是把设计说明书转变为用某种程序语言编写的程序。也就是说，根据设计说明书中的模块结构图和模块功能描述编写程序。如果设计说明书中包含模块处理过程描述，则可把模块处理过描述中的程序流程图或伪码等直接转变为用某种程序语言编写的程序。

1. 结构化程序设计

结构化程序设计（SP: Structured Programming）是编写程序的一种基本技术。可以证明，任何程序逻辑都可用顺序、选择、重复这三种基本控制结构来表示（见图 8.5）。它们具有一个共同的特征：每种结构只有一个入口和一个出口。SP 方法用这三种基本控制结构反复嵌套构成结构化程序。结构化程序有如下优点：

(1) 可用自顶向下逐步求精的方式编写程序；

(2) 程序易于阅读、理解、测试、排错和修改；

(3) 程序易于验证其正确性。

这种结构化程序与含有 GOTO 语句的非结构化程序形成了鲜明的对照，消除了 GOTO 语句对程序结构的有害影响。

大多数高级程序语言都包含有表示三种基本控制结构的语句，也含有 CALL 一类的过程调用语句，由此可将程序设计为模块结构。

2. 程序的评价标准

一个可采纳的程序，起码必须是正确的，也就是说，它必须符合系统说明书和设计说明书的功能描述。然而，一个正确的程序不一定是一个好的程序。一个好的应用程序，至少应当注重以下几个方面:

(1) 程序语言的选择

应用程序的质量与程序语言的选择关系很大，程序语言的功能和特性对于程序质量有着直接的影响，因此，为了提高应用程序的质量，必须选择适合的程序语言。选择程序语言要考虑下述因素:

①应用领域（例如，商用数据处理、科学计算、实时处理、系统程序设计、人工智能等);

②算法和计算复杂性;

③系统的执行环境（例如，机器已配置的程序语言);

④性能（例如，实时处理要求响应时间快);

⑤数据结构的复杂性。

(2) 应用程序的用户界面

由于应用程序的用户大多是非计算机专业的最终用户，因此应用程序应尤为重视用户界面（包括屏幕显示、打印、键盘输入等)，要提供友好的用户界面。屏幕显示编排应适宜，层次应清晰。应采用菜单方式引导用户按菜单提示去选择功能，通过人机对话的交互方式进行分支处理。菜单提示应是汉字的，并尽量采用行业术语，而不采用程序设计或数据处理的概念。用户容易出错的地方，应采用确认方式来保证安全性。此外，键盘输入应尽量减少按键次数。充分利用汉字字体的多样性，实现多种字体的混合显示和打印。打印报表格式灵活，使用方便，不仅能打印各种表格线，而且最好能打印多种字体。如果应用程序中包含报表生成器和图形生成器，则更为理想。

最好采用菜单驱动的模块结构，主程序模块中包含主菜单，

主菜单中每个选择项各对应一个子程序模块，而每个子程序模块中又包含一个子菜单，子菜单中的每个选择项又各对应一个下一层的子程序模块，依次类推。菜单将程序的功能显示在屏幕上，并将每一功能对应一个选择项，菜单提示引导用户操作，使用户象点菜一样，通过把光标置于选择项或键入它的代表字母或数字，便去执行一个功能。由于菜单的使用，可使模块之间的联系是单入口单出口的，即只有一条途径可以通过主菜单到达子模块，而且只有一个出口返回到主菜单，主程序模块通过主菜单控制其它全部子程序模块，所有子程序模块都返回调用它们的程序模块，而且用户从整个程序正常出口的唯一途径是通过主菜单。

(3) 应用程序的可阅读性

由于程序经常要被人阅读，例如，测试、调试、维护时都需要阅读程序（读程序是发现错误的有效手段），读程序的时间往往比写程序的时间还要多，因此程序的可阅读性是十分重要的，尤其是程序规模较大时显得更为重要。

程序实际上是供人阅读的一种特殊文章，只不过它是用程序语言而不是用自然语言写的。一个逻辑上绝对正确，但杂乱无章的程序是没有什么价值的，因为它无法供人阅读，因此难于测试、调试和维护。

同编写文章一样，程序设计人员在编写程序时，就要做好以后再阅读这个程序的准备（无论是您本人还是其他人）。宁可在编写程序时多化些精力，讲究程序设计风格，使程序具有良好的可阅读性，具有良好的文体，这将大大节省读程序的时间，总的来说这样做是值得的。

为了提高程序的可阅读性，起码要做到结构良好，层次清楚，思路清晰。

要注意程序的书写格式。通过使用统一的书写格式来写程序的相关部分，能够实质性地增加程序的易读性。书写格式应当反

映程序的逻辑结构。可用锯齿状行首缩进来表明哪些行是相关的，使程序更易读。对于嵌套结构，行首缩进更为有用，使人乍眼一看就能确定控制结构是否终止。

程序中适当地加上注释，可以使程序成为一篇自我解释的“文章”，读程序时就不必再翻其它说明材料，因而使用注释是提高程序的可阅读性的有利手段。

注释可分为以下四种:

①序言性注释

序言性注释安排在每个程序模块开头，特别是主程序模块开头，用于指明模块名、标题、作者、编写日期、主要功能、参数、重要变量的名字、用途、约束和限制及其它有关信息。大型软件系统一般都是由多人同时并行编写程序的，通常以模块为单位分工，为了便于测试和维护，为每个模块书写序言性注释是必要的。

②功能性注释

功能性注释用于说明一段程序的功能，通常放在这段程序之前。

③状态性注释

状态性注释用于说明一段程序执行后的数据状态，通常放在这段程序之后。

④标识性注释

可在每个控制结构的结尾行用标识性注释来重复开始行中的条件，由此可清楚地看出哪一个开括号与哪一个闭括号匹配，起到嵌套结构的标识作用。

精心选择的有意义的标识符或名字，亦可以在使用它们的场合中把它们看成是注释。如果能够按统一标准适当地命名，使变量名、文件名等能够贴切地代表它们的用途，则可以减少所需注释的数量。

程序中加注释需利用程序语言中提供的注释功能，例如，注

解语句等。

阅读结构化程序可采用自顶向下或自底向上的方法。如果程序带有较详细的注释或其它说明材料，则可采用自顶向下逐步细化的方式来阅读，此时可充分利用中间各层的注释或说明帮助理解，直至最下层。如果程序的注释很少，则宜采用自底向上逐步求抽象的方式阅读，此时可以逐层补写出中间各层次的抽象说明，直至最上层。

(4) 应用程序的错误检测和校正能力

应用程序的查错功能应当是自封闭的。也就是说，使用应用程序而出现的错误应由应用程序输出出错信息，而绝不应出现程序语言的解释系统或编译系统产生的出错信息（例如，语法错误、语义错误等），更不能出现操作系统产生的出错信息。否则，当出错时，不懂计算机技术的用户会不知所措，陷入困境。除有较强的查错功能外，最好还要有较强的错误校正能力。

应用程序最好设有错误捕获程序，以确保考虑用户可能会出现的所有潜在错误，例如，如果想要用户键入 Y 或者 N 回答是／否问题，则必须确保用户在键入其它错误符号时，程序知道应做什么。这一概念称为废进废出（GIGO: Garbage-In-Garbage-Out），即无用输入无用输出，它意味着输出直接与输入有关，输入错误数据，输出亦无效。解决废进废出问题的程序称为错误捕获程序。

(5) 应用程序的时间效率和空间效率

为了保证应用程序的质量，必须提高应用程序的时间效率和空间效率。下面，仅就如何提高应用程序的时间效率和空间效率提出以下几点建议，供大家参考。

①选择合适的程序语言，充分利用程序语言的功能和特性。

②选择最佳算法，讲究程序设计技巧和方法。

③合并公用子表达式。当一子表达式在程序中重复出现多次时，要形成一条把这个子表达式的值赋给一个变量的赋值语句，

并用这个变量来替换这个表达式的值未发生改变的所有出现。

④把循环内不改变值的运算尽量移到循环之外。

⑤削减循环内的运算强度，例如，把乘法改变为加法。

⑥尽量避免使用多维数组，因为下标地址计算强度较大。

⑦尽可能避免使用指针及复杂的表结构。

⑧尽量减少磁盘的访问次数，以防影响程序执行的速度。

⑨把一个大型的数据文件设计成若干较小的但又相关的数据文件，可以减少磁盘的访问时间。对于一个大型的数据文件，如果每次仅仅存取一小部分数据，就需要访问一次磁盘，而在磁盘上存取大的文件需要化费较长的时间，因此应把经常存取的数据放在一个较小的数据文件中。由于数据文件较小，将使程序运行加快。

⑩对于大量复杂计算或使用频度很高的算法，要用汇编语言或机器语言来编写程序，而利用程序语言的外部接口功能，在程序中调用和运行汇编语言程序或机器语言程序，从而可大大提高运行速度。

⑪查找数据应尽量采用有效的查找算法，用以提高查找速度。

⑫不提倡把汉字用作变量名、文件名等，这是因为汉字的处理速度较慢。

⑬利用虚拟存储技术，解决内存不够用的问题。

⑭利用覆盖技术，解决小内存运行大程序的问题。覆盖技术的基本思想是：把内存分为常驻区和覆盖区，把待运行的程序分为主模块和各层子模块，并把主模块放在常驻区，而把那些不会同时被调用的子模块安排在同一个覆盖区；当程序执行时，需要用到哪个子模块就把哪个子模块调到预定的覆盖区，这样多个子模块就可以共用一个覆盖区，从而减小了运行整个程序所需要的内存量。

8.2.4 系统测试

编完程序后就要测试和调试它。所谓测试，就是设计一些例子去执行程序，从中发现程序的错误。所谓调试，就是根据出错症状查找程序的出错位置和出错原因，并纠正和排除错误。出错症状是在测试过程中或实用过程中执行程序时发现的。测试与调试是同时进行的。测试与调试的工作量往往占整个软件开发工作量的一半以上。

程序错误大致可分为两类：语法错误和逻辑错误。语法错误的产生主要是由于未按程序语言的语法规定编写程序。通过人工对静态程序结构认真检查。或通过编译程序、解释程序、汇编程序，可检查出程序中的语法错误。逻辑错误主要指程序在逻辑上的错误。您可能认为您的程序是正确的，而程序却并不按您所希望的去做。这是由于程序逻辑与您的算法不一致造成的，往往是由于程序的动态执行过程与预期的结果不符合而引起的。逻辑错误是无法用编译程序、解释程序、汇编程序查出来的，常常在程序执行时才能暴露出来。测试就是有意识地设计例子去执行程序，有意识地发现程序的错误。调试就是查找并纠正错误。

1. 测试方法

首先，让我们通过一个例子来看看测试工作是如何进行的。

程序 Triangle 输入三个整数，它们表示一个三角形的三条边长，该程序产生的结果指出该三角形是等腰三角形、等边三角形还是不等边三角形。

为了测试这个程序，随手可写出一些用于测试的例子，例如，边长 a，b，c 分别为 3，4，5，或 5，5，6 或 6，6，6 等。然而，如果对这些测试用例，程序都能给出正确的结果，那么是否可以认为这个程序通过了测试呢？回答是否定的。上述这些测试用例只是几种可能的情况，完整的测试至少应包括下面这些情况：

①合理的不等边三角形（即输入的数据满足两边之和大于第三边，例如，输入数据 1，2，3 或 2，5，10 就不能算作这一类）；

②合理的等边三角形（例如，输入数据 0，0，0 就不能算作这一类）；

③合理的等腰三角形（例如，输入数据 2，2，4 就不能算作这一类）；

④等腰三角形的三种排列次序（例如，边长 3，3，4 的等腰三角形输入数据时的三种排列次序为：3，3，4；3，4，3 和 4，3，3）；

⑤三个正数，其中两个之和等于第三个；

⑥第 5 种情况的三种排列次序（例如，1，2，3；1，3，2 和 3，1，2）；

⑦三个正数，其中两个之和小于第三个；

⑧第 7 种情况的三种排列次序（例如，1，2，4；1，4，2 和 4，1，2）；

⑨输入数据中含有零值；

⑩输入数据中含有负数；

⑪输入数据中含有小数值；

⑫三个数均为零；

⑬输入数据不是三个数（例如，只有两个输入数据）。

上述十三种情况是根据常见错误列举出来的，因此它们确实是测试中应当考核的。然而，对于这样的简单的小程序，即使是有经验的软件设计人员，也经常会考虑不全。更何况，即使测试用例中包括了上述十三种情况，也不能保证经过测试的程序就不再含有其它错误。这个例子说明：即使测试一个很小的程序，也不是轻而易举的。

测试的关键是如何设计测试用例。设计测试用例的方法一般有以下两类：

①黑箱法

黑箱法把要测试的程序视为一个“黑箱”。也就是说，它不关心程序的内部结构和内部逻辑，而只是根据程序的功能描述来设计测试用例。黑箱法是在程序界面上进行测试，看它能否满足功能描述，输入能否正确地接收，输出结果是否正确。

如果想用黑箱法发现程序中的所有错误，则必须用输入数据的所有可能值来检查程序是否都产生正确的结果。例如，一个简单的程序有两个输入变量 x，y，一个输出变量 z，假定程序是在 32 位机上运行，又假定 x，y 都是整数，则输入数据的可能值有 $2^{32}\times 2^{32}=2^{64}$ 种。如果这个程序执行一次约需一毫秒，那么用所有这些数据测试这个程序将需要 5 亿年！因此要试遍所有数据是不可能的。

②白箱法

白箱法把要测试的程序视为一个透明的“白箱”。也就是说，需要了解程序的内部结构，根据程序的内部逻辑设计测试用例。

如果想用白箱法发现程序中的错误，则至少必须使程序中每种可能的路径都执行一遍。例如，一个小程序由一个循环语句组成，循环次数可达 20 次，循环体中是一嵌套的分情况语句，其可能的路径有 5 条，所以从程序的入口 A 到出口 B 的路径数就达 $5^{20}\approx 10^{14}$ 条。如果编写一个例子，用它来测试这个程序一次要用 1 秒钟，则试遍全部路径要花费 500 万年！因此，同试遍所有的输入数据一样，要试遍所有路径也是不可能的。

由此可见，测试只能证明错误的存在，但不能证明错误不存在。测试是假定程序中存在错误，因而想通过执行这个程序来发现尽可能多的错误，而不是为了证明这个程序能正确地执行它应有的功能。显然，那些只能使程序正确执行的例子是没有意义的，而能够发现错误的例子才是有意义的测试用例。

由于对一个程序进行完全彻底的测试是不可能的，因此经过测试的软件系统不可能达到 100%的可靠性。只有通过程序证

明，证明程序能够完成给定的功能描述所规定的功能，才能真正确认程序是正确的。然而，由于形式证明的复杂性，至今很少有人真正用形式证明技术去验证他们的程序。因此，截目前为止，测试仍是程序确认的主要方法，仍是软件可靠性的唯一现实保证。在程序测试过程中，最好用程序证明的推理风格去指导测试，把程序测试与证明结合起来。经过严格全面的测试，可以使软件系统达到实际使用所提出的可靠性要求。

(1) 黑箱法

黑箱法设计测试用例的方法有以下几种:

①等价分类法

如果想用黑箱法发现程序中的所有错误，则必须用所有可能的输入数据来测试这个程序，然而这是不可能的。我们只能在输入数据中选择一个子集，因此，问题就归结为如何选择一个适当的子集，使它尽可能多发现一些错误。

人们自然会想到，如果选择具有代表性的测试用例，用一个例子代表一类例子，会减少测试用例的个数，比用随机杂凑方式选择测试用例优越得多。

等价分类法是黑箱法设计测试用例的一种方法。它将输入数据的可能值分成若干个等价类。每一类的一个代表性的值在测试中的作用等价于这一类中的其它值。也就是说，如果某一类中的一个例子发现错误，则这一等价类中的其它例子也能发现同样的错误；反之，如果某一类中的一个例子没有发现错误，则这一类中的其它例子也不会发现错误，除非等价类中的某些例子又属于另一等价类，即几个等价类是相交的。

例如，对于程序 Triangle 来说，它的一个等价类可以是“大于零的三个等值数”(合理的等边三角形)。这样，如果该等价类中的一个例子 (比如，a＝b＝c＝6) 没有发现错误，那么该等价类的其它例子 (比如，a＝b＝c＝12 或 a＝b＝c＝234 等) 也不会发现错误。于是，就可以转移到其它等价类的测试。

用等价分类法设计测试用例的过程是：首先从程序的功能描述中找出一个个输入条件，并为每个输入条件划分一个或多个等价类，或用一个等价类表示一个或一组输入条件，然后选择测试用例，使每个测试用例要么代表尽可能多的合法等价类，要么只代表一个非法等价类，直至测试用例已代表所有的合法等价类和所有的非法等价类为止。合法等价类是指程序的合法输入数据。非法等价类是指程序的非法输入数据。一个程序不仅当输入数据合法时能正确运行，而且当输入数据非法时能够拒绝这些非法输入数据并给出提示信息。因此，我们要特别注意设计非法输入的测试用例。在设计包含非法等价类的测试用例时，每个例子只能代表一个非法等价类。这是因为，程序中的某些错误检测往往会抑制其它的错误检测。例如，对于 Triangle 程序，测试用例 0，−2，5 代表了两个非法等价类，则程序在发现“输入数据中含有零值”非法之后，可能不会再检查“输入数据中含有负数”是否合法，因此这一部分程序实际上没有测试到。

②边界值分析法

程序往往在处理边界情况时容易出错，因此检查边界情况的测试用例发现错误的概率较高。这里的边界情况是指等价类边界上的情况。

边界值分析法与等价分类法的主要区别在于：边界值分析法不是从一个等价类中任选一个例子作代表，而是有意识地精心选择一个或几个测试用例（例如，刚好等于、小于或大于边界值的数据），使得该等价类的边界情况成为测试的主要目标。

下面，以程序 Triangle 为例来说明边界值分析法与等价分类法的差别。程序的功能描述指出：三角形两边之和大于第三边。如果采用等价分类法，则至少可找出两个等价类：一类是满足这个条件的合法等价类，另一个是不满足这个条件的等价类，由此可设计两个例子：

a＝3，b＝4，c＝5

a＝1，b＝2，c＝4

如果程序中将表达式 a+b＞c 错误地写成 a+b＞＝c，则上述两个例子是无法发现这一错误的。若采用边界值分析法，选择例子

a＝1，b＝2，c＝3

则会使上述错误暴露出来。由此可见，等价分类法与边界值分析法的主要差别是：后者着重检查等价类边界上的情况。

③因果图法

上述两种方法没有检查各种输入条件的组合。因果图法着重检查输入条件的各种组合情况。因果图用于表示程序的逻辑流向，表示条件与相应动作之间的逻辑关系。

因果图法设计测试用例的过程是：

i. 从用自然语言书写的功能描述中找出一个模块的原因（输入条件）和效果（动作），并为每个原因和效果赋予一个标识符；

ii. 画出因果图；

iii. 把因果图转换成判定表（详见 8.2.1 节 4.(2)）；

iv. 把判定表的每一列各转换成一个测试用例。

④错误推测法

人们也可以通过经验或直觉推测程序中可能存在的各种错误，从而有针对性地编写检查这些错误的例子，这就是错误推测法。

错误推测法不是系统的方法，没有确定的步骤，在很大程度上是凭经验进行的。例如，输入数据为零或输出数据为零是容易发生错误的情况，因此可选择输入值为零的例子和使输出值为零的例子。又例如，输入表格为空或输入表格只有一行是较易出错的情况，因此可选择表示这些情况的例子。

(2) 白箱法

白箱法又称逻辑覆盖法。白盒法考虑测试用例对程序内部逻辑的覆盖程度。当然，最彻底的白箱法是覆盖程序中的每一条路

径，但由于程序中含有循环，路径的数目极大，要执行每一条路径是不可能的，所以，我们只希望覆盖的程度尽可能高些。目前常用的一些覆盖技术如下：

①语句覆盖

语句覆盖的含义是：选择足够的测试用例，使得程序中每个语句至少执行一次。

假定要测试下列一段 PASCAL 程序：

```
:
IF A>1 AND B=0 THEN X:=X/A;
IF A=2 OR X>1 THEN X:=X+1
:
```

它对应的程序流程图如图 8.6 所示。

如果设计的测试用例能够通过 ace 路径，就能保证程序中的每个语句至少执行一次，例如，A=2，B=0，X=3 就是一个这样的测试用例。

语句覆盖测试很不充分。在上例中，如果第一个条件语句中的 AND 错误地写成 OR，上述测试用例是不能发现这一错误的；如果第二个条件语句中 X>1 误写成 X>0，这个测试用例也不能暴露它；此外，沿着路径 abd 执行时，X 的值应该保持不变，如果这一方面有错误，上述测试数据也不能发现它们。

②判定覆盖

判定覆盖的含义是：选择足够的测试用例，使得程序中的每个判定至少都能获得一次“真”值和“假”值，从而使得程序中的每一个分支至少都通过一次。

在图 8.6 中，如果设计的测试用例能通过路径 ace 和 abd，或 acd 和 abe，就可以满足判定覆盖的要求。例如，下列两个测试用例通过路径 acd 和 abe：

A=3，B=0，X=1 (沿路径 ace 执行)

A=2，B=1，X=3 (沿路径 abe 执行)

判定覆盖比语句覆盖严格，这是因为，如果每个分支都执行过了，则每个语句也就都执行过了。但判定覆盖测试仍不充分，例如，上述测试用例不能检查出第二个条件中的 X>1 误写成 X<1 的错误，而且不能检查沿着路径 abd 执行时 X 的值是否保持不变。

③条件覆盖

条件覆盖的含义是：选择足够的测试用例，使得程序判定中的每个条件获得各种可能的结果。

图 8.6 中有四个条件：

A>1，B=0，A=2，X>1

为了达到条件覆盖的要求，需要有足够的测试用例，使得 a 点出现 A>1，A≤1，B=0，B≠0 等各种可能的结果，并在 b 点出现 A=2，A≠2，X>1，X≤1 等各种可能的结果。只需设计以下两个测试用例就可满足这一要求：

A=2，B=0，X=4（沿路径 ace 执行）

A=1，N=1，X=1（沿路径 abd 执行）

条件覆盖一般比判定覆盖测试充分，这是因为，条件覆盖使判定中的每个条件都取到了两个不同的结果，而判定覆盖则不保证这一点。但也有例外，例如，下面的两个测试用例满足条件覆盖，但不满足判定覆盖：

A=2，B=0，X=3

A=2，B=1，X=1

这是因为它们未能使程序中第一个判定的结果为“真”，也未能使第二个判定的结果为“假”，因而仅覆盖了路径 abe。

④判定／条件覆盖

判定／条件覆盖的含义是：选择足够的测试用例，使得程序判定中的每个条件取到各种可能的值，并使每个判定取到各种可

能的结果。

判定／条件覆盖解决了测试用例满足条件覆盖但不满足判定覆盖的例外情况。例如，

A＝1，B＝0，X＝3

A＝2，B＝1，X＝1

尽管满足条件覆盖，但不满足判定／条件覆盖的要求。

判定／条件覆盖似乎能测试所有条件的所有可能结果，但事实并非如此。我们知道，源程序只有转换为目标程序才能由计算机执行，而大多数计算机没有用单条指令实现多重条件判定的功能，必须将源程序中对多重条件的判定分解成几个基本判定。因此，更加彻底的测试应该检查每一个基本判定的所有可能的结果。

图 8.7 给出了图 8.6 源程序对应的目标程序，它把源程序中的多重条件判定分解成基本判定。

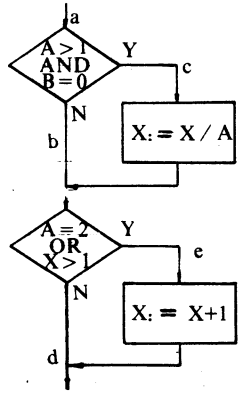


图 8.6　程序流程图

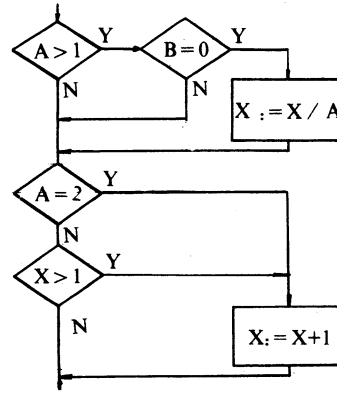


图 8.7　多重条件判定的分解

条件覆盖中的两个测试用例

A＝2，B＝0，X＝4

A＝1，B＝1，X＝1

能满足图 8.6 源程序的判定／条件覆盖的要求，但却不能使图

8.7 目标程序中的每一个基本判定取到各种可能的结果：它们不能使判定 B＝0 的结果为“假”，也不能使判定 X＞1 的结果为“真”。

原因何在？就是因为在含有 AND 或 OR 的逻辑表达式中，一个条件的结果抑制或阻碍了其它条件的测试。例如，在逻辑表达式 A＞1 AND B＝0 中，若条件 A＞1 为“假”，则目标程序就不再检查条件 B＝0 了，这样，B＝0 若写错了也发现不了；同样，在逻辑表达式 A＝2 OR A＞1 中，若条件 A＝2 为“真”，则目标程序就不再检查条件 A＞1 了，这样，A＞1 若写错了也发现不了。因此，利用判定／条件覆盖技术，不一定能测试出逻辑表达式中的全部错误。

⑤条件组合覆盖

条件组合覆盖的含义是：选择足够的测试用例，使得每个判定中的条件的各种可能组合都至少出现一次。

显然，满足条件组合覆盖的测试用例一定满足判定覆盖、条件覆盖和判定／条件覆盖。

对于图 8.6 的程序，必须使选择的测试用例覆盖下述八种条件组合：

A＞1，B＝0	A＝2，X＞1
A＞1，B≠0	A＝2，X＜1
A＜1，B＝0	A≠2，X＞1
A＜1，B≠0	A≠2，X＜1

要覆盖这八种条件组合，并一定需要设计八组测试数据，下面的四组数据就可以使上述八种条件组合至少出现一次：

A＝2，B＝0，X＝4

A＝2，B＝1，X＝1

A＝1，B＝0，X＝2

A＝1，B＝1，X＝1

上面的测试用例虽然满足条件组合覆盖的要求，但并没有覆

盖程序中的每一条路径，例如，路径 acd 就没有执行到。

在实际测试过程中，要把设计测试用例的各种方法结合起来使用。选取并测试一些数量有限的重要逻辑路径，并对一些重要数据结构的正确性进行完全的检查，这样不只证实软件系统接口的正确性，同时有选择地保证软件系统内部工作也是正确的。

2. 测试与调试步骤

测试与调试是同时进行的，边测试边调试。当测试发现程序的错误后，根据出错症状，使用调试工具去查找程序的出错位置和出错原因，并纠正和排除错误。程序语言一般都提供调试功能，例如，线路跟踪、赋值跟踪、设置断点等。在调试过程中，要充分利用这些调试工具。

测试与调试亦可采用模块化方法，先分模块进行测试与调试，当各模块都通过分调以后，再联起来进行测试和调试，通过联调后，便可试运行。试运行无误时即可投入正常使用。测试中发现错误后，除了需要通过调试查找并纠正错误外，还需要回到编写、设计、分析阶段作相应的修改，也就是说，需要进行再编写、再设计和再分析。测试的依据是设计说明书。

测试与调试的基本步骤如下:

(1) 模块测试

模块测试，亦称为单元测试，就是我们通常所说的“分调”。

模块测试以详细设计描述为指导，以模块为单元逐一测试。可同时对多个模块并行进行测试。

为模块测试设计测试用例大多以白箱法为主，一般可先用白箱法分析模块内部的逻辑，再用黑箱法补充一些例子。

模块测试主要对下列五个基本特性进行考察:

①模块接口;

②局部数据结构;

③重要的执行路径;

④出错处理能力;

⑤边界条件。

由于模块不是一个独立的程序，不能单独运行，既要靠调用它的上层模块，又要依赖于它所调用的若干个下层模块。因此，模块测试时要为每个模块测试设计一个驱动模块和若干个支持模块。驱动模块用于模拟调用被测模块的上层模块。支持模块用于模拟被测模块所调用的下层模块。

(2) 整体测试

整体测试，亦称为联合测试，就是我们通常所说的“联调”。

当测试完各个模块后，要把它们连接在一起进行整体测试。整体测试的任务主要是发现各模块的接口问题。例如，模块接口时，数据可能会丢失，一个模块可能会破坏另一个模块的功能，把子功能组合起来时可能不产生所要求的主功能，单个模块可以接受的误差在模块连接后可能会放大到不可接受的程度，全局数据结构可能会出问题，等等。

整体测试有以下两种测试方式:

①自顶向下测试

自顶向下测试就是从主模块开始测试，沿着层次结构向下移动，逐个把各个模块装配在一起。它不需要驱动模块，但需要设计支持模块。

自顶向下测试可分为以下五个步骤:

i. 用主模块作为测试驱动模块，它的直接下层模块用支持模块代替;

ii. 每次用一个实际模块替换一个支持模块;

iii. 每组合进一个模块，就进行相应的测试;

iv. 每完成一组测试后，用实际模块替换另一个支持模块;

v. 为保证不引入新的错误，可以进行回归测试，即重复以前进行过的部分或全部测试。

这一过程从第 ii 步开始连续进行，直到模块都组合起来为止。

自顶向下测试的主要优点是：与支持模块相联系的问题可能会被提前测试，而且不需要设计驱动模块；主要缺点是：需要支持模块，并且与支持模块有关的测试较为困难。

②自底向上测试

自底向上测试就是从最底一层模块开始测试，沿着层次结构向上移动，逐个把各个模块装配在一起。它不需要支持模块，但需要设计驱动模块。

自底向上测试可分为以下四个步骤：

i. 把低层模块组合成实现一个特定软件子功能的模块族(如图 8.8 所示)；

ii. 为每一模块族各设计一个驱动模块，用以控制和协调测试用例的输入和输出（图 8.8 中虚线框 D_1，D_2，D_3 分别代表模块族 1,2,3 的驱动模块)；

iii. 对模块族进行测试；

iv. 按模块结构图依次向上扩展，用实际模块替换驱动模块，将模块族与新的模块组合，再进行测试，直至全部测完。例如，在图 8.8 中，模块族 1 和 2 均不属于模块 M_a，去掉驱动模块 D_1 和 D_2，将这两上模块族直接与 M_a 接口，并测试这样新的模块族；同样，在模块族 3 与模块 M_b 连接前将驱动模块 D_3 去掉，测试之；……最后 M_a 和 M_b 将与 M_c 连接。

自底向上测试的主要优点是：设计测试用例较为容易，而且不需要支持模块；主要缺点是：只有在加入最后一个模块之后，程序才作为一个整体存在。

总之，自顶向上测试和自底向上测试各有优缺点，测试人员可根据程序的具体特点和测试工具的情况决定选择哪一种整体测试方式。最好将两种方式结合起来使用，对较上层模块采用自顶向下测试方式，对较下层模块使用自底向上测试方式。

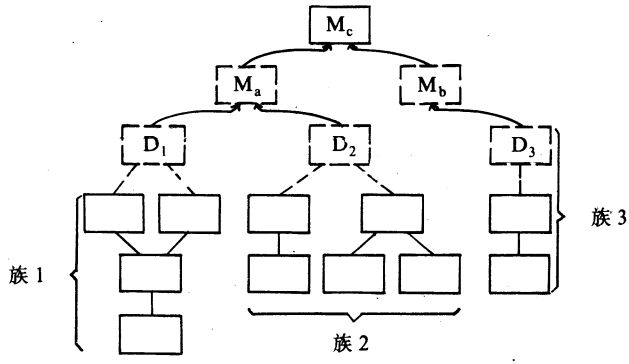


图 8.8　自底向上测试

(3) α 测试

在实际开始使用软件系统之前，把程序交给不参加该项程序设计的人，让他们独立测试，这叫做 α 测试。

(4) β 测试

当程序令人满意地通过测试后，可以让一些用户试用它，这就是 β 测试阶段。程序通过 β 测试就可以投入正常使用了。

3. 测试任务书

测试前，应制定测试计划，准备测试用例和测试用的程序(例如，驱动模块和支持模块)，测试时要认真做好记录，测试后要写出测试报告。所有这些资料均应长期保存下来，以便为将来的维护工作提供参考。

测试任务书将软件系统测试的全面计划以及对特定测试的描述加以文件化，是软件文档的一个重要部分。下面，给出测试任务书的一个参考格式:

(1) 测试范围

简要说明测试的目的、预期结果以及测试的全部步骤。

(2) 测试计划

给出测试任务的总体安排。

概括说明测试各主要阶段的次序、进度和方法。

列出测试进度，安排好各模块组的测试日程使之相互协调。

概括说明全部测试软件，包括驱动模块、支持模块及有关测试工具。

说明测试所需的计算机运行环境，包括内存要求、外部设备和终端等。

(3) 测试步骤

说明每个测试阶段的测试内容，包括模块功能、界面正确性、数据和文件访问、时间效率、设计约束规定等。

说明每个测试阶段将采用什么方法进行测试。

说明每个测试阶段需用的全部驱动模块、支持模块及其它测试软件。

写出每个测试阶段的测试数据及期望结果。

说明每个测试阶段的测试用例、输入方法、期望处理情况及输出格式。

(4) 实际测试结果

详细记录每个测试阶段的实际测试结果。实际测试结果要与预期结果相比较。

8.2.5 系统维护

经过测试的程序仍可能隐含着错误，甚至经过良好测试的程序也难免出毛病，用户的需求和程序的运行环境又可能发生变化，所以在程序运行过程中仍需要不断地对软件进行维护，继续排错、修改、扩充和完善。软件维护占软件系统全部开销的大部分。

1. 软件维护的内容

软件系统维护的内容大致可分为以下四类:

(1) 正确性维护

测试不可能发现软件系统中所有潜伏的错误。正确性维护就是要诊断和纠正软件系统在使用过程中出现的错误。

(2) 适应性维护

计算机不断更新换代，新的操作系统或操作系统的新版本层出不穷，外部设备及其它部件也要经常更新。另一方面，应用软件的使用寿命一般都超过原先开发这个软件时的系统环境的寿命。因此，为适应外部环境的改变而要对软件进行的修改，称为适应性维护。

(3) 完善性维护

当软件投入正常使用后，用户会提出增加新功能、修改已有功能以及一般性的改进要求和建议等。为了满足或部分满足这类要求，就要进行完善性维护。这类维护占软件维护工作的大部分。

(4) 预防性维护

为了进一步改进软件的易维护性和可靠性，或为进一步改进提供更好的基础而对软件进行的修改，称为预防性维护。比较而言，在软件维护中这类维护相对来说是很少的。

2. 软件维护文档

软件维护要有专门的文档，用来搜集用户反映、提出维护报告以及做维护的记录和评价。

所有关于软件维护的请求应该按一种标准的方式提出。软件开发者通常提供一个维护申请表，又称软件问题报告书，由要求维护的用户填写，内容包括：问题发现的时间，问题的描述，问题的性质、软件版本号，软件文档号，测试实例，问题分析等。对于正确性维护，必须对导致出现错误的条件作完整的描述，包括输入数据、错误情况、有关的源程序及其支持文档资料。对于适应性或完善性维护，只要提出个简明的要求维护的申请即可。

相应地，软件开发者根据来自用户的软件维护申请表，要制定一个软件修改报告，又称软件维护变动报告书，内容包括：维

护登记号，登记日期，报告人，要求修改的性质，修改所需要的工作量，修改的实际数据，新老版本标识，修改要求优先等级，修改是否测试，修改记录和维护评价，文档更新通知等。

8.2.6 原型化开发方法

结构化开发生存期法是一种预先定义的方法。它把软件开发的全过程划分为几个阶段，并预先规定每一阶段的目标和任务。然而，在很多情况下，很难对用户的要求建立一个严格、完备、一致和正确的说明。此时，采用原型化开发方法较为有效。

原型化开发方法对用户的要求不是预先定义的，而是在软件开发过程中逐步定义、补充和细化的。它借助于原型开发工具(例如，应用程序生成器、第四代语言)，能够快速地建立一种接近用户要求的原型，然后随着用户和软件开发人员对系统的理解加深而不断完善这一原型。原型化方法构造的原型是一种实际模型，比抽象模型容易理解，并允许用户提出更多的要求，把用户放在容易合作的位置上，便于开发出用户满意的软件。

原型化方法的基本步骤如下:

(1) 基本需求确定

确定系统要解决的主要问题的基本目的、目标、数据元素、记录联系以及要完成的功能。

(2) 开发工作模型

快速构造一个工作模型，实现基本需求中的一些关键性问题，然后再凭系统开发经验对用户的基本要求进行补充。重要的是要快速交付第一个模型，以提高用户的兴趣和信心。

(3) 表演、求精和补充

把模型演示给用户，广泛征求用户意见，观测、分析、评价原型的每一部分，充分说明它们的功能，促使用户尽多地提出修改和改进模型的意见。

(4) 原型完成

对原型不断修改和演示，直到原型所提供的功能为用户和软件开发者共同认可为止。

原型化方法要求原型构造过程要快，因此对原型开发工具的要求较高。随着软件技术的发展，出现了一批较为理想的原型开发工具，例如，屏幕生成器、报表生成器、超高级语言、非过程查询语言、集成数据词典、高适应性数据库管理系统、自动文档编排器、原型工作台等，从而保证了软件开发的高效率和高质量。

8.3 中文应用系统

8.3.1 中文管理信息系统

鉴于管理信息系统应用的广泛性，特别是中文管理信息系统目前已广泛应用于我国的各行各业，我们在 8.1 节中介绍了管理信息系统的定义、分类、层次、实现、信息结构、评价标准，以及它与电子数据处理系统、决策支持系统、专家系统的关系。

中文管理信息系统与西文管理信息系统的主要差别是：中文管理信息系统具有处理中文信息的能力。用于管理的中文信息主要有以下几种形式:

1，汉字文档，例如，文章、报告、公文、笔记、通报、函件等;

2，汉字表格，例如，帐本、帐单、发票、统计报表、合同等;

3，汉字统计图形，例如，附有汉字说明的条形图，点线图、圆饼图、高／低／收／开图等 (见 7.3 节 3.)。

为了实现中文管理信息系统的中文信息处理功能，除了必要的汉字设备外，还需要开发或选购中文应用软件。

如果决定选购软件，那么，当系统分析和总体设计完成后，就可以选购软件了。软件的选择范围通常是面向对象的程序语言或数据库管理系统，字处理软件、数据表软件、统计图形软件、组合软件、软件族等通用应用软件，工资管理系统、财务管理系统、人事管理系统等专用应用软件。选择的软件必须符合系统说明书和设计说明书的功能描述。如果选择的软件不是中文软件，还需要对软件进行二次开发——软件汉化，用以增加中文信息处理功能。然而，由于这些软件不是为解决具体单位的具体问题专门设计的，因此很难完全符合用户的要求。如果没有适合的软件可供选购，那么只能进行软件开发了。

如果决定开发软件，那么必须经历软件系统开发的各个阶段：系统分析、系统设计、编写程序、系统测试，还要进行系统维护。中文应用软件一般是用中文程序语言来编写的。由于程序语言的功能和特性对于程序质量有着直接影响，因此，选择适合的程序语言是应用软件开发的一个重要问题。选择中文程序语言除了考虑应用领域、算法和计算复杂性、系统的执行环境、性能、数据结构的复杂性等因素外，还要考虑它的汉字处理功能对应用软件的传递性。也就是说，中文应用软件依赖中文程序语言的汉字处理功能，因此，在选择中文程序语言时，要充分考虑中文程序语言的汉字设备硬件环境、中西文兼容性、汉字输入输出的灵活性、效率性和多样性、中国人的习惯、用户的汉字界面、汉字处理功能的可扩充性和经济性等。当然，也可以使用西文程序语言来编写应用软件，但这就需要在应用软件一层上重新建立汉字处理功能，使中文应用软件开发的工作量太大，因此，一般很少采用这种方法来开发中文应用软件。

8.3.2 中文桌上排版系统

用计算机来实现编辑、排版、印刷自动化是出版业发展的必然趋势。出版自动化系统可分为两大类：专业性的电子排版系统

和非专业性的桌上排版系统（Desk Top Publishing System，简称 DTP）。

专业性的电子排版系统早在 1965 年就出现了。它是由计算机控制的自动排版系统。编辑人员将排版内容和组版格式输入计算机，计算机通过专用排版语言自动安排版面，然后用激光印刷机或照排机自动制成版面，供印刷使用。由于整套系统过于昂贵，应用范围仅限于一些大型印刷厂、出版社或报社等专业领域。

从八十年代起，随着微型机及其输入输出设备的迅速发展，非专业性的桌上排版系统呈爆炸性的发展。这是微型机、图形图象处理技术、高密度高速度印刷输出设备结合的产物。桌上排版系统是一种普及的物美价廉的计算机排版系统。它使出版从专业出版界中解放出来，非专业人员也可轻易地制造出具有相当水准的出版品，成为办公自动化的有力工具。尤其是，近年来中文桌上排版系统的发展，更是引人注目。

本节主要介绍桌上排版系统的硬件需求和软件功能，幕后排版与幕前排版的基本概念，以及中文桌上排版系统的选择标准。

1. 桌上排版系统的硬件需求

桌上排版系统的硬件需求如下:

(1) 微型计算机主机

这是桌上排版系统的核心，它控制外部设备的输入输出，并对数据作必要的加工处理。目前广为采用的主要有 IBM PC 及其兼容机和 Apple 的 Macintosh。

IBM PC 及其兼容机较适合于文字处理，但处理图象和图形则较为麻烦。由于它们的市场占有率较高，因而开发者仍在其上开发桌上排版系统，并尽力开发尽可能好的软件支持环境，例如 MS-Windows，以弥补硬件支持功能上的缺陷。

Apple 的 Macintosh 本身就是一种绘图计算机，具有很强的绘图能力，因此在处理图形和图象时，不但可以提供较完善的功

能支持，而且速度上也很快，适用于桌上排版系统。但它在市场占有率方面逊于 IBM PC 及其兼容机。

(2) 高分辨率显示器

桌上排版系统强调的一个重要观念是 WYSIWYG。也就是说，印出的实际结果在操作过程中可预先在屏幕上看到，只有这样，才能确保输出文件的质量符合设计者的要求。为此，显示器的分辨率必须大到足以能显示出近似于印出文件的效果，而且显示的结果必须尽量降低横向与纵向比例上的失真。如果屏幕的大小能配合纸张大小及摆置方向，那么就更有助于达到 WYSIWYG 的要求。此外，要使显示器发挥它的功能，还必须要有与它相匹配的显示适配器配合使用，例如，EGA，VGA，8514A 及其它特殊的显示卡。

(3) 高容量外存储器和扩展内存

对于文字和图形，都可用个体 (Object) 的观念来处理和存储，所需要的内存和外存均很有限，但对于图象，以现有的技术只能以位映象 (Bit Map) 的方式处理和存储，因而对内存和外存的需求量相当大。以一张信纸大小的面积来计算，光是黑白图象的点密度在 300dpi (每英寸的点数) 时，就需要 1MB 左右的存储容量。若是具有 256 种灰度 (Gray Scale) 或色彩的图象，在 300dpi 的点密度下，则需要 8MB 以上的容量才够存储。更何况一份文件往往不止一页，而且在文件处理过程中也常常需要一些额外的内存，以作为缓冲存储区，因此对于主机内存的需求至少要 1MB 以上，最好能再加上 2–4MB 以上的扩展内存。至于外存储器方面，通常需要 20MB 以上的硬盘。除此之外，若能有网络的文件服务器作为外存储器，则可以与网络中的其它用户共享数据资源或已处理完成的文件。若目前已渐趋成熟的光盘能在存取时间与技术上更求突破，则也许会是另一种容量更大的有效外存储器。

(4) 鼠标器及其它指示装置

桌上排版系统强调的另一个重点是用户界面的友好性，使用户操作简易。尤其是在进行版面设计时，更需要操作方便的输入装置，以便轻易地指定文字、图形和图象的摆设位置与延伸范围。另外，也可以方便地辅助用户选定要操作的功能项目，或决定数据的处理方式等等。目前，鼠标器是被广泛接受的指示装置，其它指示装置有光笔、数位板等。

(5) 图象扫描器

图象扫描器是目前最主要的图象输入工具。若再配合日趋成熟的光学字符识别（OCR: Optical Character Recognition）技术，则可作为文字输入的辅助设备。若配合图象－向量(Raster-to-Vector) 处理技术，则可作为图形的输入工具。由图象扫描器输入的图象经过一些基本的处理后，便可与文字和图形一起进行排版。

目前使用最广泛的图象扫描器的分辨率为 300dpi。300dpi 以上的产品，比如，400dpi，480dpi 或 600dpi 的图象扫描器已有产品问世，但价格较贵，故使用不够广泛。

除了单色调（Bi-level）的图象扫描器外，目前还有很多可提供灰度或彩色扫描能力的图象扫描器，用以扫描相片或图片等，使所摄取的图象逼真度提高。

由于在使用图象时通常以小面积居多，很少需要用到象一般图象扫描器所提供的 A4 / Letter Size 或 B4 大小的图象，因此小型而价廉的四英寸掌上型图象扫描器（Handy Scanner）正逐渐风行。

(6) 向量-图象文字处理装置

虽然在一份文件中可能包含有图象、图形和文字，但一般仍以文字为其主体。而文字表现在文件上时，除了有不同字体外，还有大小、摆置方向、中空、加重等许许多多的问题待处理。

传统上对字形的处理，均采用点阵方式，这是因为它较适合于现有显示或印刷设备的操作方式。但采用点阵方式的缺点是：

若要达到美化的要求时，系统必须备有大小不同、各种各样的点阵字形库，这需要占较大的存储量。然而，桌上排版系统对各种不同字形的需求又特别强烈，因此最好能以向量方式存储各种字体，而在必要时，再通过向量-图象文字处理装置，转换成各种不同效果与大小的点阵字形，供印刷和显示用。

(7) 文件输出设备

产生高质量的文件是桌上排版系统的主要目标。而要达到这一目标，需要相当质量的输出设备与系统相互配合，以使由主机排版并处理定稿的文件得以完全而忠实地展现在纸上或其它媒介物上。

目前，300dpi 的激光印刷机的输出质量虽难与专业印刷相比拟，但对于一般桌上排版系统的用户而言，已经足够了。为了提高印刷质量，400 或 600dpi 的激光印刷机相继推出。大多数激光印刷机都倾向于提供更易于处理图形、图象与灰度功能的整页描述语言（Page Description Language），以便减少系统在处理输出时的负担，并提高输出文件的质量。

除了激光印刷机外，其它的输出设备，例如，彩色喷墨式印刷机、绘图仪等，也可作为文件输出设备，但只在某些特定场合下才会用得上。

(8) 网络及通信设备

网络的最大目的在于实现资源共享，这一点对于桌上排版系统也不例外。在计算机网络中，桌上排版系统可共享外部设备，例如，图象扫描器，文件输出设备或文件服务器等，同时在文件制作过程中常需取用的图象、图形或文件，亦可借助网络互通有无，各取所需。甚至对于一份较复杂的文件，也可分散至网络中的各个文件处理工作站，先进行预处理工作，然后再借助网络集中至某一特定工作站进行最后的组版和输出工作，以缩短文件的制作过程，提高制作效率。

除了网络之外，其它的通信设备，例如，传真系统等，虽非

一般桌上排版系统所必备，但对于文件的传递，却有很大的帮助。

2. 桌上排版系统的软件功能

桌上排版系统的排版软件大致可分为三个部分：预处理软件、组版软件和系统支持软件。预处理软件主要包括文字处理软件，图形编辑软件、图象编辑软件和表格处理软件，对输入的文字、图形、图象、表格等预先做编辑处理，交由组版软件依据版面的编排，组合成一页一页的版面，再通过激光印刷机等输出设备印出。从软件的观点看，桌上排版系统是由许多软件组成的系统。这些软件之间都有一些共同的工作需要处理，例如，用户界面，文件存取、显示与印刷等等。为此，一方面，必须要有一个良好的公用操作环境，提供与设备无关的输入输出界面，对文字和图形提供必要的支持，例如，Macintosh 本身就具有一个很好的操作环境，而 IBM PC 则有 DOS 的 MS-Windows 或 OS／2 的 Presentation Manager；另一方面，必须要有公用的系统支持软件，对预处理软件和组版软件提供必要的支持，例如，文件管理软件、字形处理软件、通信软件、网络支持软件、文件转换软件等等。下面，逐一介绍组版软件、各个预处理软件和各个系统支持软件的软件功能。

(1) 组版软件

组版软件是桌上排版系统的核心软件。组版软件的主要功能是版面设计。版面编排设计功能包括：纸张大小的设定，纸张摆置方向的设定，文字编排方向（横排或竖排）的设定，版边留空白，文字栏位分割，页码设定，文字、图形、图象框的设定，单页处理和整册处理等等。

除版面设计功能外，组版软件还具有文字、图形和图象处理功能。当组版软件将经过预处理的文字、图形和图象组合成版面时，由于通常在尺寸上、形状上、完整性上不一定完全配合，因此，组版软件应具有一定的文字、图形、图象编辑和修改能力，

使用户在组版时仍可做适当的修改，以符合版面的整体要求。

文字处理功能包括以下几个方面:

①多栏组版

设定栏位方向和大小。一般说来，一个版面上若分成若干个栏，则其栏宽应是相等的，但也有时为了版面的活泼而有不同的考虑，因此栏与栏之间的距离、各栏的宽度可由用户自行设定。

②文字齐栏方式

靠左（上）对齐、靠右（下）对齐、靠中对齐（居中）、靠两边对齐等。

③各段文字的缩排方式

段首内缩、行首内缩、行尾内缩、段首外张等。

④文字的间距

字间距、行间距、段与段之间的距离等。

⑤章节号码的编制及跳行跳页

⑥目录和索引的编制

⑦页码和脚注的编制

⑧字体、字形大小的设定

⑨文字本身的编辑和修改

图形和图象处理功能包括以下几个方面:

①输入已编辑完成的图形、图象和表格

组版软件应能接受由各种不同预处理软件产生的各种不同格式的图形、图象和表格，至少具备解释一些标准格式的能力。

②图形和图象的编辑和修改

图形和图象的摆置、放大、缩小、旋转、裁剪、搬移等。

③基本图形的输入

在组版软件本身的绘图功能方面，除了考虑图素（例如，圆、椭圆、直线、曲线等）外，还应考虑图形属性（例如，线的种类：实线、各类虚线、文武线、花边，粗细，内填图样、背景等)。

(2) 文字处理软件

文字处理软件为组版软件提供组版所需要的文字。它必须具备完善的文字编辑功能，除一般字处理软件所具有的插入、删除、查找、修改、搬移、复制等功能外，还可由用户自行设定某段文字的属性，例如，字体、字形大小、笔画粗细、反白、加底线、文字变形、上标、下标、字间距、行间距、段间距、段首内缩或外张等。

(3) 图形编辑软件

图形编辑软件为组版软件提供组版所需要的图形。它必须能使用户方便地绘制一些基本图形，例如，直线、矩形、圆角矩形、椭圆、圆、多边形、弧、弦、扇形、曲线、区域填涂等，而且构成各种图形的线段粗细也可任意改变。每个基本图形均可视为单一个体，多个单一个体亦可组合成一个大个体。随时可对个体作放大、缩小、旋转、搬移、复制、修改、删除、变换等操作。个体与个体之间也有上下层关系，不透明的上层个体会遮盖住下层个体，透明的上层个体会使下层个体浮现。

除了一般图形外，还应包含一些商用统计图形和工程技术图形。

(4) 图象编辑软件

图象编辑软件为组版软件提供组版所需要的图象。它通过图象扫描器输入所需要的图象。图象编辑功能包括对图象的剪辑、搬移、复制、放大、缩小、旋转等，还包括图象的合并及细部修改等。

除图象编辑功能外，一般的图象编辑软件还提供了基本的绘图功能和文字输入功能，以便用户在图象上加入图形和文字。

由于单色调的图象存在失真、图象不易处理等缺陷，因此需要能处理灰度图象的编辑软件。除了上述的基本图象编辑功能外，它还可以对图象进行灰度调整、渐层处理、杂点去除、边线加强、明显化 (Sharpening)、柔和化 (Softening)、平均化

(Equalization) 等等，使图象所能显现的效果更为生动、逼真和丰富。

(5) 表格处理软件

表格处理软件为组版软件提供组版所需要的表格。由于表格的表现效果比一般文字好，因此表格在文件中经常出现。然而，在一个表格中，除了文字外，还包含图形甚至图象，所以有关表格的处理、编制、文字位置的安排，是一般文字处理软件和图形编辑软件所不能胜任的，需有专门的表格处理软件，使用户可以方便地设计表格。

表格处理软件需提供框线绘制、文字定位。栏位定义、文字输入与查核，以及框的移动、放大、缩小等基本功能。此外，它还需要提供一些基本的绘图能力和图象输入功能，以增强表格的表现效果。

(6) 文件管理软件

文件管理软件是文件存取的管理工具，功能类似数据库管理系统。它的主要功能是使用户可以很方便地对数量庞大的图象文件和完稿的版面文件做各种存取的管理工作，例如，文件存入、删除、增页、删页、查询、取出、印出等。它亦可为预处理软件和组版软件快速存取文件提供支持。

(7) 字形处理软件

字形处理软件的主要功能是支持预处理软件和组版软件所需要的繁多的字形制作。通常字形的制作过程包括字形原稿的数字化、修补、增删、存储等。其中，字形的存储方式又可分为点阵式、向量式、笔画式等（见 10.4.2 节），各种方式各有其优缺点。一般在将字形原稿输入计算机数字化时，通常都必须借助图象扫描器或其它图象输入工具。

(8) 桌上型通信软件

桌上型通信软件的主要功能在于为文件传送与接收提供支持，其中包括文件传真的支持、文件传送、远程查询、终端模拟

等。

(9) 网络支持软件

网络支持软件为预处理软件和组版软件提供网络通信能力，以便各个软件得以通过网络及其它系统共用软件资源和硬件资源，例如，图象扫描器、文件输出设备、传真系统及各种文件。另外，可使各系统预处理软件并行分担预处理工作，加速文件制作速度。

(10) 文件转换软件

通常文件组版所需要的文字和图形，除了可由预处理软件输入和产生外，还可由其它的通用软件（例如，dBASE，Wordstar，PE，Lotus 1-2-3，AUTOCAD 等）产生文字和图形，经过文件转换软件，将其格式转换成预处理软件和组版软件所能接受的格式，然后直接交由组版软件组合成版面，或者由预处理软件编辑整理后再进行组版。这样，可以扩大文字和图象文件的来源，减轻预处理软件的负担，加速文件的制作。

3. 幕后排版与幕前排版

目前的桌上排版系统可分为两大类：一类是幕后排版系统，另一类是幕前排版系统。

幕后排版系统是一类命令驱动式排版系统。系统定义了一套排版语言，用户依据该语言的语法，可在要输出的文本文件中插入编辑命令或排版命令，用于说明版面的格局以及排版时所需用到的文本文件、图形文件、图象文件、表格文件以及字体、字形等等，系统根据用户的命令，以批处理的方式组织版面，获得要编排的结果，并从输出设备印出。这类排版系统通常亦提供预演功能，可以预先在屏幕上观看编排结果，如果用户对编排有何不满意的地方，需要回到预处理软件去更改。

幕后排版软件设计容易，不需要高分辨率的显示器，排版速度也比较快。但是，它不能看到排版的结果，修改时必须回到预处理软件；而且只能编排简单的文件，只适用于版面变化不大的

文件，例如，书籍、技术手册等；而复杂一点的版面很难用编辑命令或排版命令表达清楚，例如，一个流程图，图形与文字的相对位置，要用编辑命令或排版命令指明，即使提供了命令，也会令人望而生畏。

幕前排版系统是一类会话式排版系统。系统将有关的编辑命令或排版命令都以多层次的菜单形式显示在屏幕上，用户可利用鼠标器或键盘，在屏幕上直接编排文字或段落、移动它们的位置、改变文字的字体和大小、设定文字的横排或竖排、设定字间距和行间距、设定纸张大小、分裂栏框，并可直接在屏幕上绘图，把文字、图形，图形、表格直接组合到版面上。所编排的结果亦可立即从屏幕上显示出来，若不满意，则可随时即刻修改。

幕前排版可在屏幕上立即见到编排的结果，达到了WYSIWYS的要求，提供了友好的用户界面，符合人体工学。它可以编排版面较复杂的文体，例如，广告、杂志、说明书等，并可一边编排一边思考版面的设计。由于编辑命令或排版命令均以菜单的形式显示在屏幕上，一目了然，操作简单，容易为用户接受，尤其适合非计算机操作人员使用。但是，软件设计较困难，需要较长的开发时间；需要高分辨率的显示器和鼠标器等相关设备，费用较高。

4. 中文桌上排版系统的选择标准

近年来，中文桌上排版系统层出不穷。那么，如何选择一套适合自己使用的中文桌上排版系统呢？下面，提出一些选择标准，供大家参考。

(1) 排版功能

排版功能是中文桌上排版系统的主要功能，因此，排版功能的完整性是对中文桌上排版系统评价的主要标准。尽管用户对排版功能的要求不同，但排版功能至少要具有较强的版面设计功能和文字、图表、图象、表格处理功能，以达到图文并茂、版面美观的效果。

(2) 出版质量

出版质量是评价桌上排版系统的一个重要标准。影响出版质量最直接的因素是输出设备，而输出设备的输出质量又受其输出方式和点密度的影响。所谓输出方式，就印刷机而言，常见的桌上型印刷机有点阵式、菊轮式、热感式、喷墨式、LED、LCD、激光等印刷机。每种输出方式的输出质量都不同。目前，桌上排版系统使用最普遍的是激光印刷机。仅就激光印刷机而言，还有许多决定其输出质量的因素，例如，点密度、输出点的形状和大小、碳粉颗粒的大小和特性、写黑或写白等等。

除了输出设备外，排版软件也是影响输出质量的重要因素。有些排版软件为了增加排版速度，简化程序的复杂度，不得不牺牲质量。例如，字间距和行间距的调适，对版面的协调有很大的影响，然而，许多排版软件在做两边对齐的齐栏方式时并没有对字间距做最佳的调适；更严重的是，有些排软件在排英文单词时，未做断字（Hyphenation）、调和字（Proportional Spacing）、叠置（Kerning）等处理，也未对版面禁则，例如，行首禁则、行尾禁则、分离禁则、弧文（Orphan）、寡字（Widow）等做适当处理；许多排版软件在做文字绕图时，在图文的交界处没有做很好的处理，可能间隔太宽或太窄，有些甚至会在交界处产生异于它处的行间距，非常显眼，对版面的美观影响很大。

对中文排版软件来说，汉字与英文单词之间的协调也是影响版面美观的重要因素。这些因素包括汉字与英文字形大小的配合，汉字间、英文单词间、汉字与英文单词间的配合。上标和下标的字体、字形大小及其上升或下降的高度也是常被忽略的地方。另外，加底线或顶线时，线的粗细及其与文字的距离，以及在编排脚注时，脚注本身的格式及脚注与内文之间版面的处理也是文字组版时容易疏忽之处。

不同的排版软件在图形绘制方面也有所不同。以圆或椭圆为

例，有些画得非常精确而且平滑，有些则非常粗糙。对于直线或图形外框的处理，也会有许多差异，例如，斜线的端点，尤其是线较粗时，有些画得很平整，有些则显得歪斜；图形外框若为虚线，在交角处，如矩形的四角、圆角矩形的圆弧与直线交接处、椭圆的四分法交接处等，有些处理得非常协调，有些则非常粗糙。图形加背景或内填图样时，有些能充分与外框宽度配合，有些则不能。

(3) 字形

字形的美观是制作高质量出版品的基本因素，是评价桌上排版系统重要标准。

在文章中，各种不同地位或性质的文字（例如，章节标题、正文、眉注等）往往需用不同的字形来表现，例如，标题字一般都比正文字大，用黑体字表示重点内容，以突出其地位，增加易读性。字体和字形大小体现了字形的变化（见 3.2.1 节）。

为了保证印刷质量，中文桌上排版系统至少要提供 128×128 点阵的汉字字模，最好能提供 256×256 点阵的汉字字模。

在桌上排版系统中，通常依字形的输出目标把字形分为屏幕显示字形和印刷机印刷字形。在 WYSIWYG 的要求下，这两种字必须充分配合。所谓充分配合，并不是使用同一套字形，由于屏幕显示的分辨率与印刷机不同，若要看到与印刷大小相同的结果，则必须配合一适当大小的屏幕显示字形，而且该屏幕显示字形与对应的印刷机印刷字形在外形上必须类似，能够将字形的特性表现出来。

(4) 用户界面

无论是什么系统，用户界面都是影响操作效率和学习时间的直接因素，因此在选择桌上排版系统时，对用户界面应仔细评估。

对于会话式幕前排版系统来说，用户必须不断地与系统进行对话，因此用户界面显得尤为重要。若将用户界面分为输入界面

和输出界面，则会话式幕前排版系统在这两方面都很重要，而批处理式幕后排版系统则着重于输出方面。

输入界面是指下达命令和输入文字、图形、图象和表格的方式，因此除了输入设备外，很可能还需要输出设备的配合。例如，屏幕显示菜单，对话框等。输入界面在系统的易学性上，必须考虑菜单的规划是否将各功能项做了适当的分类，所有功能的操作方式是否一致。例如，物件导向的用户界面都是先选目标物，再对其下指令；若是系统中有些功能必须先选命令，再指定目标物，则很容易造成用户学习上的困扰。在易用性考虑上，必须对输入设备及其操作方式深入了解。一般来说，桌上排版系统的主要输入工具是键盘和鼠标器。鼠标器必须配合屏幕的显示和鼠标器的指标，用户必须看着屏幕，并操作鼠标器，反而会造成熟练的用户提高效率的障碍，因此一般的系统会提供加速键，使用户可以用键盘做所有的输入工作。命令的下达既可以用鼠标器又可以用键盘，但是应该是每个命令都可以用任一种方法下达，若是有些命令是用鼠标器，而另一些命令要用键盘，则不但会影响用户的学习，而且对操作效率更是一大障碍。

输出界面是指屏幕上显示的结果和印刷机上印刷的结果。屏幕显示最重要的是要 WYSIWYG，也就是说，要在印刷机上印出什么，屏幕上就要显示什么。但是，由于屏幕大小和分辨率都无法与印刷机完全相同，因此百分之百的 WYSIWYG 是不可能的，一般的软件只能使屏幕上的显示尽量象印刷机上所出的结果。所谓象，就是将印出结果上的各项信息尽量确切地表现出来。信息表现得越正确，就越象；信息表现得越多，就越象。就信息的正确性而言，各物件的大小及相关位置是必须注意的重点。例如，字形大小、字间距、行间距等的大小比例及文字之间的相关位置必须符合输出结果。就信息的丰富性而言、字体、笔画粗细、斜体、反白、加底线、上下标等都是表现的重点。由于屏幕显示的分辨率与印刷机不同，而且屏幕大小也受到限制，因

此无法用单一显示尺寸在屏幕上显示足够的信息。一般说来，屏幕显示必须提供多种大小的显示，并配合屏幕滚动功能，才能弥补屏幕分辨率和大小的限制，使用户在使用时有足够的灵活性和方便性。

(5) 印刷机的结构

目前桌上排版系统最常用的输出设备是激光印刷机，而以激光印刷机的结构来看，可将其分为控制器和印刷机构两部分。控制器负责将输出指令点阵化，并控制印刷机构将点阵化后的全页信息印出。控制器与微型机主机之间可用两种不同的方式连接。一种是使用 RS-232 或 Centronics 界面，另一种是将控制器做成控制卡，插在扩展槽，直接使用主机的总线。

第一种方式的优点是控制器和印刷机构合为一体，直接使用输出端口，因此一台印刷机可供多部主机共用。但是这种方式在传输信息时，其传输速率受到传输界面的限制，很可能在印刷大量文件时（尤其是图象）形成瓶颈。就中文桌上排版系统而言，目前还没有任何控制器可提供丰富的字形产生能力，因此汉字字形必须以图形形式传给控制器，在这种情况下，用户应考虑第一种方式的输出速度是否符合要求。

第二种方式的优点是控制器直接使用主机的总线，不需做信息传输，因此输出速度快，但是因为其控制器插在主机的扩展槽上，所以一台印刷机无法供多部主机共用。

在英文桌上排版系统中，全页描述语言是非常重要的技术，因为它提供了很好的绘图功能和字形功能，而且与输出设备的点密度无关，所以可扩充性相当高。更重要的是，它提供了一标准界面，使输出界面的无关性大大提高。因此，如果中文桌上排版系统拥有中文的全页描述语言，将使中文桌上排版迈入一个新的境界。

(6) 可维护性

无论是什么系统，系统维护都是使系统正常运行的最重要的

支柱，因此用户在选择桌上排版系统时，应对其可维护性予以慎重的评估。

一个系统的维护工作包括硬件维护和软件维护。硬件维护包括硬件的维修和新设备的更新。软件维护包括错误的修改和功能的增强。功能的增强是指软件本身功能的扩充、处理速度的加快、提供新的驱动程序给新的设备、提供新的字形、提供配合的软件或公用程序等等。

整个维护工作不但影响系统的正常工作，而且对整个系统的生存期影响也很大。因此，用户在选择中文桌上排版系统时，应注意系统开发者的技术实力，是否有较强的维护能力。

(7) 可扩充性

可扩充性是指硬件的扩充能力。就桌上排版系统而言，用户可能会考虑能否换用密度更高质量更好的印刷机，能否换用屏幕更大分辨率更高的显示器，是否可增加外存储器的容量，是否可增加内存容量以提高处理速度等等。

当然，影响可扩充性的最直接因素是硬件本身的扩充能力，但是，仍有许多其它因素值得考虑。例如，系统的标准化程度，如果系统的每项界面都采用标准化，则各种标准化的设备均可适用。又例如，系统对输出设备的依赖程度，如果系统可以不管输出设备的点密度和输出方式，均可正常输出，则对于新的输出设备均可适应。当然，这与系统设计时各个逻辑层次均有关，如果在输出之前，以高度的逻辑特性来描述输出对象，例如，描述输出对象的位置时采用尺寸而不采用点数，则对各种输出设备的可保持高度的适应性。

(8) 成本

使用一套系统，除了购买系统时的软硬件费用外，还必须考虑许多其它投资，例如，操作人员的教育培训费用、系统维护费用、系统操作费用等，都必须做好评估。

影响操作人员的教育培训费用最主要的因素是系统的易学

性。当然，应用领域的专业知识也是需要灌输的知识，例如，桌上排版系统在进行版面设计时，需具备美工知识，但这基本上是与系统无关的。在选择桌上排版系统时，要考虑用户界面是否符合操作人员的习惯，是否有更简便的操作方式。例如，若组版软件提供了版面的样板，则操作人员只要会针对其文件的版面特性选择适当的样板，即可轻易地编制精美文件，而无需太多训练。另外，用户手册也是影响该项费用的重要因素，如果没有完备的使用手册，则用户不但无法系统地获取正确的观念，而且还得多方尝试错误，对学习时间影响甚大。

系统维护费用包括硬件的维护、扩充，软件版本的更新以及新软件的购置等，不但要评估其趋势和可行性，还得考虑其成本。

系统操作费用泛指系统运行时所需要的一切费用，例如，电费、操作人员工资、耗材费用等。若要降低操作人员的成本，必须提高工作效率。为此，除了考虑系统的效率外，操作方式更是重要的因素，例如，汉字输入方式。耗材费用是指纸张、碳粉等费用，降低此项费用的方法是减少输出时的出错率，甚至在印刷机的选购时，碳粉匣的使用效率也是考虑的因素。

8.3.3 英汉机器翻译系统

机器翻译系统是利用计算机模拟人工翻译过程把一种自然语言自动翻译为另一种自然语言的自然语言处理系统。

机器翻译是人工智能的一个重要课题，是利用计算机处理自然语言的一个重要应用领域。它是一门吸取多门学科（计算机科学、软件工程、认知科学、语言学、信息论、控制论和系统论等）的研究成果的综合学科。

机器翻译可分为文字机器翻译和语言机器翻译，也可分为一对一、一对多或多对多语言的机器翻译。

世界上至今尚未有一个机器翻译系统能把一种自然语言的科

学文本全自动、完美、准确地翻译成另一种自然语言文本。这有赖于对自然语言理解的研究取得较大突破后才能解决。尽管如此，机器翻译系统仍有很大的实用价值。

我国对机器翻译的研究早在五十年代已经开始，经历了坎坷曲折的历程。近年来，我国的机器翻译研究工作日趋实用化，诞生了一些正确率较高的机器翻译系统，并可在小型机或高档微型机上运行。随着机器翻译研究的深入发展，更高质量的机器翻译系统可望早日实现。

下面，以"译星"英汉机器翻译系统为例，来说明机器翻译的信息处理过程和机器翻译系统的结构。

"译星"是一种双语单向的机器翻译系统，它只进行英文到中文的单向翻译。它的机器翻译原理与人工翻译原理类似，如图8.9所示。

1. 机器翻译的信息处理过程

"译星"英汉机器翻译系统的翻译过程是：输入英语原文句子，经查词典后得到句中各词相应在词典中给出的初始信息，再通过规则库相应规则的作用，把英语原文链改造为多叉树结构；在此基础上，根据生成规则转换为汉语译文链的译文表层结构，最后按译文链输出汉语译文，从而实现英文到中文间的翻译。

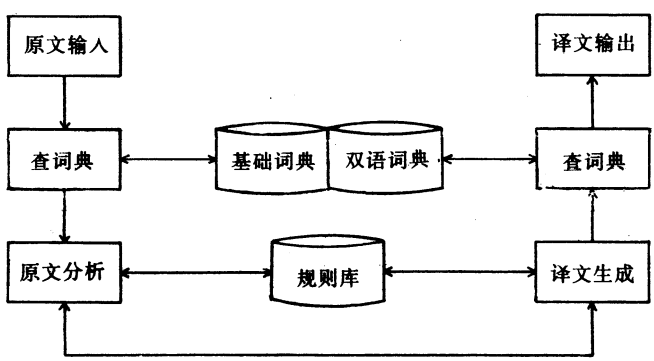


图 8.9 机器翻译原理示意图

(1) 原文输入

用户向系统输入要翻译的英文原文。输入方式有以下两种:

①原文以文本方式输入。文本文件可用系统提供的编辑程序建立，亦可用外部字处理软件或编辑程序来建立，例如，Wordstar，PE等。

②系统可以连接光电扫描器，识别多种标准的印刷体（识别率在90%以上），从而简化了输入工作，可迅速地输入大量英文资料，提高了工作效率。

(2) 词典查找

系统把输入的英文原文切分为单词、标点、符号、句子、段落。经过查词典向各词赋予初始信息。由于系统词典采用的是原形存词法，因此，对于未查到的字符中要进行标点、词尾、前缀、后缀等的分离。经过反复分离、查找，对查到的原形词给出相应的信息，对于生词（包括拼写错误的）显示在屏幕上，提请用户修改。系统将词典里存储的词汇信息调入句子场，供原文分析用。

(3) 原文分析

逻辑语义是原文分析的主要目标。与任何一种语义关系一样，逻辑语义也必然寓于一定的语法结构之中。原文分析的基本思想是：原文中单词的语义因子和语法特性进入一定的语境之后，根据语言的特定的语法规则，产生了多种语法、语义关系，从而达到信息传输的效果。因此，原文分析就是在特定语境中分析出给定的各词的语法、语义和语用关系。

(4) 译文生成

通常在汉语中，以中心词为核心，其它依附成分是按逻辑语义有序地排列在中心词的前后两侧。也就是说，可以归纳出一条汉语逻辑语义，按照这条既定语义链，来确定语义单位在句子中的相对位置。这是译文生成的基本思想。通过递归地调用这条规律，用逐次膨胀法逐步地完成译文的生成。

(5) 译后编辑

由于自然语言是一个无穷集，而机器翻译所依赖的知识库是有限集，不能完全覆盖自然语言这个无穷集，因此，目前的机器翻译还不能达到人工翻译的水平，翻译后用户需修改生成的译文，对译文加工润色，直至满意为止。

2. 机器翻译系统的结构

按结构可把机器翻译系统分为三部分：词典、规则库和程序。前两部分统称为知识库。

(1) 词典

词典包括基础词典、专业技术词典、双语词典和惯用法词典以及用户自定的专业技术词典。

基础词典包括英语基本词汇，除日常生活用语外，还包括一些科学技术、医学等方面的常用词汇。

专业技术词典是选配的词典。它可以根据用户的需要，按用户所翻译的专业领域进行选配，例如，计算机、经济、通讯、火力发电、印刷机械、陶瓷、汽车和拖拉机、石油物探等专业技术词典。

双语词典是一部英汉对译的可面向人用的计算机化的英汉词典。

惯用法词典实质上是一部规则词典。它根据据单词的习惯用法，列出一系列的判别规则，以保证释义的正确通顺。

(2) 规则库

规则库包括英语分析规则和汉语生成规则两大部分。这些规则用于对自然语言的词法、句法、语义等不同层次的抽象结构进行形式化描述，用于进行原文分析和译文生成。

英语分析规则实现的主要功能是：对原文句子进行句法分析和语义分析的加工，在处理共性问题的同时解决个性问题，利用静态信息置换为动态的联系特征，在自底向上加工的同时做自顶向下的检查，通过多模块、多遍扫描将原文的链结构转换为多结点、多叉树的结构。

汉语生成规则是在原文分析得到的多结点、多叉树结构的基

础上，从根结点开始逐层次地由多个模块经多次扫描的过程，最后得到的是汉语译文的链结构。在单句本身生成之后，再确立从句与主句的位置关系。

(3) 程序

程序有核心程序及系统开发和应用软件包两大部分。

核心程序包括原文预处理及查词典、分析与转换、汉语链生成等程序模块，是机器翻译系统的核心。

系统开发和应用软件包分为两个层次，分别提供给开发者和用户。它给开发者创造一个良好的系统开发环境，给用户提供一个友好的应用环境。它包括词典和规则的维护程序、录入程序、显示程序等管理程序。简要解释如下:

①词典管理

词典管理分为两个层次。一个层次是面向系统开发者，提供了词典录入、批量装载、词典文法的自动检查、自动填词以及修改、显示等功能。另一个层次是面向用户，所不同的是提供的信息比较直观简单。

②规则管理

规则管理仅面向系统开发者，用于规则的录入、语法检查、修改、统计等。

③中间结果输出

机器翻译在每一阶段都有中间结果产生出来，及时观察分析每一中间结果是对系统进行维护的必要手段。系统允许开发者将这些中间结果显示或打印出来。另外，用户在建立自己的专用词典时，也可以要求系统提供一定的中间结果信息。

④语言分析统计

语言分析统计就是对原文中出现的静态信息和动态信息进行分析。静态信息是指词数、句长、句数、词的原始信息等，动态信息是指时态、语态、句型、同形数量与判定结果等。这种调查分析对于系统开发者宏观地认识语言问题是非常有用的。

第九章　软件汉化的基本概念

随着计算机在国内的推广使用，对中文软件的需求日益增多。目前，软件汉化是中文软件开发的主要途径。由于西文软件层出不穷，版本不断翻新，致使软件汉化工作的负担越来越繁重。为了提高软件汉化工作的效率和质量，必须着手研究软件汉化工作的规律，使其逐渐形成一套完整的理论。虽然目前软件汉化距离形成一门学科尚不成熟，但是，我们相信，随着软件汉化研究的不断深入发展，软件汉化理论会逐步完善，软件汉化终将会成为计算机科学的一门新学科。

本章力图总结和概括软件汉化的任务、层次、方式、原理、工具等基本概念，第十章和第十一章将分别讨论操作系统的汉化和基于中文操作系统的软件汉化，试图为软件汉化新学科的形成起到抛砖引玉的作用。

第一章到第八章主要从应用的角度讨论了中文软件的基本概念和特点。中文软件的应用和设计是相辅相成的。要进行中文软件设计，必须首先学会如何应用中文软件；了解中文软件的应用，有助于中文软件的设计。反之，中文软件的应用依赖中文软件的设计，中文软件的设计制约着中文软件的应用。因此，通过前八章的内容，先了解中文软件应用的一般原则，了解中文软件与西文软件在使用方面的区别，对于软件汉化是必要的。可以说，前八章的内容是软件汉化的预备知识。

9.1　软件汉化的任务

软件汉化，就是对西文软件实行汉字化或中文化。

软件汉化的任务及其解决的问题可概括如下:

(1) 增加汉字处理功能，解决西文软件的“先天不足”问题;;

(2) 考虑中文环境下的特殊性问题，解决西文软件的“水土不服”问题;

(3) 提示信息汉化，解决西文软件的“言语不通”问题。

本节将逐一讨论软件汉化的这三方面任务。

9.1.1 增加汉字处理功能

由于西文软件是为处理西文设计的，未考虑汉字和汉字处理的特殊性，更未考虑中国人的习惯，因此，原有的西文处理功能对汉字处理无能为力。为了弥补这种“先天不足”，需要对西文软件增加汉字处理功能。

软件汉化在对西文软件增加汉字处理功能的同时，应当考虑以下几个问题:

1. 保持中西文兼容性

保持汉字与西文字符处理、存储和输出格式的一致性，使凡是能够处理西文字符的场合同时也能够处理汉字。

2. 尽量保持西文软件的原有功能

软件汉化有时可能会被迫牺牲原有的一些功能，但绝不允许丧失西文软件的主要功能。

3. 充分考虑汉字和汉字处理的特点

例如，汉字的绕回问题，汉字的排序问题，汉字输入的灵活性、汉字识别和处理的效率性、汉字显示和打印的多样性等。

4. 符合中国人的习惯

例如，中国人习惯于表格处理，因此软件汉化最好考虑表格处理问题，尤其是显示和打印表格线的问题。

5. 使软件汉化工作量达到最低限度

为了使西文软件尽量少做修改甚至不做修改便可处理汉字，

常常采取下列措施:

(1) 选择汉字内码尽量保持与西文字符编码在信息加工处理上的一致性，并尽量避免产生二义性。同时，把汉字纳入程序语言等软件中常有的字符类型或字符串类型，把汉字与西文字符一视同仁。基于这两项原则，汉字与西文字符在信息加工处理方面没有本质的区别，因此，软件汉化的工作量主要集中在增加汉字输入输出功能上。

(2) 采用基于中文操作系统的软件汉化途径，并尽量采用硬件方式实现操作系统的汉化，使操作系统的汉字处理功能尽可能强，从而使基于中文操作系统的各个高层软件的汉化工作量，乃至整个计算机系统的软件汉化工作量尽可能少。

6. 提供友好的用户界面

例如，应具有便于用户掌握的多种汉字输入输出方法和交互式操作方式等。

7. 考虑系统的可扩充性

鉴于汉字处理技术不断发展，已建立的中文系统应具有可扩充新的汉字处理功能的能力。例如，当用户要求增加新的汉字输入方法或变更外部设备时，必须具备灵活的扩充能力。

8. 考虑系统的经济性

中文系统的建立要权衡系统功能与价格，权衡硬件实现技术与软件开销的合理性，用尽可能低的成本实现尽可能多的汉字处理功能。

9.1.2 考虑中文环境下的特殊性问题

由于汉字的特殊性和汉字处理的特殊性，软件汉化所建立的中文环境与原西文环境有一定的差异，因此，当把原来在西文环境中运行的西文软件搬到中文环境中运行时，在某些方面（例如，显示方式、屏幕显示行数、显示字符和打印字符与汉字内码的冲突等）可能会不适应于中文环境，因而会出现“水土不服”的

现象。为了解决这一问题，要求西文软件的原有功能必须适应于中文环境，西文字符的处理必须适应于汉字处理。

例如，CCDOS 采用图形显示方式来显示汉字，则对于 CGA 彩色显示器，一屏只能显示 11 行 16×16 点阵汉字，其中 10 行正文，1 行提示。因此，在 CCDOS 所建立的中文环境中，无论是操作系统，还是程序语言、数据库管理系统或通用应用软件，乃至应用系统都只能显示 10 行正文，由此导致屏幕画面显示不完整，而且还存在屏幕滚动问题，这就无法保持软件在西文环境下显示 25 行的原有功能。为此，无论哪一层次的软件汉化，都要考虑屏幕行数的协调问题，凡是与屏幕显示有关的地方都需要汉化，以便使各层次的软件都适应 CCDOS 的中文显示环境。

又例如，在很多中文系统中，汉字内码的码值往往占据了 ASCII 扩充字符集中图形字符的码值位置，比如，许多西文软件常常采用 ASCII 扩充字符集中的制表符和背景符来显示边框线、表格、窗口和背景。因此，西文软件中凡是显示这些图形字符的地方均会出现汉字，两两显示一个汉字。为此，无论哪一层次的软件汉化，凡是显示这些图形字符的地方都必须汉化，而且凡是涉及这些图形字符的命令、语句和函数也均需要汉化，以适应中文环境的要求。同样，这些图形字符的打印也存在类似的问题。

当然，为了考虑在中文环境下的特殊性问题，在不得已的情况下，可能要牺牲原西文软件的一些次要功能，但绝对不能破坏西文软件原有的主要功能。尽量保持西文软件的原有功能，这是软件汉化的一个重要原则。

9.1.3 提示信息汉化

软件通过提示信息为用户提供适于交互式人机对话的友好界面。由于在中文语言环境中使用西文提示信息存在“言语不通”的

问题，因此必须对西文提示信息实行汉化。

提示信息汉化，就是把西文软件中的西文提示信息替换为相应的中文提示信息。提示信息包括显示提示信息和打印提示信息。

增加汉字处理功能和考虑中文环境下的特殊性问题，是软件汉化的必要条件，而提示信息的汉化则是软件汉化的充分条件。提示信息是否需要汉化，取决于软件的使用环境和使用对象。对于中国的非专业人员，特别是对西文了解不多的用户，提示信息汉化是必要的。然而，对于象新加坡这样一些中英文并用的地方，提示信息无须汉化。即使是在中国，对于专业人员，提示信息也无须汉化。因此，汉化软件一般应提供具有中文提示信息和具有西文提示信息的两种版本。

提示信息汉化要注意以下几点:

(1) 翻译要准确，要在理解软件用法之后再进行翻译。

(2) 在修改提示信息时，千万不要修改提示信息的结束标志及非显示或非打印字符。

(3) 有的提示信息一身兼两用，即当作提示信息，又作为程序的一部分，提示信息汉化时一定要小心，以免发生副作用。

(4) 有些西文软件对重要的提示信息（例如，版权屏幕）往往采用密码方式给出，这些信息在汉化之前首先需要解密，然后再根据密码方式进行汉化。

9.2 软件汉化的层次

电脑中的软件是分层次的。不同层次的软件具有不同的软件汉化方法。

按软件层次与汉字处理功能的传递性，可大致把软件汉化分为以下两种途径:

(1) 基于低层中文软件的软件汉化

由于低层软件是中文软件，它的汉字处理功能能够传递给高层软件，也就是说，高层软件依赖于低层软件的汉字处理功能，受低层软件汉字处理功能的影响和制约；因此，基于低层中文软件的软件汉化，只需在继承低层中文软件的汉字处理功能基础上补充增加无法从低层中文软件获得的汉字处理功能。

(2) 基于低层西文软件的软件汉化

由于低层软件是西文软件，而西文软件中无汉字处理功能，这样，高层软件对它的汉字处理功能的依赖性也就无从谈起；因此，基于低层西文软件的软件汉化，必须在本层软件上重新建立汉字处理功能，通常是通过扩充汉字处理模块来实现的，例如，汉字输入模块、汉字显示模块、汉字打印模块等。显然，基于低层西文软件的软件汉化比基于低层中文软件的软件汉化工作量大得多。

由此可见，软件层次、汉字处理功能的传递性，以及在哪一软件层次上建立汉字处理功能，是决定软件汉化方法的重要因素。本节将从这个角度去讨论各个软件层次上软件汉化的特点。

9.2.1 操作系统的汉化

操作系统的汉化，通常采用汉字处理模块代替西文处理模块。由于这样做并不改变操作系统的系统功能调用界面，于是，高层软件可以利用原有的用于调用西文处理模块的系统功能调用，去调用中文操作系统的汉字处理模块。因此，中文操作系统建立的汉字处理功能很容易传递给它所支持的程序语言、数据库管理系统、通用应用软件，甚至应用程序。

由于操作系统是计算机系统的软件核心，整个计算机系统的各层次软件都依赖于中文操作系统的汉字处理功能，并受中文操作系统汉字处理功能的影响和制约；因此，基于中文操作系统的软件汉化，无需在每个高层软件中都单独地重复性地建立汉字处理功能，而只需在操作系统中把公用的汉字处理功能（主要是汉

字输入输出功能）建立一次，便可供各个高层软件共同使用，这样，各个高层软件只需补充增加无法从中文操作系统获得的特有的汉字处理功能。因此，在操作系统这一层上建立汉字处理功能，是整个计算机系统软件汉化的最简单最有效的方法。

衡量操作系统汉化好坏的一个重要标准是，它所建立的汉字处理功能是强还是弱。中文操作系统所建立的汉字处理功能越强，基于中文操作系统的高层软件的汉化就越容易。例如，有些中文操作系统采用图形显示方式来显示汉字，则原来在西文操作系统环境下以字符显示方式显示西文字符的西文软件，在这种中文操作系统环境中无法显示汉字，必须对基于这种中文操作系统的高层软件在屏幕显示方面进行汉化。然而，如果中文操作系统用硬件实现汉字的字符显示方式，则把原来在西文操作系统环境中运行的西文软件移植到这种中文操作系统环境中运行时，在屏幕显示方面基本上无须汉化便可显示汉字，有的西文软件甚至可以原封不动地搬到这种中文操作系统环境中运行。显然，基于后一种中文操作系统的软件汉化比基于前一种中文操作系统的软件汉化容易得多。因此，操作系统的汉化，必须考虑它的汉字处理功能对高层软件汉化工作的影响。

9.2.2 程序语言、数据库管理系统和通用应用软件的汉化

程序语言、数据库管理系统和通用应用软件的汉化，一方面要考虑它对操作系统的汉字处理功能的依赖性，另一方面要考虑它本身的汉字处理功能对应用程序及其数据的影响和制约。

程序语言、数据库管理系统和通用应用软件的汉化，指的是对它的翻译程序的汉化，有以下两种增加汉字处理功能的途径：

1. 基于中文操作系统的软件汉化

这是程序语言、数据库管理系统和通用应用软件汉化的一种常用的、简单的、切实可行的途径。

由于操作系统的汉化并未改变高层软件对操作系统的系统功

能调用界面，基于中文操作系统的程序语言、数据库管理系统、通用应用软件的解释程序或由编译程序产生的目标程序，可以利用原有的系统功能调用来调用中文操作系统的汉字处理模块，因此，它们很容易继承中文操作系统的汉字处理功能（主要是输入输出功能）。每当用户要输入输出汉字时，解释程序或由编译程序产生的目标程序中都含有相应的系统功能调用，这些系统功能调用原来用于调用西文处理模块，而现在用来调用汉字处理模块。相应地，解释程序执行系统功能调用的程序以及编译程序中用于产生系统功能调用的程序均无需改变，无需汉化，便可使用中文操作系统的汉字处理功能。

鉴于上述原因，基于中文操作系统的程序语言、数据库管理系统或通用应用软件的汉化，只需补充增加无法从中文操作系统继承的汉字处理功能。

2. 基于西文操作系统的软件汉化

这是程序语言、数据库管理系统和通用应用软件的另一种软件汉化途径。

基于西文操作系统的软件汉化，可以看作是基于中文操作系统的软件汉化的一种极端情况。也就是说，由于西文操作系统中无汉字处理功能，高层软件对它的汉字处理功能的依赖性也就无从谈起。因此，基于西文操作系统的程序语言、数据库管理系统、通用应用软件的汉化，必须在它们的翻译程序中重新建立汉字处理功能，通常是通过扩充汉字处理模块来实现的，以使它们的解释程序或由编译程序产生的目标程序能够调用这些汉字处理模块。

9.2.3 应用程序的汉化

应用程序汉化的目的是使它所形成的最终用户界面具有汉字处理功能。

解释程序解释执行源程序本身。编译程序把源程序转换为目

标程序。本节将讨论的应用程序的汉化，指的是对它的源程序的汉化。至于目标程序的汉化，由于目标程序与翻译程序属于同一软件层次，因此，目标程序的汉化类似于翻译程序的汉化，也存在着两种增加汉字处理功能的途径：基于中文操作系统的软件汉化和基于西文操作系统的汉化。本节就不再讨论了。

重新设计和开发中文操作系统与操作系统汉化的基本思想是一致的，重新设计和开发中文程序语言、中文数据库管理系统或中文通用应用软件与程序语言、数据库管理系统或通用应用软件汉化的基本思想是一致的，只是在软件设计和编程过程中充分考虑汉字处理功能。同样，在用程序语言、数据库管理系统、通用应用软件编写程序或处理数据的过程中，充分考虑汉字处理功能，实质上就是应用程序及数据的汉化过程。当然，亦可在原有西文应用程序及数据基础上增加汉字处理功能。

应用程序及数据的汉化，有以下两种增加汉字处理功能的途径：

1. 基于低层中文软件的应用程序的汉化

应用程序是通过使用程序语言、数据库管理系统、通用应用软件提供的命令、语句、函数来继承低层中文软件的汉字处理功能的。这些有关汉字处理的命令、语句、函数被解释程序解释执行，或先由编译程序转换为目标程序，然后再执行之。如果应用程序中使用的有关汉字处理的命令、语句、函数能够被解释程序解译为或被编译程序编译为中文操作系统的系统功能调用，那么，应用程序所继承的汉字处理功能是在操作系统这一层上建立的；否则，应用程序所继承的汉字处理功能就是在程序语言、数据库管理系统、通用应用软件这一层上建立的。

基于低层中文软件的应用程序，除了通过使用程序语言、数据库管理系统、通用应用软件提供的命令、语句、函数来继承低层中文软件的汉字处理功能外，还可以通过使用它们提供的外部接口命令、语句或函数来调用汉字处理模块，用以在应用程序这

一层上补充增加无法从低层中文软件获得的汉字处理功能。

2. 基于低层西文软件的应用程序的汉化

基于低层西文软件的应用程序的汉化，可以看作是基于低层中文软件的应用程序的汉化的一种极端情况。也就是说，由于低层西文软件中无汉字处理功能，应用程序对低层西文软件的汉字处理功能的依赖性也就无从谈起。因此，基于低层西文软件的应用程序的汉化，必须在应用程序这一层上重新建立汉字处理功能，通常是通过使用程序语言、数据库管理系统、通用应用软件提供的外部接口命令、语句或函数调用汉字处理模块来实现的。

9.3 软件汉化的方式

软件汉化大致有以下三种方式:

1. 纯软件方式;

2. 纯硬件方式;

3. 软件与硬件结合方式。

第一种方式用于各个层次软件的汉化，而第二种方式和第三种方式主要用于操作系统的汉化。

本节将分别介绍这三种软件汉化方式，并对它们进行比较。

9.3.1 纯软件方式的软件汉化

纯软件方式的软件汉化，就是只通过软件改造来对西文软件增加汉字处理功能。这种软件汉化方式用于各层次软件的汉化。

这种软件汉化方式，在选择适当的汉字设备（汉字键盘、汉字显示器、汉字印刷机、汉字终端等，见 2.4，2.5，5.2.1 节）基础上，将汉字处理功能全部用软件来实现。

纯软件方式的软件汉化包括以下两种方法:

1. 直接修改西文软件

直接修改西文软件是一种常用的纯软件方式的软件汉化方

法。

例如，基于 CCDOS 的 dBASE，Wordstar，Lotus1-2-3 等高层软件的汉化，均是通过直接修改西文软件来继承 CCDOS 的汉字处理功能的。

当然，这种软件汉化方法存在一个版权问题。解决办法是：

(1) 与西文软件的开发者联合汉化；

(2) 仅仅提供修改西文软件的 MODIFY 或 DRIVER 程序。

2. 在西文软件外层上加一个汉字处理功能外壳。例如，在程序语言的编译程序的外层增加一个独立的预编译程序，用于处理汉字。中文程序语言的编译过程是：首先，通过预编译程序把中文程序语言程序转换为西文程序语言程序，主要是翻译中文程序语言中具有汉字处理功能的语句、函数及其它语言成分；然后，再用原编译程序按常规进行编译。这种预编译程序就是一种汉字处理功能外壳，可用高级语言来编写，因此便于移植。

又例如，利用 AUTOCAD 提供的外部接口，在西文 AUTOCAD 基础上增加一个汉字处理功能外壳，构成了中文 AUTOCAD。

由此可见，这种纯软件方式的软件汉化方法无需修改西文软件，不牵涉版权问题，为西文软件增加汉字处理功能提供了灵活性，并可根据用户的需要和特定的外部设备进行汉字处理功能的扩展。

9.3.2 纯硬件方式的软件汉化

纯硬件方式的软件汉化，就是只通过硬件改造来对西文软件增加汉字处理功能，而不必修改西文软件，也不必在西文软件外层上加一个汉字处理功能外壳。这种软件汉化方式主要用于操作系统的汉化。

这种软件汉化方式对计算机系统中所有的汉字设备均进行硬

件改造，把汉字设备驱动模块固化到 ROM 上，用到汉字设备上去。我们把这种设备称为接插兼容的汉字设备。接插兼容的汉字设备虽然是硬件，但它的设计思想是从软件来的，它的硬件组织中包含了固化的软件功能。例如，接插兼容的汉字终端含有固化的汉字终端驱动模块（包括汉字字库和汉字输入码到汉字内码的转换表），接插兼容的汉字印刷机含有固化的汉字印刷机驱动模块（带有汉字字库）等等。由于接插兼容的汉字设备能够区分汉字和西文字符，因此，对于与这种汉字设备连接的主机来说，无需区分汉字和西文字符，从而不需要增加新的汉字设备驱动模块，只要用原来的西文设备（字母数字设备）驱动模块，就可以驱动接插兼容的汉字设备来输入输出汉字。

9.3.3 软件与硬件结合方式的软件汉化

软件与硬件结合方式的软件汉化，就是用软件改造与硬件改造相结合的方式来对西文软件增加汉字处理功能。这种软件汉化方式主要用于操作系统的汉化。

这种软件汉化方式对计算机系统中的某些汉字设备进行一些硬件改造，同时对软件中不适应的地方进行一些软件改造。例如，在计算机系统中只引入接插兼容的汉字印刷机，那么，对其它的汉字设备仍需要配置相应的汉字设备驱动模块；只把汉字字库固化在汉字终端中，而把汉字输入码到汉字内码的转换表放在汉字终端驱动模块中；用汉卡来容纳汉字字库，或用汉卡来实现汉字字符显示方式，汉字设备驱动模块随之做相应的改变。

9.3.4 操作系统汉化的三种方式

我们可以用软件层次与汉字处理功能的传递性的观点，来解释操作系统汉化的三种方式。

由于接插兼容技术把软件功能固化到硬件上，因此可把固化在硬件上的软件看作是支持操作系统的最低层软件，我们把它称

为固化软件，简称为固件（firmware）。

纯软件方式的操作系统汉化，可看作是基于西文固化软件基础上的软件汉化。也就是说，非接[illegible]的汉字设备中无汉字处理功能，操作系统对它的汉字处理功能的依赖性也就无从谈起，因此，必须在操作系统本身这一层上建立汉字处理功能，通常是通过用汉字处理模块代替原有的西文处理模块来实现的。

纯硬件方式的操作系统汉化，可看作是基于中文固化软件的软件汉化，而且完全依赖于中文固化软件的汉字处理功能。也就是说，固化软件的汉字处理功能非常强，以至于基于它的操作系统无需修改操作系统本身。

软件与硬件结合方式的操作系统，可看作是基于中文固化软件的软件汉化。一方面，操作系统继承中文固化软件的汉字处理功能；另一方面，适当地修改操作系统本身，用以补充增加无法从硬件获得的汉字处理功能。

9.3.5 三种软件汉化方式的比较

采用哪种汉化方式实现软件汉化，这要取决于用户的需要，取决于人力物力等条件。

用纯软件方式实现软件汉化投资少，是一种经济实惠的汉化方式，适合于基于中文操作系统的软件汉化；但是，对于操作系统的汉化，特别是大、中、小型机上操作系统的汉化，工作量大，系统效率低，而且汉字处理功能较弱，致使基于中文操作系统的程序语言、数据库管理系统、应用软件几乎全需要汉化。

用纯硬件方式实现软件汉化，无需对软件进行改造，而且系统效率高；但由于汉字字库及其它一些软件功能固化在每个汉字设备上，不仅投资多，而且无法共享某些软件资源，例如，汉字字库；同时也丧失了某些灵活性，例如，汉字输入方案的随时更换，多字体汉字字库的不断引入等等。

软件与硬件结合方式正是实现软件汉化的一种折衷方案。也

就是说，充分利用软件和硬件的各自优点，在人力物力允许的情况下，尽量提高整个系统中各层次软件的时空效率，并使整个系统的各层次软件的汉化工作量达到最少。

9.4 软件汉化的原理

软件汉化实质上是把被汉化的西文软件看成是一个缺少汉字处理功能的错误程序。软件汉化就是要测试和调试这个程序，直至它具有汉字处理功能为止。

所谓测试，就是设计一些例子去执行程序，从中发现程序中的错误；所谓调试，就是查找程序中的错误，并予以纠正（详见8.2.4节）。

根据测试和调试的原理和方法，我们把软件汉化分为三个阶段：发现“错误”、查找“错误”、纠正“错误”。这里的“错误”是指“先天不足”、“水土不服”、“言语不通”三方面问题。通过这三个阶段的反复使用，反复迭代，反复试验，来完成软件汉化的全过程。

本节将按照测试和调试的一般原则和方法，分别讨论软件汉化的三个阶段。

9.4.1 发现“错误”

软件汉化的发现“错误”阶段，就是测试程序的过程。

测试的关键是如何设计测试用例。设计测试用例的方法可分为两类：黑箱法和白箱法。

对于软件汉化这种特殊的软件测试，它所测试的软件一般均是经过翻译的目标代码，而不是源代码。软件汉化人员往往只有软件的使用说明书，很少具有软件的技术说明书，因此，面对可读性极差的目标代码，很难了解程序的内部结构，很难了解程序的内部逻辑。鉴于上述原因，软件汉化设计测试用例一般无法采

用白箱法，而只能采用黑箱法。

黑箱法的测试用例由两部分组成：输入数据和预期的输出结果。如果软件汉化的对象是操作系统，则输入数据是指它的内部功能（例如，汉字输入和显示功能）和外部命令（例如，汉字打印命令）等。如果软件汉化的对象是程序语言、数据库管理系统或通用应用软件，则输入数据是指应用程序和数据文件等。如果软件汉化的对象是应用程序，则输入数据就是应用程序的最终用户界面所使用的输入数据。在执行程序（指被汉化的软件）之前应该对期望的输出有很明确的描述，测试后可将程序的输出结果同它们仔细地对照检查。如果不事先确定预期的输出，就有可能把似乎是正确的而实际上是错误的结果当成是正确的结果。

测试是为了发现错误而执行程序，而不是为了证明这个程序能正确地执行它应有的功能；因此，那些只能使程序正确执行的例子是没有意义的，而能够发现错误的例子才是有意义的测试用例。对于软件汉化，被汉化西文软件的原有功能已经过严格测试和调试，不需要再进行测试和调试，它的错误仅仅在于缺少汉字处理功能，因此，软件汉化测试用例的设计只需考虑能够发现这种错误的例子。

测试用例是根据程序的功能描述设计的。软件汉化的测试用例必须根据汉字处理功能描述来设计。为此，软件汉化人员应事先对被汉化的软件拟定一个汉字处理功能描述，也就是说，要明确在西文软件基础上需要增加哪些汉字处理功能。至于需要增加哪些汉字处理功能，取决于要被汉化的软件本身以及用户的要求。例如，对于操作系统的汉化，需要增加汉字输入、汉字显示、汉字打印等功能，但具体设置哪几种汉字输入方案和哪几种输出字形，取决于用户的要求。又例如，对于程序语言的汉化，需要增加汉字的连接、比较、排序、查找、截取、赋值、输入、显示、打印等汉字处理功能。本着把汉字纳入程序语言的字符类型或字符串类型的原则，凡是能处理西文字符的语言成分都应能

处理汉字，例如，名字（变量名、文件名、过程名、函数名等）、字符型数据、语句、命令、函数、文本文件等。程序语言中引入汉字处理功能的关键问题是：如何在程序语言的字符类型或字符串类型中扩充汉字。再例如，对于字处理软件的汉化，增加汉字处理功能就是使它具有中西文两种文字的处理功能，原来用于西文处理的插入、删除、增加、替换等功能亦可用于汉字处理，同时要考虑汉字处理的特殊性问题，比如，换行时右端不应出现半个汉字的禁则现象，在文本文件中插入 Esc 序列或其它控制符来变换汉字打印字形等。

软件汉化测试用例的设计也采用黑箱法的两种常用方法：等价分类法和边界值分析法。

下面，让我们首先来看看如何在软件汉化的发现“错误”阶段采用等价分类法设计程序用例。

根据汉字处理功能描述中的“标识符是以汉字或字母开头的由汉字、字母、数字组成的字符串”这一输入条件，可以划分以下七个合法等价类：

(1) 首字符为汉字；
(2) 首字符为大写字母；
(3) 首字符为小写字母；
(4) 非首字符含有汉字；
(5) 非首字符含有大写字母；
(6) 非首字符含有小写字母；
(7) 非首字符含有数字。

设计三个例子，使它们代表上述所有的合法等价类，例如，汉字 Aa1，A 字 Aa1，a 字 Aa1。

考虑到西文软件的原有功能已经过严格测试和调试，“标识符是以字母开头的由字母、数字组成的字符串”这一输入条件无需重新测试，因而在设计测试用例时可省略上述第 (2)、(3)、(5)、(6)、(7) 个等价类，只考虑第 (1)、(4) 个等价类即可，

因此上述测试用例可简化为一个：汉字。

又考虑到第（1）个和第（4）个等价类往往对应两处程序，为了分别测试和调试这两处程序，常常各设计一个测试用例，分别代表第（1）个和第（4）个等价类，例如，汉 a，a 汉。

同样，根据汉字处理功能描述中的“标识符是以汉字或字母开头的由汉字、字母、数字组成的字符串”这一输入条件，可以划分以下两个非法等价类：

（1）首字符为除汉字和字母外的其它字符；

（2）非首字符含有除汉字、字母、数字外的其它字符。

设计两个例子，使它们代表上述所有的非法等价类，每个例子各包含一个非法等价类，例如，9a，a[。

考虑到汉字、大写字母、小写字母、数字的 ASCII 编码并不连续（汉字编码为 A1−FF（这是简化值，准确值为 A1A1−FEFE），大写字母编码为 41−5A，小写字母编码为 61−7A，数字编码为 30−39），程序在检测首字符是否为除汉字和字母外的其它字符以及非首字符是否为除汉字、字母、数字外的其它字符时要在几处分别进行，因此，为了分别测试和调试这几处程序，应划分为以下七个非法等价类：

（1）首字符编码小于 41；

（2）首字符编码大于 5A 且小于 61；

（3）首字符编码大于 7A 且小于 A1；

（4）非首字符编码小于 30；

（5）非首字符编码大于 39 且小于 41；

（6）非首字符编码大于 5A 且小于 61；

（7）非首字符编码大于 7A 且小于 A1。

设计七个例子，使它们代表上述所有的非法等价类，每个例子各代表一个非法等价类，例如，9a，[a，{a，a／，a：，a[，a{。

又考虑到西文软件的原有功能已经过严格测试和调试，“标识

符是以字母开头的由字母、数字组成的字符串”这一输入条件无需重新测试，因此在设计测试用例时可省略上述的第（1）、（2）、（4）、（5）、（6）个等价类，只考虑第（3）、（7）个等价类即可，因此上述测试用例可简化为两个：{a 和 a{。

同时，我们还看到，划分为“首字符为除汉字和字母外的其它字符”和“非首字符含有除汉字、字母、数字外的其它字符”这样两个非法等价类是不适宜的。代表这两个非法等价类的例子是很难准确选择的，也就是说，所选择的测试用例不一定具有代表性。例如，9a，a[，它们虽然各代表上述的两个非法等价类之一，但这两个测试用例只能用于测试西文软件的原有功能，而西文软件的原有功能已经过严格测试和调试。因此，划分为上述七个非法等价类是适宜的，由此而选择的测试用例是具有代表性的，简化后的两个例子 {a 和 a} 用作软件汉化测试用例是恰当的。

必须注意，在设计包含非法等价类的测试用例时，每个例子只能代表一个非法等价类。这是因为，程序中的某些错误检测往往会抑制其它的错误检测。如果一个例子代表两个非法等价类，例如，{{，则程序在发现“首字符为 {”非法之后，可能不会再检查“非首字符含有 {”是否合法，因此后一处程序实际上没有测试到。

下面，让我们再来看看如何在软件汉化的发现“错误”阶段采用边界值分析法设计测试用例。

根据汉字处理功能描述中的“标识符是以汉字或字母开头的由汉字、字母、数字组成的字符串”这一输入条件，按照等价分类法设计的代表合法等价类的软件汉化测试用例是汉 a 和 a 汉，代表非法等价类的软件测试用例是 {a 和 a{。由于它们是从等价类中任选的例子，因此不一定具有代表性，有的错误不一定能暴露出来，例如，在程序中检测判断是否属于汉字时，如果程序中的判断范围误为比 A1–FF 窄，则通过测试用例“汉 a”和“a

汉”是无法检验出这一程序错误的。类似地，如果程序中的判断范围误为比 A1−FF 宽，则通过测试用例“｛a”和“a｛”是无法检验出这一程序错误的。因此，采用边界值分析法，选择检验边界情况的测试用例，才能暴露出上述程序错误。

等价分类法通过划分等价类来选择测试用例，而边界值分析法着重检查等价类边界上的情况。在软件汉化过程中，应把这两种方法结合起来使用。一方面，要使测试用例尽可能少，以减少工作量，提高测试效率；另一方面，又要确保测试结果可靠，使汉化软件达到实际使用所提出的可靠性要求。

通过执行被汉化软件，检验所选择的测试用例是否符合预先拟定的汉字处理功能描述，你会发现，有的测试用例符合汉字处理功能描述，有的测试用例不符合汉字处理功能描述，前者说明被汉化软件本来就具有这种汉字处理功能，因此在这一点上无需汉化。后者说明被汉化软件不具有这种汉字处理功能，因此在这一点上需要汉化，相应的测试用例要用于软件汉化的全过程。至于所设计的测试用例是否合乎预先拟定的汉字处理功能描述，也就是说，哪些地方需要汉化，取决于软件汉化的层次、方式以及被汉化的软件本身。例如，基于中文操作系统的软件汉化，有些西文软件在某些操作方面（比如，显示字符串）能处理高位为 1 的字符，因此在这些操作方面也能处理汉字，比如，基于 CCDOS 的 BASIC、FORTRAN 语言等软件；而有些西文软件在某些操作方面不允许使用高位为 1 的字符，因此在这些操作方面无法处理汉字，必须对它进行汉化，比如，基于 CCDOS 的 COBOL 语言的字符串显示、dBASEⅢ PLUS 的 MODIFY COMMAND 编辑操作等。

在发现“错误”的过程中，无论是拟定汉字处理功能描述，还是设计测试用例，或是运行软件试用各种汉字处理功能，都需要充分了解被汉化的西文软件的功能和所处的环境，尤其是与增加汉字处理功能有关的原有功能。

总之，软件汉化发现“错误”阶段的任务是：根据西文软件的功能和用户的要求，对被汉化软件拟定汉字处理功能描述，并据此设计测试用例，通过运行软件试验测试用例是否符合汉字处理功能描述，把不符合汉字处理功能描述的测试用例继续用于软件汉化的另外两个阶段。

9.4.2 查找“错误”

软件汉化查找“错误”阶段和纠正“错误”阶段，就是调试程序的过程。

查找错误就是检测、跟踪、定位程序中的错误。软件汉化的查找“错误”阶段是软件汉化的关键阶段。它的任务是根据出错症状诊断出错位置和出错原因。出错症状是根据从发现“错误”阶段得到的不符合汉字处理功能描述的测试用例观察而来的。查找错误主要依靠软件汉化人员的推理和归纳能力，也要借助于一些调试工具。在软件汉化中查找“错误”可采用下列两类不同的方法：

1. 白箱法

这种方法把对被汉化软件的执行路径的分析看作是一个“黑箱→灰箱→白箱”的测试和调试过程。为了确定“出错”位置，要对被汉化软件的目标程序的功能和结构进行反复的较全面的分析，以使它的内部逻辑逐步清晰。这种方法的可靠性高，有利于从根本上改造西文软件。然而，这种方法往往是针对可读性极差的目标程序的反汇编代码分析的，因此在没有软件技术说明书的情况下，要全面了解整个软件的功能是不容易的，工作量很大。鉴于上述原因，除非要对西文软件做彻底的改造，很少有人把白箱法用于软件汉化的查找“错误”阶段。

2. 灰箱法

这种方法把对被汉化软件的执行路径的分析看作是一个“黑箱→灰箱”的测试和调试过程。所谓灰箱，是指通过分析对整个软件的功能和结构有个大概的了解。这种方法强调针对问题的现

象有目的地寻找和定位与测试用例等价类相对应的目标代码段。这种方法既避免了繁重的全局分析工作，又可有效地确定“出错”位置，因此，灰箱法是软件汉化查找“错误”的行之有效的方法。

上述两种方法均采用在发现“错误”阶段设计的测试用例来查找“错误”，而测试用例又是采用黑箱法设计的，因此，上述两种查找“错误”的方法均是从黑箱着手使用的。

灰箱法查找“错误”方法一般采用下列具体方法，往往是各种具体方法配合使用。

(1) 猜测试探查找法

根据从发现“错误”阶段得到的不符合汉字处理功能描述的测试用例，从错误的迹象和线索着手，提出对出错原因的种种猜想，列出发生错误的所有可能原因，并用测试用例去试探可能会引起这些原因的程序代码，通过调试，逐个证实或排除有关出错原因的猜想，最后把出错范围缩小到最低限度，再经过认真分析和试验，最终找到程序出错的位置。

例如，根据出错时响铃这一迹象，在程序中查找响铃指令，顺藤模瓜，从而找到出错位置。

又例如，根据输入输出汉字时在屏幕上出现替代字符（比如,??）这一现象，在程序中查找把替代字符传送到某单元的指令（这个替代字符可能存于数据段中被间接取出，也可能以立即方式传送），试探有可能引起这种现象的种种程序代码，直至找到出错位置为止。

猜测试探查找法没有确定的步骤，在很大程度上是凭经验或直觉来猜测程序出错原因和试探程序出错位置。这些经验取决于软件汉化人员对被汉化软件了解的程度，是否熟悉编译原理，程序设计经验多少等。例如，查找与测试功能对应的目标代码，最好具有编译知识，知道哪种功能应对应哪种形式的目标结构。由此可见，尽管猜测试探法似乎是一种“瞎凑”、“蛮干”和“凭运气”的方法，但对于具有丰富的程序设计经验和软件知识的软件汉化

人员来说，却是一种常用的行之有效的方法。

(2) 逻辑分析查找法

逻辑分析查找法从分析程序的内部逻辑着手查找“错误”。

对于全局逻辑分析，要分析被汉化软件的文件结构及其内存映象结构，分析各个文件及所含子程序的功能及其相互调用关系。通过全局逻辑分析，可大致了解整个软件的整体逻辑结构，为进一步进行局部逻辑分析奠定了基础，提供了前提。

当我们通过全局逻辑分析，确定或基本上确定错误所在文件或子程序时，就可以对其中的各个疑点所在的局部程序逐一进行局部逻辑分析，进一步确定出错原因和出错位置。局部逻辑分析又可以分为静态分析和动态分析两种。静态分析就是把与错误有关的局部程序的反汇编代码打印出来，仔细分析。动态分析就是通过调试工具分析程序的执行情况。静态分析可以帮助我们掌握局部程序的静态结构，动态分析则可以帮助我们了解局部程序的动态执行过程。在局部逻辑分析过程中，往往要把静态分析和动态分析结合起来使用。这种对局部程序进行逻辑分析的方法，犹如在黑箱中点燃了一个个亮点，亦可称为窥孔分析法。

(3) 启发式查找法

有时，我们可以借助一些启发信息，来分析出错原因和确定出错位置，这样做有时确能找到解决问题的捷径。启发信息来源很多，例如，通过观察工作环境、查看历史、暂时去掉程序的某些部分等手段，可以从中发现一些启发信息。启发式查找法的关键在于善于发现和善于利用启发信息。

例如，在 dBASEⅢPLUS 伪编译程序的软件汉化过程中，我们发现，在字段名和内存变量名中，不仅不允许有汉字，而且把{ ; }～△也视为非法字符（它们的 ASCII 码分别为 7B，7C，7D，7E，7F)。由此，我们受到启发：在程序中一定存在判断 ASCⅡ码是否超过 7A 的指令，经过查找反汇编代码，确实存在一条这样的指令：

CMP AX，007A

把 7A 改为 FF，竟轻而易举地解决了这一问题。

(4) 跟踪查找法

在无法采用上述方法找到出错位置的情况下，可采用跟踪程序的方法来查找“错误”。

跟踪是调试程序的一种有效手段。跟踪查找法一般采用线路跟踪方式，即进入主程序后，对每个过程调用设立断点，当发现经过某个过程后出现了错误现象时，再对这个过程进行逐步跟踪，而对该过程所调用的过程继续做如上处理，这样一层一层地跟踪，直到找到出错位置为止。

程序跟踪常把单步方式和断点方式结合起来使用。单步方式是计算机的一种执行方式。在这种方式下，每当用户发出一次命令（通常用按键），机器就执行一条指令。借助于调试工具，用户可以看到每一条指令执行时的操作及结果。在为汇编语言而设计的调试程序中，跟踪信息通常包括上一条指令执行后的累加器值、所有通用寄存器的内容、下一条指令的地址和指令内容等。断点方式通过设置断点来调试程序。断点是程序执行时的停顿点，用户利用调试工具可在任何需要的地方设置程序断点。程序一旦执行到断点位置，便会自动停顿，使用户有可能检查当时的系统状态和执行到断点时的中间结果。在软件汉化的查找“错误”阶段，常常可以用断点执行方式以较快的速度通过确认无错的程序段，而对可能有故障的程序段采用单步方式跟踪。

除线路跟踪方式外，软件汉化的查找“错误”阶段也可采用追溯跟踪方式，即当程序执行出错时，把出错前的一段程序的有关线路和赋值情况输出，以供软件汉化人员分析出错原因用。由于这种跟踪方式从发现错误征兆的地方开始往回追溯程序代码，因此有时称为回溯跟踪方式。

9.4.3 纠正“错误”

纠正错误就是排除和校正程序中的错误。软件汉化的纠正“错误”阶段的任务是：根据出错原因和出错位置，修改被汉化软件的程序，并用测试用例反复执行程序，检验是否符合汉字处理功能描述，直到它具有汉字处理功能并且排除因修改程序而引入的新错误为止。

软件汉化对程序的修改有可能会引入新的错误。这些新错误一般是由下列两类原因引起的：

(1) 程序改错，修改了不应该修改的地方；

(2) 牵一发动全身，产生了副作用。

对于那些由于修改程序而引入的新错误，一方面，在修改程序时，要格外小心，尽量避免动到无关的地方和不应该修改的地方；另一方面，在修改程序之后，要进行回归测试，及时发现这些新错误，并通过调试查找和纠正这些新错误。

软件汉化常常会产生副作用，这是难免的。副作用产生的原因是多种多样的，往往是由于考虑不周或未完全弄清被修改程序的含义而引起的。例如，在被汉化软件中，一处的数据可能被多处程序所使用，因而当修改了这一处数据后，虽使一处程序纠正了错误，但却影响到另一处程序。消除副作用的办法是：查找副作用是由哪一个被修改过的文件、哪一段被修改过的程序、哪一处被修改过的数据引起的，恢复西文软件原来的样子，看看副作用是否消除。如果消除了，则不要再动此处信息，另想办法来做此项汉化工作。

在软件汉化修改程序时，要斟酌修改后的程序代码，最好修改后的程序代码长度比修改前的程序代码短或一样长，以便覆盖。如果修改后的程序代码长度比修改前的程序代码长，那么就需要采用“打补丁”的办法把多余的部分接出来，接到空白位置，有时还需要增加文件的驻留长度。修改程序的方法因软件本身的

性质而异，很难总结出统一的方法。但有一点是值得注意的，就是要讲究修改程序的技巧。

在实际的软件汉化过程中，提示信息汉化往往不严格按照软件汉化的三个阶段逐一进行。西文提示信息常常是以 ASCII 字符的形式存在于文件中，甚至集中在一两个文件中，因此，提示信息的汉化有时可部分省略或全部省略发现“错误”阶段和查找“错误”阶段的工作，而在纠正“错误”阶段直接把被汉化软件中的西文提示信息修改为中文提示信息。

软件汉化对程序的修改是否正确，是否存在副作用，除了回归测试外，还需要在软件汉化后的使用过程中逐渐发现。要及时反馈用户发现的问题，不断地维护和修改汉化软件。这是因为，测试和调试只能证明错误的存在，但不能证明错误不存在。因此，汉化软件作为产品投入使用，必须不断地维护和修改。为此，在软件汉化过程中，一定要认真详细地做好记录，以备后用。

对于直接改造西文软件的软件汉化方式，考虑西文软件的版权问题，常常要提供西文软件改造用的 MODIFY 或 DRIVER 程序。这种程序是在上述软件汉化基础上，根据软件汉化记录或汉化软件与原西文软件的对照比较而编写的。对于在西文软件外层上加一个汉字处理功能外壳的软件汉化方式，亦可采用类似的方法来处理。

以上仅是软件汉化的一般原理和方法，而具体的汉化要根据具体的软件，具体问题具体分析。下面，以基于中文操作系统的 FORTRAN 语言软件汉化中的一例，来简要说明如何按照发现“错误”、查找“错误”、纠正“错误”三个阶段来实现软件汉化的。

(1) 发现“错误”

根据对 FORTRAN 语言拟定的汉字处理功能描述，设计下列测试用例:

```
WRITE (5, 10)
```

```
10  FORMAT (1X, '汉字')
    STOP
    END
```

其中，5 表示设备号，10 表示 FORMAT 语句标号，1X 表示走纸符，单引号中的内容是所要打印的字符串。根据汉字处理功能描述，单引号中的字符串应允许含有汉字。

对该 FORTRAN 源程序进行编译、连接并运行后，发现该目标程序不能打印单引号中的“汉字”二字，可见此处不符合汉字处理功能描述，需要汉化。

(2) 查找“错误”

先用软件汉化工具（例如，DEBUG 调试程序），对上述目标程序进行跟踪查找，发现在处理单引号之前就已经把单引号中的汉字内码的高位 1 滤掉了。找到原因后，再用软件汉化工具对 FORTRAN 编译程序进行查找，将所有处理单引号的地方均找出来进行分析，并通过设置断点进行跟踪，从而找到处理 FORMAT 语句中单引号的位置。

(3) 纠正“错误”

在找到问题出现的原因和位置后，对 FORTRAN 编译程序的有关程序代码进行适当的修改，然后再用修改后的编译程序重新对上述 FORTRAN 源程序进行编译和连接，看看执行相应的目标程序是否可以打印单引号中的“汉字”二字。通过大量例子的检验，FORMAT 语句中处理汉字的问题得到解决。

9.5 软件汉化的工具

由于软件汉化是依照测试和调试程序的原理和方法进行的，因此需要借用测试工具和调试工具当作软件汉化工具。

目前，测试工具尚很少用于软件汉化，而常常把调试工具用于软件汉化。大多数系统都为汇编语言提供了调试工具。调试程

序是一种主要的典型的调试工具。

调试程序是用于帮助程序的调试和排错的一种交互式程序。它通常与被调试的程序一起运行。它一般包括以下功能:

(1) 为被调试程序设置和清除执行时的断点;;

(2) 将控制转移到被检测的程序;

(3) 跟踪程序的执行;

(4) 显示和修改指定的寄存器或存储单元内容;

(5) 打印指定的存储单元的内容;

(6) 在存储器与磁盘或存储器与打印机之间进行存储器内容的转储;

(7) 将存储器的指定存储区填以某种固定值;

(8) 在存储器的指定存储区中查找所需要的字符串。

由于各种计算机系统的汇编语言各不相同，因此它们为汇编语言提供的调试工具也各不相同。例如，DOS 系统提供了 debug，inside，pctools，edlin，type，copy，rename，dir，cv 等调试工具；UNIX 系统提供了 od，ls，ed，cat，cp，mv，tail，adb，gothe，gethead 等调试工具。下面，仅简要介绍 debug，inside，pctools 这三个软件的主要功能。

debug 是 DOS 操作系统为汇编语言提供的调试程序。它可用于提供一个可控制的调试环境，因此可监视和控制被调试程序的执行，直接查找程序中的错误，程序修改后无需重新汇编就可立即执行，从而确定修改是否正确。debug 的许多命令都适合于软件汉化用。例如，D 转储（Dump）命令用于观察提示信息，E 输入（Enter）命令用于修改程序和提示信息，G 执行（Go）命令用于执行被汉化程序；R 寄存器（Register）命令用于观察或修改寄存器的内容；S 查找（Search）命令用于查找字符串；T 跟踪（Trace）命令用于跟踪执行被汉化程序；U 反汇编（Unassemble）命令用于把目标代码转换为类似的汇编语言程序；W 写（Write）命令用于把汉化后的程序或数据保存到磁盘

上；等等。

inside 是一个反汇编程序，它把目标代码转换为类似的汇编语言程序。inside 名字本身的含义是：该程序帮助你进入内部……。inside 除了具有 debug 的功能外，还具有对标号的加入、编辑、删除、打印和存储的功能。inside 反汇编过程需要两遍扫描：第一遍扫描用于建立一个地址表，其中包含被程序调用的每个子程序的入口地址和程序中每条转移指令转向的目标地址；第二遍扫描用于从目标代码形成汇编语言程序，它根据地址表查找每一条机器指令的地址，如果找到一个匹配，则在汇编语言代码的对应行上产生一个标号，并显示、打印或存储在磁盘上。

pctools 是一个功能很强的磁盘管理工具软件（见 7.6 节）。它包括 SHELL，DESKTOP，BACKUP，COMPRESS，CACHE，FORMAT，MIRROR／REBUILD，SECURE 几部分。由于 SHELL 部分具有很强的观察和编辑磁盘文件内部代码的功能，适用于软件汉化。作为软件汉化工具，SHELL 中的浏览／编辑文件命令（View／Edit）和在文件中寻找字符串命令（Find）最为有用，尤其是对提示信息的汉化。View／Edit 命令可按 ASCII 字符或 HEX 值两种方式来浏览文件，并可按两种方式修改文件。Find 命令可在文件中查找指定的 ASCII 字符串或 HEX 值，并确定位置。

debug 和 inside 均用到反汇编。反汇编是一种常用的程序调试手段，是剖析机器语言程序的重要工具，因此也是软件汉化的重要工具。大多数为调试汇编语言程序而设计的动态调试程序中都有反汇编命令。

反汇编可视为汇编的一种逆过程。汇编程序把用汇编语言写的源程序转换为用机器语言表示的目标程序。反汇编程序把机器语言程序转换为类似的汇编语言程序。所谓类似的汇编语言程序，是指经反汇编所得到的指令只有操作码和寄存器为助记符，而操作数地址仍然为十六进制代码形式。这是因为，目标程序中

只包括机器指令代码和运算数据代码，因此不可能完全转译为标准的汇编语言程序形式。此外，在反汇编过程中，目标程序中的数据也会被不加区分地转译为毫无意义的指令助记符，因此，在阅读经反汇编所得到的程序时，要格外留心数据区的地址，以排除那些无意义的助记符的干扰。

近年来，许多科研单位和生产厂家在软件汉化方面已经积累了一定的经验，摸索到一套软件汉化的方法。为了提高软件汉化的效率和质量，有必要总结软件汉化的规律，研制适合软件汉化的专用工具。软件汉化专用工具一方面要吸收软件汉化借用工具的优点，另一方面要力求实现软件汉化的自动化。由于软件汉化是一个复杂的推理、分析、判断、归纳和思维过程，因此软件汉化的自动化很难完全实现。目前只能研制一些软件汉化辅助工具，来帮助软件汉化人员提高软件汉化的速度和质量，例如，提示信息汉化辅助工具（见 11.3 节）。

第十章　操作系统的汉化

在计算机系统中建立汉字处理功能，经历了以下几个发展阶段:

1. 在应用程序中调用汉字处理模块

早在七十年代中后期，国内就有人在计算机上尝试汉字的输入输出。这种方法是在应用程序这一层上建立汉字处理功能。它构造若干个汉字处理模块，主要是汉字输入输出驱动模块。当应用程序运行到要输入输出汉字时，可调用这些汉字处理模块，来完成汉字输入输出及其它汉字处理功能。用户亦可自己编写各种汉字处理模块，以满足用户自己特殊的需要。既可以在汇编语言程序或机器语言程序中直接调用汉字处理模块，也可以在高级语言程序中用 CALL 等外部接口命令、语句、函数调用汉字处理模块。

例如，下列 BASIC 程序显示输出一个汉字:

```
100  A$:="127164"
110  CALL  &B000, VARADR(B$), VARADR(A$)
120  CALL  &B024
130  CALL  &B006, VARADR(C$), VARADR(B$)
140  CALL  &B00F, 2,1
150  CALL  &B01B, VARADR(C$)
160  CALL  &B01E
170  CALL  &B021
180  END
```

第 100 语句将“电”字的键盘输入码作为字符串赋给 A$，其中 A$ 的长度为 6 字节。

第 110 语句将 A$ 中的汉字输入码转换为汉字地址码，送入 B$。其中，用十六进制表示的 B000 为汉字显示模块的入口地址。

第 120 语句将屏幕上的光标消去。

第 130 语句根据 B$ 中的汉字地址码从汉字字库中取出汉字字形信息，送入 C$。

第 140 语句将显示位置定于第二行第一列。

第 150 语句在当前显示位置显示一个“电”字。

第 160 语句将显示指针位置移到下一位置。

第 170 语句在当前显示位置上显示光标。

这种在应用程序层上建立汉字处理功能的方法有以下几个缺点:

(1) 由于汉字输入输出驱动模块是建立在应用程序这一层上，而不是建立在操作系统这一层上，也不是建立程序语言这一层上，因此，汉字的输入输出和原有的西文字符的输入输出不能统一到同一语句之中。这种用高级语言的外部接口命令、语句、函数调用汉字输入输出模块的方式很不自然，很不方便。例如，在 BASIC 程序中不能使用原有的输入输出语句（例如，INPUT，PRINT，READ，WRITE 等）输入输出汉字，只能使用 CALL 语句输入输出汉字，这就显得十分烦琐。

(2) 不能充分利用原系统中的软硬件资源，例如，原有的字处理软件就无法为汉字编辑服务，在汉字处理时使原系统效率大大降低。

(3) 用户界面比较复杂，很多汉字处理工作都要用户自己去做。在应用程序中调用汉字处理模块，用户需要了解汉字处理模块的有关细节，例如，汉字输入码、汉字字库、汉字地址码、汉字字形码、显示缓冲区、打印缓冲区等。

2. 在高级语言中增加汉字处理功能

到了八十年代初期，国内开始研究直接利用高级语言处理汉

字的问题。随着微型计算机的引入，过去在中、小型计算机上进行的汉字输入输出试验转移到微型计算机上来进行，并在 BASIC、FORTRAN、COBOL 等高级语言中扩充汉字处理功能。为了实现这一功能，主要是在高级语言中增加汉字输入输出语句，或利用原有的输入输出语句输入输出汉字，把原有语言改造为具有汉字处理功能的程序语言。这种在程序语言层上扩充汉字处理功能的方法，可以提高运行效率，简化应用程序的编制工作。

3. 操作系统的汉化

在一个计算机系统中，往往包含多种程序语言、数据库管理系统、通用应用软件，以及多个应用程序。如果要对它们逐个建立汉字处理功能，那么软件汉化工作量会非常庞大。

随着软件汉化工作的发展，人们逐渐认识到，计算机要能有效地实现汉字处理功能，必须要对计算机系统的软件核心——操作系统实行汉化。由于操作系统的汉化，把计算机系统公用的汉字输入输出功能集中在操作系统一次性建立，因此，各个高层软件通过系统功能调用便可使用操作系统汉字输入输出功能，从而避免了逐个建立汉字处理功能的重复性工作，使各高层软件的汉化工作变得容易。由此可见，在操作系统这一层上建立汉字处理功能，是整个计算机系统建立汉字处理功能的最简单、最有效、最切实可行的方法。

本章首先以 IBM PC 机上的 DOS，CP／M 和 PDP−11 机上的 RSX−11M 为例，来说明操作系统汉化的基本方法；然后将分别介绍汉字输入处理、汉字输出处理和汉字字库管理的基本方法。由于不同型号的计算机具有不同的体系结构，而且同一种型号的计算机上又可以有多种不同的操作系统，因此，操作系统的汉化因机而异，因操作系统而异，本章只能就操作系统汉化的基本原理作扼要介绍。

4. 中文操作系统的自动生成

面对层出不穷的汉字输入方案和品种繁多的打印机和显示器，要更换一种汉字输入输出方法，必须对中文操作系统进行修改，这使操作系统汉化的工作量负担很大。为了解决这种危机，目前出现了中文操作系统的自动生成的倾向。所谓中文操作系统的自动生成，是指中文操作系统提供汉字输入输出模块的自动生成方法，设计通用的汉字输入输出模块，适应各种汉字输入输出方法，使中文操作系统具有广泛的适应性，用户可以方便地配上自己所需要的各种汉字输入输出方法。这是一种理想化的目标。

目前，中文操作系统的自动生成，还仅限于一类汉字输入方法和一类汉字输出方法的自动生成。例如，键盘汉字输入方案的自动生成，一类打印机的汉字打印驱动程序的自动生成等等。本章将分别介绍汉字输入方案的自动生成方法和汉字打印驱动程序的自动生成方法。

10.1 操作系统汉化的基本方法

操作系统汉化是在计算机系统建立汉字处理功能的最广泛最有效的方法。

操作系统的汉化，实质上是把同一个操作系统移植到有相同的 CPU 但配有汉字外部设备的计算机系统中，通过移植操作系统来增加它的汉字处理功能。为了把汉字外部设备加入到计算机系统中，如果操作系统具有系统生成功能，则需要把用户提供的汉字外部设备的驱动模块安装到操作系统中；否则，移植的主要工作是重写输入输出处理程序。无论如何，只要不改变高层软件对操作系统的调用界面，那么，根据软件功能的传递性，中文操作系统的汉字处理功能就容易传递到它的高层软件，包括程序语言、数据库管理系统、应用软件等，因此，原操作系统支持的软件容易搬到中文操作系统上运行。也就是说，在操作系统中用汉字设备驱动模块来代替原有的西文设备驱动模块，并没有改变操

作系统和高层软件之间的界面，因此，对于用户来说，使用中文操作系统的方法和手段与使用西文操作系统的方法和手段一致；同样，在中文操作系统中使用程序语言、数据库管理系统或应用软件处理汉字的方法，与在西文操作系统中使用同一程序语言、数据库管理系统或应用软件处理西文字符的方法一致。由此可见，操作系统的汉化容易使整个计算机系统上所有软件资源都能处理汉字，达到中西文兼容。

操作系统的汉化要考虑以下三个关键问题：

(1) 根据需要选择适合的汉字设备，例如，汉字键盘、汉字显示器、汉字印刷机、汉字终端等；亦可根据需要对硬件进行适当的改造，例如，用汉卡来容纳汉字字库或实现汉字字符显示方式等。

(2) 在用汉字设备代替西文设备的同时，用汉字设备驱动模块代替西文设备驱动模块。仿照操作系统与原有设备驱动模块的连接方法，把汉字设备驱动模块有机地纳入操作系统中。汉字设备驱动模块不仅具有汉字处理功能，而且还保持原有设备驱动模块的西文字符处理功能，从而使操作系统实现中西文兼容。

(3) 在计算机中处理汉字，实质上就是处理汉字代码。汉字代码的选择要使操作系统容易识别和区分汉字与西文字符，便于实现各种汉字代码之间的转换，例如，汉字内部码，汉字输入码，汉字地址码，汉字字形码等。

不论是微型计算机，还是大、中、小型计算机，操作系统的汉化原则上均可采用纯软件方式、纯硬件方式或软件与硬件结合方式。但是，由于计算机系统的配置规模不同，建立汉字处理功能的复杂程度也不同，因此微型计算机上操作系统的汉化与大、中、小型计算机上操作系统的汉化方式有所不同。本节将分别介绍微型计算机上操作系统和大、中、小型计算机上操作系统汉化的基本方法。

10.1.1 微型计算机上操作系统的汉化

由于微型计算机系统规模小，操作系统改造较为容易，因此，微型计算机上操作系统的汉化常采用纯软件方式。为了提高汉字处理的效率，适应汉字处理的特点，也经常采用软件与硬件结合方式来实现微型计算机上操作系统的汉化。例如，用汉卡来容纳汉字字库，可节省内存空间，加快汉字存取速度；用汉卡来实现汉字字符显示方式，没有额外的开销，汉字显示效率高，速度快，并使程序运行环境不变，达到中西文软件的最大兼容性；用带有汉字字库的打印机打印汉字，可大大提高汉字输出的效率；用汉字终端可在多用户系统中有效地实现汉字输入输出功能和通信功能；等等。当然，除硬件改造外，还要对较件做相应的改造。

下面，以 IBM PC 机上的 DOS 和 CP／M 为例，来说明微型机上操作系统汉化的基本方法。

1. DOS 操作系统的汉化

DOS 操作系统采用分层次的模块结构，它由三个层次模块和一个引导程序构成。这三个模块由低层到高层依次是：输入输出系统，文件管理系统（IBMDOS.DOM），命令处理系统(COMMAND.COM)。其中，输入输出系统又由基本输入输出系统 BIOS 和基本输入输出模块 IBMBIO.COM 两部分组成。BIOS 中含有若干个设备驱动程序，用于直接控制系统的外部设备。由于它被固化在主机板上的 ROM 芯片里，故亦称为 ROM BIOS。IBMBIO.COM 是 BIOS 的接口模块，它提供与 ROM BIOS 设备驱动程序的低级接口。IBMDOS.COM 提供与高层软件的高级接口，它是由文件管理程序、磁盘操作程序以及可供高层软件调用的系统功能程序。COMMAND.COM 主要用于解释并执行命令，命令包括内部命令、外部命令，批命令三类。

DOS 的启动过程是：当启动系统时（冷启动或热启动），首

先由 ROM 中的引导程序把系统盘上的系统引导程序引入内存，此后该程序运行，把系统盘上的 IBMBIO.COM 和 IBMDOS.COM 两个文件依次引入内存；然后，分别执行这两个文件内的初始化程序进行初始化工作；最后，由 IBMBIO.COM 把系统盘上的 COMMAND.COM 文件装入内存的指定区域，并把控制转给 COMMAND.COM。

BIOS 中的设备驱动模块均是以软中断指令的形式提供给高层软件使用的。中断指令的形式是:

INT n

n 是中断号。例如，与 DOS 汉化有关的软中断主要有:

INT 16H 键盘驱动程序

INT 10H 显示器驱动程序

INT 17H 打印机驱动程序

INT 5H 屏幕打印驱动程序

当高层软件使用外部设备输入输出时，总是通过软中断调用 ROM BIOS 中有关的设备驱动程序。软中断的响应，则是根据被调用的软中断的中断号，在中断向量表中找到相应设备驱动程序的入口地址，然后转去执行该设备驱动程序（IBM PC 机的中断向量表放在 PC 机内存最低地址的 256 字节，即 0000:0000–0000:0100。中断向量表中含有按中断号顺序依次排列的相应的中断处理程序的入口地址，各占四个字节）。

CCDOS 是我国第一个由 DOS 汉化而成的中文操作系统。CCDOS 对 DOS 的汉化，主要是在 DOS 基础上，对其中的基本输入输出系统 BIOS 和文件管理系统 IBMDOS.COM 扩充汉字处理功能。为了增加汉字输入输出功能，响应对汉字外部设备的调用要求，必须利用 ROM BIOS 的接口，在 ROM BIOS 之外扩充用于汉字输入输出的键盘管理模块、显示管理模块、打印管理模块和屏幕打印驱动模块，从而把 ROM BIOS 扩充为 CCBIOS。CCBIOS 是 CCDOS 中直接控制汉字外部设备的核

心部分。而对 IBMDOS.COM 部分的修改只限于一些字符处理程序和引导程序的适配，但这也是汉字内码能在操作系统上通行的关键之处。

CCDOS 的启动过程就是把 DOS 改造为 CCDOS 的过程。启动过程是：先启动 DOS。当 DOS 启动成功后，将自动执行批命令文件 AUTOEXEC.BAT。AUTOEXEC.BAT 的内容是：

```
ECHO OFF
CLS
FILE1
CCCC
ECHO ON
```

ECHO OFF 关闭显示；CLS 清屏；FILE1 检查汉字字库（即 CCLIB 文件）在系统盘上的完好性，为汉字字库申请好内存空间，并进行初始处理和模式切换；CCCC 装入扩充的汉字处理模块和汉字字库；并修改有关的中断指针，使中断 5H，10H，16H，17H 的指针分别指向 CCBIOS 的屏幕打印驱动模块、显示管理模块、键盘管理模块和打印管理模块，从而形成了 CCDOS。ECHO ON 打开显示。

由此可见，CCDOS 的汉化工作主要是构造 CCBIOS 的汉字输入输出模块。键盘管理模块的主要功能是把从键盘上输入的汉字输入码转换为汉字内码（见 10.2 节）。显示管理模块的主要功能是把汉字内码转换为汉字字形信息，并送屏幕显示出来；此外，还完成屏幕光标定位、从屏幕上直接读取汉字、屏幕滚动、提示行显示处理等功能（见 10.3 节）。打印管理模块的主要功能是把汉字内码转换为汉字字形信息，并送往打印机打印出来（见 10.3 和 10.4 节）。屏幕打印驱动模块的主要功能是把彩色／图形显示适配器上的显示缓冲区中的屏幕信息按图形方式从打印机上打印出来。

由于 CCDOS 是在 DOS 基础上扩充而成的，因此，CCDOS 的内存分配必须考虑 DOS 的原内存分配情况。装入内存的 CCDOS 分为程序区与数据区两部分。程序区主要是 CCBIOS，即汉字设备驱动模块，用于代替原来位于 ROM 中的设备驱动程序。数据区中包含汉字字库和汉字输入码到汉字内码的转换表等。为了保证原 DOS 的工作不受影响，而且使 CCDOS 不被其它程序覆盖，只能把 CCDOS 置于原用户区的底层。CCDOS 内存分配情况如下:

0000:0000 ROM BIOS的中断向量表(包括CCBIOS的各汉字处理模块的中断入口地址)

0040:0000 ROM BIOS数据区

0050:0000 DOS与ROM BIOS的通讯区

0060:0000 DOS BIOS程序区(IBMBIO.COM)

00BF:0000 DOS程序区(IBMDOS.COM和COMMAND.COM的常驻部分)

xxxx:0000 用户程序区(.COM或.EXE文件)

xxxx:0000 DOS命令解释程序覆盖区(COMMAND.COM的暂存部分)

xxxx:0000 CCDOS程序区(各汉字处理模块)

xxxx:0000 CCDOS数据区

xxxx:0000 汉字字库

xxxx:0000 未安装的RAM

B000:0000 单色显示缓冲区

B800:0000 彩色显示缓冲区

C000:0000 未安装的ROM

F600:0000 ROM BASIC区

FE00:0000 ROM BIOS区

由以上可知，从 DOS 形成 CCDOS 的关键是把 ROM BIOS 扩充为 CCBIOS，为此，在启动 CCDOS 时，只需把 CCBIOS 中的模块引入内存，然后修改相应的中断指针，使其

指向 CCBIOS 中的相应模块。以后在 CCDOS 控制下，通过软中断调用的就不再是 ROM BIOS 中的相应模块，而是驻在内存的 CCBIOS 中的相应模块。例如，在 DOS 下输入一个西文字符与在 CCDOS 下输入一个西文字符或一个汉字，虽然都是使用 INT 16H，但所执行的程序是不一样的。在 DOS 下，由于中断向量表中 16 号中断指针指向 ROM BIOS 中的键盘驱动程序，因此，当用户程序中出现 INT 16 时，去调用并执行 ROM BIOS 中的键盘驱动程序，完成一个西文字符的输入。而在 CCDOS 中，由于中断向量表中 16 号中断指针，在 CCDOS 启动时已被修改为指向驻在内存的 CCBIOS 中的键盘管理模块。这时，若输入为西文方式，则仍由 ROM BIOS 中的键盘驱动程序来完成。若输入为中文方式，则每当从键盘键入一个字符时，键盘管理模块首先判别它是否汉字输入码字符，若是，则将该字符送入缓冲区，继续接收后面的字符，直到该汉字输入码的字符全部接收完毕，再根据汉字输入码的类型，按相应的方法查找汉字输入码到汉字内码的转换表，实现汉字输入码到汉字内码的转换，从而最终完成一个汉字的输入。

CCDOS 对 DOS 的汉化，采用的是软字库方式，即把汉字字库驻留在主存储器中，每当启动 CCDOS 时，就把系统盘上的汉字字库加载到内存的汉字字库区域中去。软字库实现比较简单，不需要对硬件作任何改动，是一种灵活而又经济的汉字字库建立方式。但是，由于汉字字库容量很大，加载汉字字库需要花费一定的时间，而且用户可使用的有效内存空间也大为减小。

鉴于上述原因，有些中文操作系统对 DOS 的汉化采用了硬字库方式，即用 EPROM 或 Mask-ROM 芯片制成汉字字库卡(亦称汉卡或中文卡)，把它插入机器的扩展槽中，用以代替软字库。由于硬字库容量较大，为了不占用主存空间，常常把它安排在另一个存储空间内。通常，硬字库采用 EPROM 电路写入汉字字库。ROM 芯片的选择是制作汉卡的关键。在半导体电路制

作过程中，直接将汉字点阵做在 ROM 芯片内的制品称为 Mask-ROM，它比 EPROM 价格低，适合于大量应用的场合。汉卡的电路结构比较简单，其功能仅是为机器扩充一个 ROM 存储区，汉卡的控制电路用来完成 I／O 扩充总线接口逻辑、控制寻址、读出和变换。汉卡上还有 ROM 字库存储体、地址译码驱动和读出缓冲电路等。采用硬字库方式的中文操作系统对 DOS 的汉化，与 CCDOS 对 DOS 的汉化方法相似，只是 CCDOS 从主存储器的软字库中存取汉字字形信息，而采用硬字库方式的中文操作系统从汉卡的硬字库中存取汉字字形信息。

CCDOS 采用图形显示方式来显示汉字，因此它显示汉字的能力取决于原机器的图形显示能力，受显示器分辨率和彩色／图形适配器上显示缓冲区的容量限制。由于屏幕上显示一行 16×16 点阵汉字需占用 16 行扫描线，加上行间距占两行，共占 18 行；因此，对于每屏垂直分辨率为 200 点的图形显示，每屏只能显示 11 行汉字。这种显示能力很不理想，不仅是因为每屏显示的信息量太少，更重要的是因为，每屏只能显示 11 行汉字，无法保持中西文显示环境的一致性，致使西文软件汉化工作量增大，与屏幕显示行数有关的成分都需要汉化，而且汉化后的中文软件无法保持西文软件的原有的显示功能，尤其是数据表软件，除标题和栏目以外，数据表的有效行数所剩无几，使原西文软件面目皆非。对于这个问题，目前有以下两种解决办法：

(1) 把原彩色／图形适配器改为高分辨率显示适配器，为了保持西文软件原有的显示行数和每行的显示字数，分辨率至少应为 640×400 (25 行，每行 40 个 16×16 点阵汉字，无行间距)。这些高分辨率显示适配器的主要做法是把显示缓冲区由原来 16KB 扩充到 64KB 甚至 128KB，再配上高分辨率显示器。例如，长城 0520C 使用了分辨率为 640×450、有八种颜色的彩色／图形适配器，并配之以高分辨率彩色显示器，实现了每屏 28 行×40 字的汉字显示能力，其中 25 行为正文区，3 行为提示

区，因而汉字显示性能大为改善。

(2) 采用字符显示方式显示汉字。由于原来的单色适配器只能驱动显示器以字符显示方式显示西文字符，而彩色／图形适配器只能驱动显示器以字符显示方式显示西文字符和以图形显示方式显示汉字和图形，但无法驱动显示器以字符显示方式显示汉字，因此需要另加硬件线路来实现汉字的字符显示方式，用以构造汉字字形发生器和显示缓冲区。用于汉字字符显示方式的显示缓冲区中存放的是汉字地址码。汉字字形发生器用于把显示缓冲区中的汉字地址码转换为汉字字形信息，并在屏幕上显示出来。这种汉字字符显示方式，亦称为硬汉字显示方式。例如，联想式汉字系统就是采用联想式汉卡来实现汉字字符显示方式的。这种硬汉字显示方式在显示缓冲区中存储的是汉字地址码而不是汉字字形信息，因此节省存储器，也易于修改屏幕信息；而且由于是用硬件（汉字字形发生器）来实现汉字显示，汉字显示速度快。此外，由于这是一种接插兼容技术，因此不必修改 DOS 原有的显示驱动程序，使原西文软件的显示环境不变，从而对 DOS 的原有高层软件在屏幕显示方面基本上无需汉化，从而在屏幕显示方面达到中西文软件最大限度的兼容。

总之，对于 DOS 操作系统汉化，由于硬件设备配置不同，汉字字库构造方式不同，汉字显示方式不同，汉化方法也不尽相同。因此，产生了许多种对 DOS 汉化的中文操作系统。然而，万变不离其宗，它们的设计思想都源于 CCDOS。

2. CP／M 操作系统的汉化

CP／M 操作系统采用分层次的模块结构。它所包含的模块由低层到高层依次是：BIOS（基本输入输出系统），BDOS（基本磁盘操作系统），CCP（控制台命令处理程序）。只允许高层模块单向调用低层模块。BIOS 是 CP／M 的输入输出管理程序，是操作系统与硬件的接口，用来直接驱动各种外部设备。BDOS 是 CP／M 的基本控制部分，主要功能是进行磁盘文件管

理，它提供了若干条系统调用作为对高层软件的支持，高层软件可以方便地使用系统给出的系统功能调用方式来调用系统资源。CCP 是 CP／M 的命令解释程序，它负责装入 TPA 程序（系统软件或应用软件），程序装入后，CCP 可被 TPA 程序覆盖，执行结束后，重新从磁盘装入 CCP，并把控制转给 CCP。

在 CP／M 操作系统中，高层软件的每一个输入输出要求，都是通过系统功能调用来实现的，BDOS 模块分析这些系统要求，然后调用 BIOS 模块来完成这些要求。再由 BIOS 模块直接测试和启动外部设备。此后，BIOS 模块将控制返回到 BDOS 模块。当高层软件提出的系统功能调用全部完成后，再将控制返回到高层软件。

CP／M 操作系统提供的系统功能调用具有统一的调用方式。在 5、6、7 号单元中各放置一条转移指令，用于转向 BDOS 模块在内存中的起始位址。不同的系统由于硬件配置不一样，这个起始地址也各不一样。但是不论什么配置下的 CP／M 操作系统中的任何一个 TPA 程序，只要调用 5 号单元就可转入 BDOS 模块的入口地址。此外，还约定：系统功能调用序号存入寄存器 C，入口参数存入寄存器 D 和 E，返回参数存入寄存器 A 和 B。例如，第 10 号系统功能调用是输出字符串，它的系统功能调用序号：（C）＝09，入口参数：（DE）＝输出字符串的首地址，返回参数：无。

CP／M 给出的系统功能调用的手段和方式，使系统具有可扩充性。CP／M 还允许用户自定义外部设备，只要在 BIOS 中有相应的外部设备驱动模块即可；因此，只需改写 BIOS 便可方便地把系统移植到其它硬件配置上。CP／M 的上述特点以及 CP／M 的层次结构，为 CP／M 操作系统的汉化奠定了基础，使 CP／M 操作系统的汉化较为容易。

CP／M 操作系统的汉化，首先要选择汉字印刷机、汉字显示器、标准西文键盘，然后在 CP／M 的 BIOS 中增加以下三个

汉字驱动模块:

(1) 汉字印刷模块;

(2) 汉字显示模块;

(3) 汉字输入模块。

汉字输入模块是交互式的，既有输入又有输出，输入时涉及汉字输入码到汉字内码的转换，输出时调用汉字显示模块。

把上述汉字输入输出驱动模块纳入 CP／M 操作系统的输入输出管理程序 BIOS 中，使 BIOS 扩充为把汉字的输入输出和西文符的输入输出统一考虑的输入输出管理程序 XBIOS，从而使 CP／M 成为中西文兼容的操作系统。

这样，在中文 CP／M 操作系统的控制下，各种翻译程序本身及其由编译程序产生的目标程序都能使用和处理汉字。这是因为，它们都是 TPA 程序，它们对汉字和西文字符的输入输出要求都是以系统功能调用的方式向 BDOS 模块提出的。由于汉字代码与西文字符编码在计算机内部表示的一致性，又由于把汉字纳入程序语言的字符串中，因此 BDOS 模块对汉字和西文字符的处理是一视同仁的。只是到了 BDOS 调用 XBIOS 中的汉字输入输出驱动模块时，由于 XBIOS 中的汉字输入输出驱动模块能够识别和区分汉字和西文字符，才分别输入输出汉字和西文字符。

例如，用户要求输入字符串“啊 A1”。则高层软件中要用一个相应的系统功能调用，发向 BDOS 模块，BDOS 责成 XBIOS 执行输入这一字符串的操作。用户输入这一字符串的过程及 XBIOS 模块相应的工作过程大致如下:

(1) 用户按 Alt−F2 键。XBIOS 置某种汉字输入标识位，例如，拼音输入方案。

(2) 用户键入汉字输入码。XBIOS 根据汉字输入码从相应的汉字编码转换表中得到“啊”字的汉字地址码，并从汉字字库中取出“啊”字的汉字字形信息，在屏幕上显示出“啊”字；同时，把

汉字输入码转换为汉字内码，并送入缓冲区中。

(3) 用户按 Alt−F6 键。XBIOS 置西文字符输入标识位，例如，ASCII 输入方式。

(4) 用户键入英文字母 A。XBIOS 在屏幕上显示“A”，并将 ASCII 码 41 送入缓冲区。

(5) 用户键入数字 1，XBIOS 在屏幕上显示“1”，并将 ASCII 码 31 送入缓冲区。

(6) 用户按 Enter 键。XBIOS 本次输入处理结束。

10.1.2 大、中、小型计算机上操作系统的汉化

由于大、中、小型计算机系统比微型计算机系统复杂得多，用纯软件方式对操作系统进行汉化工作量太大，因此，大、中、小型计算机上操作系统的汉化，应尽量少修改操作系统本身，尽量采用硬件方式，即尽量采用接插兼容技术。

利用接插兼容技术，可使汉字终端和汉字外部设备通过联机仿真软件及接口同主机相连接，汉字处理功能在接插兼容的汉字终端和接插兼容的汉字外部设备上实现，或者汉字外部设备也接在汉字终端上，完全由接插兼容的汉字终端来完成汉字处理任务。由于接插兼容的汉字终端自带汉字字库和汉字编码转换表，接插兼容的汉字印刷机自带汉字字库，并能区分汉字和西文字符，因此主机不需要区分汉字和西文字符，只是把汉字作为普通字符来处理。显然，在计算机系统中，用接插兼容的汉字终端和汉字外部设备替换西文终端和西文外部设备，使得原则上不修改操作系统的设备驱动模块，便能输入输出汉字，就象输入输出西文字符一样.

利用接插兼容技术对操作系统进行汉化，有时还需要对操作系统本身作必要的修改。而且有些软件功能固化在 ROM 上会丧失灵活性，例如，汉字输入方案层出不穷，把汉字输入码到汉字内码的转换表固化在 ROM 上有时是不适当的。因此，大、

中、小型计算机上操作系统的汉化采用纯硬件方式有时是不方便的，在这种情况下，常采用软件与硬件结合方式，当然要尽量减少软件工作量，尽量少修改操作系统本身。由于大、中、小型计算机上一般均具有系统生成功能，利用这一功能，容易把用户提供的外部设备驱动模块加到操作系统中去，因此，通过把汉字终端和汉字外部设备的驱动模块加入操作系统中去，便可把相应的汉字终端和汉字外部设备加入到计算机系统中。

下面，以 PDP-11 系列机上的 RSX-11M 为例，来说明大、中、小型计算机上操作系统汉化的基本方法。

PDP-11 系列机有十多种操作系统，其中尤以 RSX-11M 使用最为广泛。RSX-11M 是一个实时、分时操作系统。RSX-11M 操作系统的输入输出过程大致如下:

当高层软件要求输入输出时，必须通过 QIO 宏指令才能进入管理程序。管理程序中的输入输出子程序接受用户程序的委托，对 QIO 宏指令中表述的输入输出要求大体上作如下处理:

(1) 对 QIO 宏指令进行预处理。根据 QIO 宏指令中提供的参数，来识别是哪个设备的哪种要求。把这种要求填写到专门的“有输入输出要求的报文”中，并把报文发送给相应的设备驱动程序，在那里排队挂号。

(2) 把控制转给相应的设备驱动程序。

(3) 当相应的设备驱动程序得到控制权时，取出最早排队挂号的“有输入输出要求的报文”进行处理，完成用户的输入输出要求。在这个过程中，对某些设备驱动程序，也许要处理若干次的输入输出中断，才能完成某一用户的输入输出要求。输入输出要求完成后，控制又转向原先的管理程序的输入输出子程序。

(4) 管理程序的输入输出子程序对用户的输入输出要求进行善后处理，把完成的情况（成功或失败）告诉高层软件，并把控制转向高层软件中的 QIO 宏指令的下一指令。

RSX-11M 操作系统的汉化，就是要把用户提供的汉字设备

加到系统配置中去，把用户提供的汉字设备驱动程序加到操作系统中去，并与原系统保持一致。

用户提供的外部设备不外乎两类：汉字终端和汉字印刷机。有以下三种形式的汉字终端：

(1) 最简单的汉字终端不具备汉字字库，字库 ROM 可设置在 PDP-11 的总线上，作为 PDP-11 的一个外部设备。对于这样的汉字终端，汉字终端驱动模块中要有汉字输入码到汉字内码的转换表。由于 RSX-11M 是具有分时功能的操作系统，汉字字库做在总线上可为多用户共享，因此成本低；其缺点是系统开销太大，而且对于 16×16 点阵的汉字字模，要在显示器上显示一个汉字就需要输出 32 个字节，输入输出量很大，致使系统效率较低。

(2) 另一种汉字终端是把汉字字库固化在汉字终端内，而把汉字输入码到汉字内码的转换表设置在汉字终端驱动模块中。这种汉字终端只要发送 2 个字节就可显示一个汉字，系统开销也不大。

(3) 如果把汉字字库和汉字输入码到汉字内码的转换表都固化在汉字终端内，则可大大节省系统开销，但由于转换表固定在 ROM 上，不如上述两种汉字终端灵活，因为在上述汉字终端驱动模块中设置汉字输入码到汉字内码的转换表是由软件实现的，可由用户选择自己喜欢的汉字输入方案。

下面，仅以第一种汉字终端为例，来说明用户从汉字终端输入汉字的过程，如下：

(1) 高层软件发出一个 QIO 宏命令：从汉字终端输入一汉字串，并把这一串汉字的汉字内码存放在高层软件指定的地址(QIO 宏指令的具体功能是由该宏指令的参数表示的)。

(2) 管理程序的输入输出子程序将上述要求转达给汉字终端驱动模块。

(3) 汉字终端驱动模块启动汉字终端，进行如下一系列工

作:

①用户键入汉字输入码;

②汉字终端驱动模块把汉字输入码转换为汉字内码，并根据汉字内码从汉字字库中找出相应的汉字字形信息，按字节逐个发送到汉字显示器上，于是显示出用户所需要的汉字;

③用户继续键入汉字输入码;

④汉字终端驱动模块重复②的工作;

⑤用户按 RETURN 键，表示汉字串输入完毕。

⑥汉字终端驱动模块完成了用户委托的输入输出要求，把控制转向管理程序中的输入输出子程序。

(4) 管理程序中的输入输出子程序把一串汉字内码传送到高层软件指定的地址，并把控制转向高层软件的 QIO 指令的下一指令。

由于 RSX−11M 操作系统具有系统生成功能，允许把用户提供的外部设备驱动模块加到操作系统中去，因此，不论什么样的汉字终端、不论什么样的汉字印刷机都可加入系统。另外，该系统有较好的“与设备无关”的性能。由于系统程序和公用程序中所用到的输入输出设备不是在程序中预先规定的，因而可以在程序执行时或程序连接时指定设备。由于有了这个功能，就可以用原操作系统中所有的翻译程序和公用程序来处理汉字。也就是说，用汉字终端驱动模块代替原有的西文终端驱动模块。

10.2 汉字输入处理

汉字输入方式有: 主控制台输入方式、媒体记录输入方式、联机终端输入方式。

汉字输入设备有: 汉字整字键盘、笔触式汉字字盘、中文打字机式汉字键盘、汉字字根键盘、标准字母数字键盘等 (见 2.4.1 节)。

汉字输入编码方案有上千种（见 3.1.2 节）。

汉字输入处理就是要在不同的汉字输入方式下，有效地管理各类汉字输入设备，并把各种汉字输入码（见 2.3.2 节）转换为统一的汉字内部码（见 2.3.3 节）。汉字输入处理功能是由汉字输入程序实现的。

本节首先介绍汉字输入的三种方式，然后给出汉字输入程序的设计原理，最后介绍实现汉字输入码到汉字内码转换的方法。

10.2.1 汉字输入方式

1. 主控制台输入方式

对于单机系统，采用的就是这种输入方式。对于联机系统，采用这种输入方式虽然较为简单直观，但在输入过程中，由于人工按键速度慢，过多地占用主机时间，因此很不经济，只适合输入少量汉字。

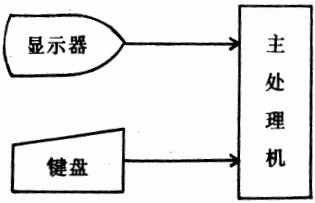
2. 媒体记录输入方式

这种输入方式首先用脱机方式将汉字数据记录在软盘、盒式磁带、纸带、卡片等信息媒体上，然后把记录有汉字数据的媒体安置在联机的汉字输入设备上，在汉字输入程序的控制下，将记录的汉字数据高速读入主机。这种输入方式占用主机时间少，便于修改，适合于大量数据的输入。

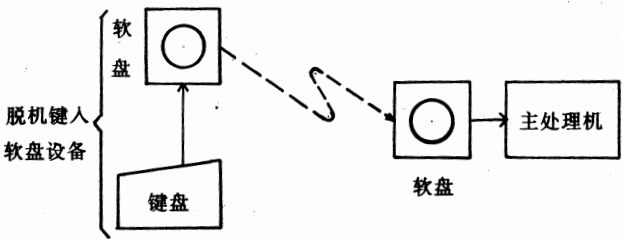
3. 联机终端输入方式

这种输入方式通过汉字终端输入汉字，并通过通信接口将汉字信息送至主机加工处理。主机系统分时或实时地响应各终端的汉字输入操作，并将各个终端输入的汉字信息分别存入各自的输入缓冲区。主机除了响应各个终端输入汉字信息外，若有空余时间，还可运行其它程序。因此，这种输入方式具有主控制台输入方式的优点，既可随时输入汉字，又可随时增删改这些信息；同时，主机时间也能得到充分合理的利用。

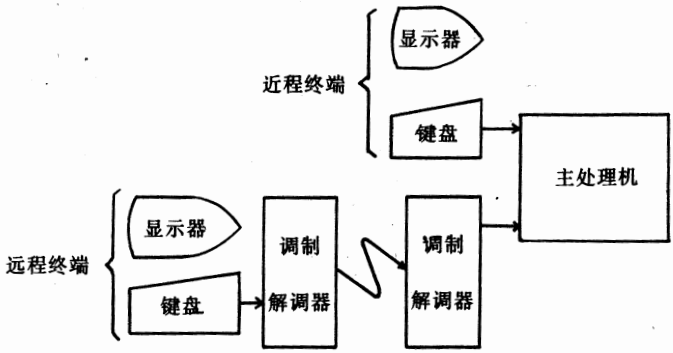
上述这三种输入方式的示意图如图 10.1 所示。



(1)主控制台输入方式



(2)媒体记录输入方式



(3)联机终端输入方式

图 10.1　汉字输入方式示意图

10.2.2　汉字输入程序的设计原理

汉字输入程序用于实现汉字输入处理功能。

由于汉字输入设备的种类不同，汉字输入编码方案也有多种，因此，汉字输入程序的设计要面向不同的汉字输入设备和多种汉字输入编码方案，以满足各类用户输入汉字的不同需要。

当用户程序请求输入汉字时，汉字输入程序首先将用户程序提供的各种必要的参数赋予操作系统的输入控制，输入控制在汉字输入过程中不断地测试汉字输入设备所处的状态，协调主机与输入设备在时间上的差异，同时把输入设备送至主机的信息转换成主机相容的格式，输入控制返回时，送出汉字输入程序所需要的结果参数。

每当从输入信息流中读取一个字符时，汉字输入程序判断输入字符是汉字输入码，还是西文字符，还是功能控制码（见2.3.6节），分别予以不同的处理。若是汉字输入码，则转入汉字输入码处理程序，并将输入的字符送入汉字信息缓冲区，继续接收后面的字符。当一个汉字的全部输入码输入完毕后，再根据汉字输入编码方案的类型，检查缓冲区中读入的汉字信息的正确性。若有错误，则将出错信息反馈给用户。

往往在一个中文操作系统中使用多种不同形式的汉字输入编码方案。这些汉字输入码输入计算机后必须转换为统一的汉字内码，才便于在计算机内实现汉字的运算、比较、排序、查找、显示，打印等操作。有的系统为了处理方便，先把汉字输入码转换为汉字地址码，用于在汉字字库中找到相应的汉字字形信息，在屏幕上显示出来，然后再根据汉字内码与汉字地址码的某种映射关系，求得汉字内码。

10.2.3 实现汉字输入码到汉字内码转换的方法

对于不同的汉字输入编码方案，实现汉字输入码到汉字内码(或汉字地址码）转换的方法也不同。转换方法大致可分为计算法和查表法两大类。

1. 计算法

对于有计算规则可循的汉字输入编码方案，通常采用计算法来实现汉字输入码到汉字内码（或汉字地址码）的转换。

例如，国标码和区位码是计算类汉字编码，其汉字输入码到汉字内码（或汉字地址码）的转换是通过计算来完成的（见3.1.2节5.）。

国标码转换为内码只需将国标码两个字节的最高位都置1，即国标码十六进制数加8080。

区位码转换为内码时，先将区位码转换为国标码，然后再将国标码转换为内码。区位码转换为国标码的方法是：先把十进制的区码和位码分别转换成十六进制的区码和位码，然后再将十六进制的区码和位码分别加上十六进制数20，合并后就得到国标码。计算公式如下：

国标码＝((区码／16)H+20H)((位码／16)H+20H)

第一字节　　第二字节

2. 查表法

对于无计算规则可循的汉字输入编码方案，则必须采用查表法来实现汉字输入码到汉字内码（或汉字地址码）的转换。为此，对于每种汉字输入编码方案，均要建立一个汉字输入码与汉字内码（或汉字地址码）的对照表。编制对照表，首先要确定汉字输入码与汉字内码（或汉字地址码）之间的映象关系，然后选择一种适合这种映象关系的查找算法。

对于无重码类输入编码方案，例如，电报码，电信码等，常采用顺序查找法、折半查找法、散列查找法来查找对照表。对于有重码类输入编码方案，例如，拼音输入法，常采用分级查找法与顺序查找法、折半查找法、散列查找法相结合的方法来查找对照表。下面，简要介绍这四种查找方法。

(1) 顺序查找法

顺序查找法按对照表中项目的自然排列顺序依次逐一进行比

较，直至找到要找的汉字输入码为止。

设 n 为对照表的长度，则平均查找次数是:

$$M=\frac{1}{n}\sum_{i=1}^{n}i=\frac{1}{n}(1+2+\cdots+n)=\frac{n+1}{2}$$

这种查找法对于表中项目不多的查找是简单有效的，但当表中项目很多时查找效率较低。

(2) 折半查找法

折半查找法又称为对分查找法或二元查找法。在使用这种方法查表之前，应事先把对照表中的项目按汉字输入码的某种顺序(例如，字母序）排序。折半查找就象查英文字典一样，首先从中间一项开始查找汉字输入码，如果未查到，则判断要找的汉字输入码是在对照表的上半部还是下半部，如果在上半部（下半部)，则到上半部（下半部）用类似的方法查找，依次类推，直至找到要找的汉字输入码为止。

设 n 为对照表的长度，则平均查找次数是:

$$N=\frac{\sum_{i=1}^{n}i\cdot 2^{i-1}}{2^{n}}=n-1+\frac{1}{2^{n}}\approx\log_2 n-1$$

这种查找法比顺序查找法快得多，适合于对照表很大的查找。

(3) 散列查找法

散列查找法又称杂凑查找法。它是一种非比较查找法，不是通过对汉字输入码的一系列比较，而是通过对汉字输入码执行某种运算来直接确定它在对照表中的位置。

设对照表的长度为 n，x_i 为汉字输入码，α_i 为它在表中的地址，$1\leqslant i\leqslant n$，则汉字输入码 x_i 与地址 α_i 之间存在着一种函数关系（用 H 表示)，使得 $H(x_i)=\alpha_i(i=1,\ 2,\ \cdots,\ n)$成立。$H(x_i)$

称为散列函数。

通过散列函数，可直接把汉字输入码映射到表中的某一地址；但是，不同的汉字输入码有可能映射到同一地址上，会发生冲突。因此，用散列查找法实现汉字输入码到汉字内码转换的关键问题是构造散列函数和解决冲突。下面，简要说明这两个问题：

①构造散列函数

构造散列函数应当做到：适合于给定的汉字输入编码方案，算法简单，能把汉字输入码均匀地散列于表中允许的地址范围内，尽量避免发生冲突。

如果汉字输入码是非数值编码，则在不丢失有用信息的原则下，应首先转换成能够计算的数值码。例如，对于码元为英文字母的汉字输入码，可用 01–26 表示 26 个英文字母，则汉字输入码 BING 可转换为 02091407。

构造散列函数的方法有直接定址法、除留余数法、移位法和折叠法等。下面，以常用的除留余数法为例，说明如何构造散列函数。

除留余数法是指把汉字输入码的数值除以一个常数，所得的余数作为散列函数的值。这一常数的取值对散列地址的分布有很大影响，一般把它取为一个小于但又接近于可用地址的质数。只要选择的质数不与组成汉字输入码的基数相关，就能得到较均匀的地址分布。

例如，假设可用地址为 1000，且取小于但又接近于可用地址的质数为 997，则散列函数的值等于汉字输入码除以 997 所得的余数。通过该散列函数，可把汉字输入码映射为表中的相对地址，如图 10.2 所示。

②解决冲突

解决冲突的常用方法是链地址法，即用链表来表示溢出表的结构。构造链表的方法是：将某一汉字输入码经过散列函数转换

得到的每个散列地址建立一个链表。若出现不同汉字输入码映射为同一散列地址的情况，则在基本表的链域中存放同一散列地址在溢出表的头指针；否则，在基本表的链域中填 0。

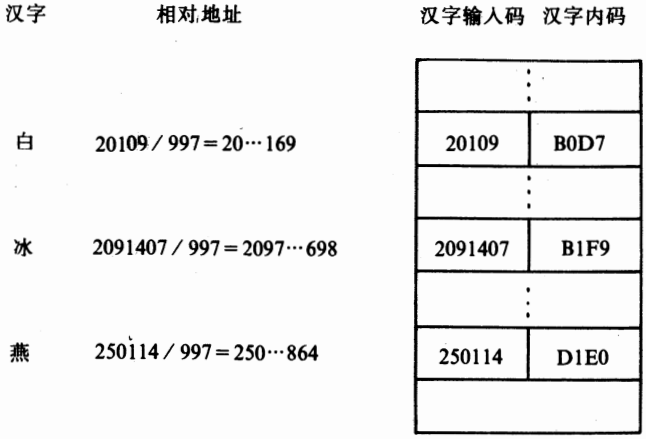


图 10.2　计算散列地址

例如，假设有三个汉字，它们的汉字输入码分别为 325，15280，48181，则使用上例中的散列函数计算得到的相对地址分别如下：

325 / 997 = 0…325

15280 / 997 = 15…325

48181 / 997 = 48…325

得到的相对地址均为 325。

用链表法解决这一冲突的散列表如图 10.3 所示。

查找散列表的方法是：对于要查找的汉字输入码，通过散列函数计算求得它在基本表中的地址，若在该地址找不到它，则再沿着链域继续查找溢出表，直至找到该汉字输入码为止。

散列查找法是一种直接计算地址的方法，因此比顺序查找法、折半查找法的效率高。然而，在实际使用中，很难解决冲

突，而查找效率又与解决冲突的方法有很大关系。

(4) 分级查找法

对于有重码类汉字输入编码方案，例如，拼音输入法，必须解决重码问题。常用的解决办法是采用人机对话式输入方法，在提示区上显示多个重码汉字供用户选择，当用户选择后，就把选中的汉字显示在正文区。鉴于上述原因，有重码类汉字输入方案常采用分级查找与顺序查找法、折半查找法、散列查找法相结合的方法，来查找汉字输入码与汉字内码（或汉字地址码）对照表。

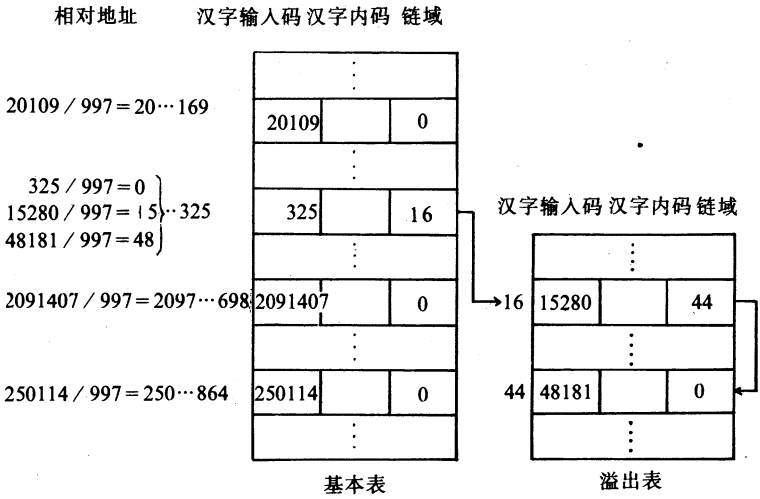


图 10.3 用键表法解决冲突

下面，以 GB2312 一级汉字的拼音输入法为例，来说明分级查找法。

由于一级汉字是按汉语拼音字母顺序排序的，而拼音输入法是把构成字音的拼音字母当作拼音输入码来输入，因此，可把一级汉字按字音分为若干个拼音重码组。

拼音输入法的对照表分为两级。一级表是拼音输入码与拼音重码组指针对照表。每个拼音输入码各代表一个拼音重码组的一组汉字。由于拼音输入码是按汉语拼音字母序排列的，因此可采用顺序查找法、折半查找或散列查找法来查找一级表。二级表中存放各拼音重码组中汉字的内部码。一级表中的拼音重码组指针指向相应拼音输入码所代表的拼音重码组的汉字内码在二级表中的首地址。

当按拼音输入法输入汉字时，首先根据用户键入的拼音输入码查找一级表，得到该拼音输入码所代表的拼音重码组的汉字内码在二级表中的首地址。当用户选择提示区中的重码汉字时，通过键入重码汉字前面的代表字符，确定了选中的汉字在该拼音重码组中的序号，从而根据该拼音重码组的汉字内码在二级表中的首地址和选中的汉字在该汉字重码组中的序号，从二级表中得到选中的汉字的汉字内码，至此实现了拼音输入码到汉字内码（或汉字地址码）的转换（见图 10.4）。

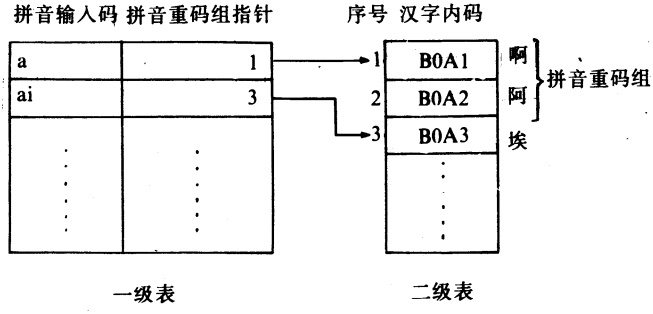


图 10.4　拼音输入法的对照表

10.3　汉字输出处理

汉字输出设备有：汉字显示器、针式汉字打印机、激光汉字

印刷机、喷墨式汉字印刷机、热感式汉字印刷机、静电式汉字印刷机、光纤管转印汉字印刷机等（见 2.5 节）。

汉字字形输出按汉字字形的存储方法可分为汉字整字字形输出和压缩信息的汉字字形输出（见 10.3 节 3.），按汉字字模可分为多种点阵多种字体的汉字字形输入（见 3.2.1 节）。

汉字输出处理就是要有效地管理各类输出设备，并把汉字内码转换为汉字地址码，根据汉字地址码从相应的汉字字库中找到相应的汉字字形码，再把汉字字形码转换为输出设备规定的信息格式，最后在输出设备上输出汉字。汉字输出功能是由汉字输出程序实现的。

本节首先介绍汉字输出程序的设计原理，然后简要介绍显示器汉字输出程序和针式打印机汉字输出程序的基本设计思想。

10.3.1 汉字输出程序的设计原理

汉字输出程序用于实现汉字输出处理功能。

由于汉字输出设备的种类繁多，而且要求输出的字形也不同，因此汉字输出程序的设计方法要因输出设备和输出字形而不同，以便使用户获得满意的输出结果。例如，即使都是针式打印机，不同型号所配置的打印驱动程序也有所不同。

当用户请求输出汉字时，汉字输出程序首先将用户程序提供的各种必要的参数置入特定的存储单元中，然后从指定的内存区中取出经加工处理后的汉字内码，然后将汉字内码转换成相应的汉字地址码，再根据汉字地址码从汉字字库中取出相应的汉字字形信息，并把它存入相应的内存缓冲区。最后，汉字输出程序根据输出设备的具体特点，将内存缓冲区中的汉字字形信息转换成输出设备所规定的信息格式，并将变换后的信息送到相应输出设备的输出缓冲区，由输出设备输出汉字。在输出过程中，汉字输出程序不断测试输出设备所处的状态，协调主机和输出设备在时间上的差异。输出完毕，汉字输出程序把输出控制馈出的结果参

数及其它有关信息置入约定的存储单元。

在输出大量汉字时，为了解决内存不够用的问题，汉字输出程序往往将汉字字形信息分批输出。它利用内存缓冲区将一部分汉字内码转换为汉字地址码，再到汉字字库中取出相应的汉字字形信息，一批一批地转换，一批一批地输出，直到把指定内存区中的汉字内码全部转换为汉字地址码，并据此输出全部汉字字形信息为止。

10.3.2 几种汉字输出程序

虽然各种不同汉字输出设备的汉字输出程序在设计原理上并无多大差异，但由于各种输出设备的特性、输出方式及其对输出信息格式的要求不同，因此不同种类输出设备所配置的汉字输出程序也有所不同。

下面，简要介绍显示器汉字输出程序和针式打印机汉字输出程序的基本设计思想。

1. 显示器汉字输出程序

汉字显示具有两种显示方式：汉字的图形显示方式和汉字的字符显示方式（见 2.5.2 节和 3.2.3 节）。对于这两种不同的汉字显示方式，显示器汉字输出程序也不同。

对于汉字的字符显示方式，显示缓冲区中存储的是汉字地址码。字形发生器把显示缓冲区中的汉字地址码转换为字形，在屏幕上显示出来。字符显示方式的显示器汉字输出程序的主要功能是：把内存缓冲区中汉字地址码转换成字形发生器所需要的格式，并送入显示缓冲区。

对于汉字的图形显示方式，显示缓冲区中存储的是屏幕点阵信息，用以映射为屏幕图象。图形显示方式的显示器汉字输出程序的主要功能是：把内存缓冲区中的汉字字形信息转换成显示器所需要的信息格式，并送入显示缓冲区。

下面，简要介绍图形显示方式的显示器汉字输出程序的基本

设计原理。由于显示缓冲区中的信息应是按点阵行排列的，因此，汉字输出程序的一项重要任务就是把汉字字形信息转换成按屏幕点阵行排列的格式。

假定显示器每行可显示的汉字个数为 m，一屏可显示 n 行汉字，并假定汉字字库中的汉字字形为 24×24 点阵。该显示器汉字输出程序的工作流程如图 10.5 所示。

2. 针式打印机汉字输出程序

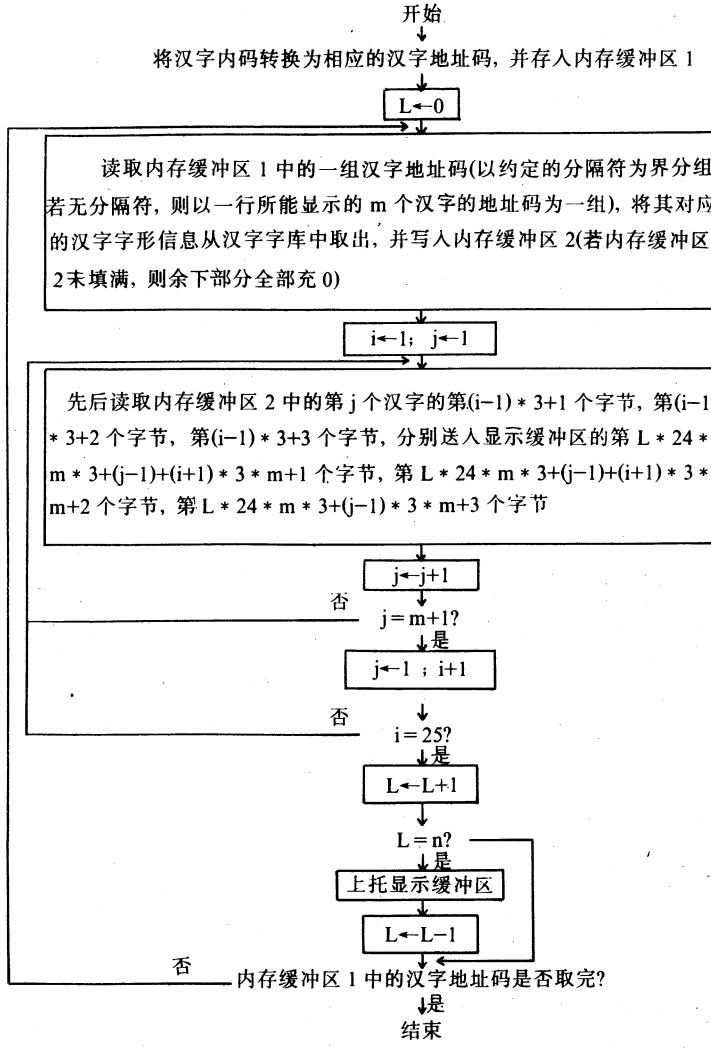
针式打印机是用一列打印针按照计算机提供的打印信息，逐列打印汉字的各列点阵，以构成汉字字形。因此，它与显示器汉字输出的情况不同，信息应是以点阵列方式组织的。针式打印机汉字输出程序的主要任务是将汉字字形信息转换成针式打字机所相容的信息格式，并送入打印机缓冲区，直到一行打印信息全部到齐，再启动打印机开始打印。此时，打印头横向移动，变换后的汉字字形信息驱动打印针电磁铁，于是纵向排列的打印针头就击打色带，从而在打印纸上打印出汉字。在输出过程中，不断测试打印机所处的状态、协调主机与打印机在时间上的差异。

假定打印机是 24 针打印机，它每行可打印的汉字个数为 m，并假定汉字字库中的汉字字形为 24×24 点阵。该 24 针打印机汉字输出程序的工作流程如图 10.6 所示。

10.3.3 汉字信息缓冲区

在汉字信息处理过程中，由于计算机内各个硬件装置的原理、结构和性能不同，因此汉字信息的存取、传送和变换的速率差异很大，常采用缓冲技术来补偿这种差异。此外，缓冲技术也用于汉字信息的分批处理。

尤其是在汉字输出处理过程中，无论是显示汉字还是打印汉字，均采用缓冲技术进行汉字信息的存取、传送和变换。一方面，在内存中开辟缓冲区，用于暂存汉字内码、汉字地址码、汉字字形信息；另一方面，在输出设备内部或在它们的控制器中装



i＝1…24　　每行汉字的点阵计数

j＝1…m　　屏幕每行显示的汉字个数

L＝0…n−1　　屏幕显示的汉字行数

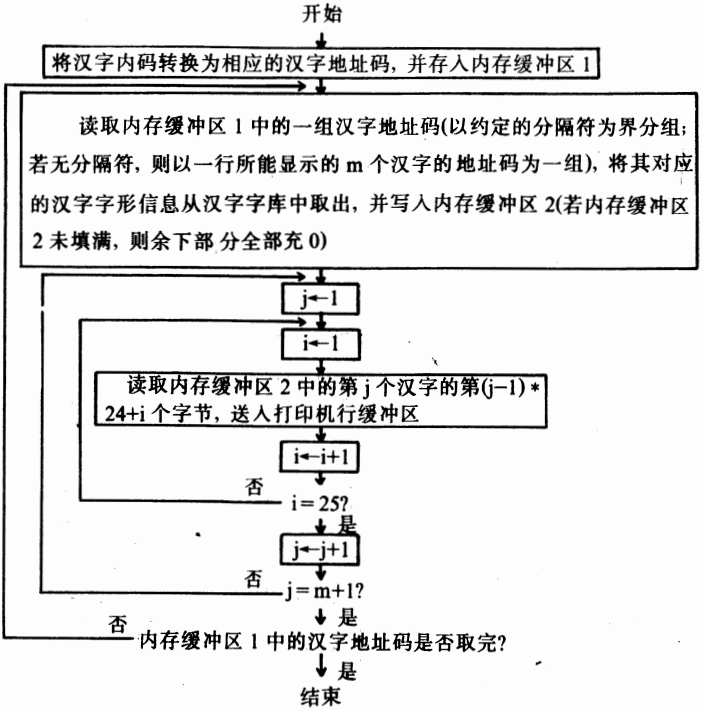
图 10.5　图形显示方式的显示器汉字输出程序流程图

有缓冲存储器，这样，内存不必等到打印完毕，就可以进行别的工作。总之，缓冲区是汉字内码，汉字字库和输出设备之间的纽带和桥梁，在汉字输出处理中起着重要的作用。

例如，对于汉字打印，预先要把需要打印的汉字文本送入内存的打印缓冲区。打印输出时，从内存的打印缓冲区取出汉字内码，并根据汉字内码从汉字字库中取出相应的汉字字形信息，存入内存的字形信息缓冲区。汉字字形信息经过变换，组织成打印机按图形方式工作时所需要的信息格式，并送往打印机的打印缓冲区，直到一行打印信息全部到齐，再启动打印机开始打印。

又例如，对于汉字的图形显示方式，由于显示器分辨率、彩色／图形适配器、显示字形的点阵数的限制，无法保持西文软件的原有显示行数和列数。为了解决这个问题，通常在内存中开辟一个显示缓冲区，并建立它与适配器上的显示缓冲区的对应关系。

下面，以 CCDOS 为例来说明内存中的显示缓冲区与彩色／图形适配器上的显示缓冲区之间的对应关系。预先把需要显示的汉字文本送入内存的显示缓冲区中，它按 80×25 字节为一块，共三块，称作字符区、标志区和属性区，分别用来存储汉字内码、字符标志、字符属性。标志区指出相应汉字内码是第 1 字节还是第 2 字节。属性区指出该汉字的彩色等属性。由于彩色／图形适配器上的显示缓冲区及其对应的显示器分辨率的限制，一屏只能容纳 40 字×11 行汉字，而内存中的显示缓冲区可容纳 40 字×25 行汉字，所以两者之间必须依靠指针来建立相互的对应关系。内存显示缓冲区中的屏幕头指针和屏幕尾指针，决定着当前屏幕上显示出来的是内存显示缓冲区中的哪一部分。头指针与尾指针的间距为 10 行，屏幕的第 11 行固定地与缓冲区的第 25 行相对应，用作提示行。通过指针移动，可以显示出内存显示缓冲区中的全部 24 行汉字文本，适配器上的显示缓冲区的作用如同一个显示窗口。



i=1…24　每行汉字的点阵行数

j=1…m　每行打印的汉字个数

图 10.6　24 针打印机汉字输出程序流程图

汉字的显示与读入都是根据光标位置进行的，而光标位置由光标指针来确定。由于内存中的显示缓冲区与适配器上的显示缓冲区容量不同，使光标定位变得十分复杂。为此，CCDOS 建立了一个以内外光标为基础的光标系统。外光标用以模拟和记录高层软件对 25 行一屏显示的操作。而内光标用于产生和记录一屏

10 行的适配器上显示缓冲区的实际操作。外光标到内光标的转换将根据内外光标的指针进行一定的换算来完成。在屏幕上显示一个汉字的过程大致如下：首先将要显示的汉字内码送至内存显示缓冲区中外光标所指定的位置，然后修改外光标，并按汉字内码从汉字字库中取出汉字字形信息，将其送入适配器上的显示缓冲区中内光标所指的位置，修改内光标，于是屏幕上原光标指定位置处就会显示出汉字来。

CCDOS 还提供了屏幕滚行处理功能。由于高层软件规定的屏幕滚动参数（滚动区界限、滚动行数等）都是按每屏 25 行考虑的，在汉字显示的情况下要与之适应，必须先滚动内存显示缓冲器的内容，字符区、标志区和属性区要一起滚动，并把滚动后空白部分的值填入内存显示缓冲区，然后再根据光标指针和屏幕头指针，计算出当前窗口中实际上要滚动的行数，来滚动屏幕和建立光标。

10.4 汉字字库管理

有的中文操作系统把汉字字库管理功能纳入各汉字输出程序之中，而有的中文操作系统则是单独设计汉字字库管理程序，由各汉字输出程序来调用。

10.4.1 汉字字库管理程序的设计原理

汉字字库管理程序用于根据汉字内码从汉字字库中取出汉字字形信息，供显示器显示汉字，或供打印机打印汉字，或供其它汉字输出方式输出汉字用。

汉字字库中的汉字字形信息通常是按汉字字符集的顺序排列的，因此汉字字库的地址编码是连续的。由于大多数汉字输入方案的汉字输入码是不连续的，所以一般不宜用汉字输入码来检索汉字字形信息。由于汉字内码与汉字地址码存在着简单的对应关

系，因此用汉字内码检索汉字字形信息是相当方便的。只需用计算公式把汉字内码转换成这个汉字字模在汉字字库中的实际开始地址，然后根据点阵的多少及取点顺序，从开始地址取出相应的汉字字形信息，供显示或打印用。

汉字字库的存储方式，对于汉字的检索和系统的运行有着直接的影响。汉字字库有三种存储方式：直接存取的汉字字库、多级存储的汉字字库，多用户共享的汉字字库（详见 3.2.2 节）。例如，对于多级存储的汉字字库，基本内存、EMS、Extend、汉卡和磁盘存储汉字字库的方式也不相同，因此汉字检索方式也不尽相同。

下面，以三级存储的汉字字库为例，来说明访问汉字字库的步骤。各汉字终端备有一级字库（不是 GB2312 的一级汉字），用 ROM 固化 512 个最常用汉字，并用 RAM 存放 512 个汉字。在主机上备有多用户共享的二级字库，存放 1024 个汉字；其余汉字放在磁盘上，为多用户共享的三级字库。

各级字库地址编码序号分配如下：

0000－0511　ROM　只读存储器字库的汉字顺序号
0512－1023　RAM　随机存储器字库的汉字顺序号
} 一级字库

1024–2047　二级字库的汉字顺序号

2048–N　　三级字库的汉字顺序号

当需要输出汉字时，根据汉字内码决定它在哪一级字库中，并得到汉字顺序号，然后按汉字顺序号计算出存放该汉字字形信息的实际开始地址，取出相应的汉字字形信息，供显示和打印用。若汉字内码在 0000–0511 之间，则从 ROM 字库中去取汉字字形信息；若汉字内码在 0512–1023 之间，则从 RAM 字库中去取汉字字形信息；若汉字内码在 1024–2047 之间，则去访问主机的二级字库；若汉字内码在 2048–N 之间，则去访问主机磁盘的三级字库，并取出相应的汉字字形信息。

10.4.2 汉字字形的存储方法

汉字字形信息有两种存储方法：整字存储法和压缩信息存储法。

整字存储法把汉字字形的点阵信息逐个字节地全部存放在汉字字库中，需要使用时可直接读出，因此亦称为点阵存储法。这种存储方法原理简单，使用方便，输出时不需要经过复杂的计算，响应时间快，也可以保证字形质量，但每种点阵的字形只适合以一种大小输出。若要放大，则容易产生锯齿状；若要缩小，则容易遗漏笔画。为了输出不同大小的字形，必须使用多个汉字字库来存放不同种点阵的字形信息，因而占存储量很大。然而，随着存储器价格的不断下降，整字存储法将会得到更为普遍的应用。

压缩信息存储法把汉字字形的压缩信息存放在汉字字库中，使用时再将压缩信息还原成点阵字形。这样做的目的是为减少汉字字库的存储量。对于点阵较多的汉字字库，特别是对于带有多个汉字字库的系统，例如，中文多字体打印系统或中文桌上排版系统，这种存储方法是节省存储空间的有效方法。但是，用这种方法生成的字形质量往往受到影响，而且字形生成速度较低。

汉字字形的压缩存储法有向量存储法、部件组字法、轮廓线字型法、笔画函数法、黑段白段法、哈夫曼树型压缩法等。这里，我们只介绍目前使用较普遍的向量存储法和部件组字法。

1. 向量存储法

向量是一个数学概念，一般是指在坐标空间中由坐标原点指向空间中任意一点的一个有方向和长度的量。用向量表示图形在图形处理领域内有着广泛的应用。

汉字字形也是一种图形，因此，可用向量表示法来描述汉字字形。将一个汉字的字形看做是由许多直线笔画组成，这些直线笔画有各种不同的方向和长度。用平面上的一系列向量来表示这

些笔画，并把这些用向量组成的汉字字形信息存储在汉字字库中，这种汉字字形的存储方法称为向量存储法。

例如，假定平面上有一个坐标系，左上角为坐标原点。数学上常常习惯把左下角取作坐标原点，由于汉字的书写习惯是从左到右、自上而下，因此把坐标原点取在左上角较为方便。又假定该坐标系用于表示 16×16 点阵字形，故 x，y 方向各有 16 个单位。平面上有一个点 A，它的坐标是 x，y，记作 A (x, y)，x，y 是大于或等于 0 且小于或等于 15 的正整数。因此，A (x, y) 表示 16×16 点阵中的一个点。

在坐标系中，可用向量表示各种不同方向和长度的直线段。x 增量$\triangle x$ 为正，表示向量从左向右；$\triangle x$ 为负，表示向量从右向左。y 增量$\triangle y$ 为正，表示向量自上而下；$\triangle y$ 为负，表示向量自下而上。折线段要用几个连接的向量来描述。

汉字字形可看作是由许多直线段组成的。我们称这些直线段为笔画。这里的笔画与一般所说的汉字笔画含义不同，是指一直线段。图 10.7 给出“次”字的笔画图。

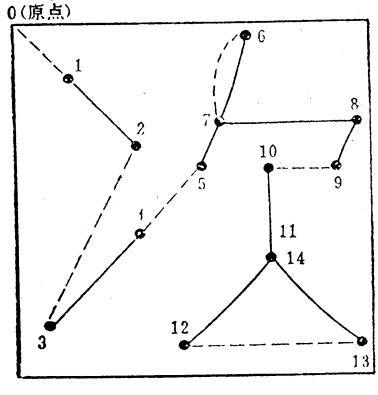


图 10.7 “次”字的向量表示

“次”字的起笔是在坐标原点。第一个向量$\overrightarrow{01}$是一个空笔画，称为虚向量，它的作用是为下一笔确定起始位置。$\overrightarrow{01}$向量的终点就是$\overrightarrow{12}$向量的起点，$\overrightarrow{12}$是一个实向量。$\overrightarrow{23}$又是一个空笔画。$\overrightarrow{34}$是一个实笔画。依次类推。其中，共十四画，八画是实笔画，六画是虚笔画，末笔是实笔画。

把“次”字的各个向量按先后次序列表如下：

序	$\triangle x$	$\triangle y$	实／虚
01	2	2	0
12	3	3	1
23	−4	8	0
34	4	−4	1
45	3	−3	0
56	2	−6	1
67	−1	4	0
78	6	0	1
89	1	2	1
910	−3	0	0
1011	0	4	1
1112	−4	4	1
1213	8	0	0
1314	−4	−4	1

总而言之，每个汉字的第一画起点一定是坐标原点。第一画的终点一定是第二画的起点，第二画的终点是第三画的起点，依次类推。每一画都有一个 x 增量$\triangle x$和一个 y 增量$\triangle y$。此外，还用一位信息来区别实笔画和虚笔画，1 表示实笔画，0 表示虚笔画。这样一组有序的笔画信息构成汉字字形的向量信息。用向量组成的汉字字形信息代替点阵信息，从而使信息存储量得到压缩。当输出汉字时，再把向量信息还原成点阵信息。

2. 部件组字法

汉字字形可以看作是由“组字部件”拼起来的。据粗略的统计，在 GB2312 基本集的一级汉字 3755 个中，由左右两部分拼起来的汉字有 2021 个，约占 53.8%；由左、中、右三部分拼起来的有 177 个字，约占 4.7%，这两种汉字合在一起占总数的 58%以上。在二级汉字 3008 个中，由左右两部分或左、中、右三部分拼起来的汉字有 2082 个，占 69%。总起来说，在基本集的 6763 个汉字中，由左右两部分或左、中、右三部分拼起来的汉字共有 4103 个，占总数的 61%。尤其值得一提的是，有一些组字部件的使用频度很高，或者说组字能力很强。下面列出在由左右两部分或左、中、右三部分拼起来的汉字中，使用频度最高的 20 个组字部件及其在 6763 个汉字中使用的次数:

氵	352	亻	213	虫	126	足	83
扌	264	讠	142	阝	121	王	80
口	261	纟	139	土	107	火	73
木	237	月	137	女	102	犭	69
全	214	忄	127	石	93	鱼	64

除了由左右两部分或左、中、右三部分拼成的汉字外，还有 7–8%的由上下两部分拼成的汉字。下面列出在由上下两部拼起来的汉字中，使用频度最高的 10 个组定部件及其在 6763 个汉字中使用的次数:

艹	336	𥫱	111	宀	68	心	67	亠	51
日	42	木	34	穴	32	雨	29	山	28

除上述两种最普遍的字形结构外，还有其它一些值得考虑的字形结构。例如，在 6763 个汉字中，含走字旁的汉字就有 104 个，含有病字头的有 98 个，含有门字部的有 45 个等等。

由于在许多汉字中都有同样的组字部件，因此在汉字字库中可只存储组成汉字的部件信息，需要输出字形时再由部件组成汉字字形。这种汉字字形的存储方法称为部件组字法。在汉字字库中有两部分信息：一部分是部件点阵信息，它是存放各种不同尺寸的部件的点阵；另一部分是组字信息，它包含每一个汉字的拼法和所需各组字部件点阵信息的所在位置。汉字字形的点阵信息是通过部件点阵信息和组字信息结合后形成的。

在部件点阵信息中，有些常用字或者不便拆开的字仍要以整字的形式存放，例如，年、月、日、我、子……，这些字估计有500—700个。

有些部件可能会有几种不同的尺寸，例如，单人旁“亻”在倒、侧、例等字中较窄，而在仍、仅、仇、代等字中较宽。为了保持字形的美观，应当收存几种不同尺寸的部件点阵信息。为了节省存储量和便于读取，可按部件点阵信息的存储长度（存储容量）来分类。凡存储长度一样的归为一类。这样，只要有一个类首址表便容易找到某一中某一部件的点阵信息。

据统计，汉字的组字部件，如果不考虑其尺寸大小，总共约一千个左右，其中有一半左右本身就是独立的字。但考虑了组字部件的尺寸以后，部件的数目有四千多个。用程序方法可实现部件字形的“变倍”，即只保存一种尺寸的部件点阵信息，需要时再由程序把它压扁或变窄，以减少信息的存储量。然而，用程序方法来变换点阵的尺寸有一定的限制和困难，程序开销较大，字形的正确性也要受到影响。因此，在大多数情况下只得收存各种大小尺寸的部件点阵信息。

用部件拼出来的汉字字形不大美观，一个重要原因是两部分拼起来的衔接部分太生硬。这是因为，实际上很多汉字的各个部件不是截然分开的，而是各部分之间相互渗透。例如，“妙”字中的“女”和“少”截然分开不大好看，互相有些交叉才较为自然美观。这种渗透现象相当普遍，必须予以考虑。改善的办法是：使

两个相互渗透的部件都放宽一些，在组字的时候故意使衔接部分的信息有些重叠。

总之，用部件组字法压缩汉字字形信息存储容量，要权衡字形质量和压缩比例，一方面要保持字形的美观，生成过程不太复杂，节省时间；另一方面要使汉字字形信息存储量的压缩比例尽可能大，尽量节省存储空间。

用向量存储法和部件组字法相结合的存储方法，可以得到更大的信息压缩比。利用这种方法组成和输出汉字字形时，只要用地址链的方式把组成该汉字的有关部件（这些部件是用坐标、斜率、长度等原始信息记载的）从存储器中取出，经过信息还原、比例尺选择、部件间相对位置的确定后，就可以得到相应的汉字字形。

10.5 汉字输入方案的自动生成

汉字输入编码方案层出不穷，但要选择功能最佳、适应面广的汉字输入编码方案却是相当难的。这是因为，首先，对于特定场合，最适用的方案因人而异，每个人喜欢的汉字输入方法不尽相同。例如，熟习汉语拼音的人喜欢字音输入法，不懂得汉语拼音的人喜欢字形输入法；会盲打的操作员喜欢重码少的输入方案，记忆力差的人喜欢规则简单的人机对话输入方案。其次，对于某个汉字输入方案来说是难找的字，换成另一种方案就可能很容易找。例如，用拼音码找一个不知道读音的汉字很难，换成字形码就可能很容易找到；反之，有的汉字用字形码很难分解，而用字音码则可能很容易找到。再其次，对于专职操作员，可以要求他们把汉字字符集中所有汉字的编码都背下来，所以只用一种汉字输入编码方案就可以工作；而一般人员就难作到这一点。最后，汉字输入编码的研究还在发展，用户要求中文系统能够随时装入新的更合用的汉字输入方案。

然而，目前大部分中文系统还仅限于含有几种汉字输入方案，而且固定在中文系统中。尽管中文系统中收集了目前最常用最时髦的汉字输入方案，也不能满足各类用户对汉字输入的不同要求。为此，有的中文系统提供了多种可供用户选择的汉字输入方案，用户可根据自己的需要，装入一种或多种外部汉字输入方案，成为常驻的汉字输入方案。还有的中文系统提供汉字输入方案的自动生成方法，用户可用这种方法自行设计、自动生成和修改任一汉字输入方案，并可随时切换常驻的汉字输入方案，从而使中文系统支持任意汉字输入方案（可容纳的汉字输入方案仅受容量的限制），具有较大的灵活性，能够较好地适应汉字输入编码研究的现状和未来发展。

对于各种汉字输入编码方案，无论是字编码还是词编码，汉字输入编码实质上均是用几个键盘值来表示一个或几个汉字（词组）。汉字输入方案的自动生成方法正是依据这个宗旨来设计的。

这种自动生成方法不仅适用于生成各种通用的汉字输入方案，而且可用于生成适合于用户特殊需要的专用汉字输入方案，或在已有的通用汉字输入方案中加入自己常用的词组。用户可根据自己所从事的行业建立专用的汉字输入方案，例如，户籍管理人员可建立街道名称输入方案，药剂师可建立药品名称输入方案，火车调度可建立火车站名输入方案，等等。用户亦可在已有的汉字输入方案（例如，拼音输入方案）中增加自己经常使用的人名、地名、单位名称、常用术语等。

下面，介绍两种汉字输入方案的自动生成方法：联想式汉字系统的编码词典的生成、修改方法和零壹中文系统的万用输入法。

1. 编码词典的生成和修改

联想式汉字系统支持任意多个汉字输入方案（仅受磁盘空间的限制）。每种汉字输入方案都具有一个编码词典。编码词典是

汉字输入码与汉字内码之间的对照表。每当按某种汉字输入方案键入汉字输入码后，系统就去查找相应的编码词典，从中取出对应的汉字内码。

联想式汉字系统提供编码词典的生成和修改方法，用户可以自己装入各种新的汉字输入方案，也可以随时修改系统中提供的各种汉字输入方案。

(1) 编码词典源文件的格式

编码词典源文件是用于生成编码词典的原始文本文件，它是汉字或词及其对应的输入编码之间的对照表。

字编码词典源文件的格式是：每行的左边是单个汉字，右边是编码（字母、数字或其它键盘符号组成的字符串），汉字与编码之间空一格。允许一个汉字对应多个不同的编码，码长也可不同，各个编码之间以空格隔开。也允许多个汉字对应一个编码。

例如，

啊　a

阿　a

行　xing　hang

词编码词典源文件的格式是：每行的左边是编码（字母、数字或其它键盘符号组成的字符串），右边是词（由一个或多个汉字组成），编码与词之间空一格。允许一个编码对应多个词，各个词之间以空格隔开。也允许多个编码对应一个词。

例如，

diannao　电脑

computer 电脑

yiyi　意义 意译 疑义 一亿 奕奕

1985　一九八五年

(2) 编码词典的生成

编码词典的生成过程如下：

①用中文字处理软件或编辑程序，按规定格式编辑编码词典

源文件。

②调用通用的词典生成程序，把编码词典源文件转换为目标文件，常把编码词典目标文件简称为编码词典。

词典生成程序生成编码词典的步骤是:

i. 确定生成词典的类型

这是一个通用的词典生成程序，既可用于生成编码词典，亦可用于生成联想字词典或联想词组词典。

ii. 输入词典源文件名

词典源文件名可以根据需要随意取，文件名的规定与操作系统相同，例如，PINYIN.TAB，CODE 等。

iii. 输入生成的方案名称

汉字输入方案名称可以根据需要随意取，一般用两个汉字或四个 ASCII 字符表示，例如，拼音、五笔、QUWE 等。

iv. 输入码元类型

码元类型取决于要生成的具体汉字输入方案，大致可分为以下几类:

1—码元为 10 个数字: 0...9

2—码元为 26 个字母: a...z

3—码元为 26 个字母加上 10 个数字

4—码元为右边小键盘（按下 NumLock 键)

5—码元为 47 个下档字符

v. 确定编码方案是字编码还是词编码

字编码和词编码的区别不仅在于它们的编码词典源文件的不同，而且在提示区中的显示方式不同，因此两者的处理方式也不同。

对于字编码，重码字（同音字、同形字或同义字）在提示区显示时是紧挨着的，中间没有符号隔开。重码字的显示顺序按汉字字符集的排列顺序，与源文件的顺序不一定相同。使用时只能逐字输入。

对于词编码，重码词（同音词、同形词或同义词）在提示区显示时，中间以箭头隔开，但若这些重码词都是由单字组成，则它们之间的箭头将不显示，看起来和字编码没有差别。重码词的显示顺序与源文件相同。既可以逐词输入，也可以逐字输入。

例如，对于拼音词组输入方案，键入 yiyi，提示区显示如下：

Q意 W义 E→ R疑 T义 Y→ U意 I译 O→ P一
A亿 S→ D奕 F奕 G H J K L
Z X C V B N M

又例如，对于英文输入方案，键入 news，提示区显示如下：

Q新 W闻 E→ R消 T息 Y U I O P
A S D F G H J K L
Z X C V B N M

一般说来，词编码可以取代字编码，尤其是当需要保证重码的显示顺序与源文件一致时，必须采用词编码格式。

vi. 确定编码输入到最大码长且无重码时是否自动输入对应的字或词

自动输入是指键入的码元个数达到最大码长时，所需的字或词能自动显示到屏幕正文区的当前光标处，例如，电报码、区位码、国标码等。要做到这一点，一般要求输入编码方案无重码或少重码，而且码长一致。为此，若确定自动输入，则应当根据输入编码方案的情况规定最大码长，以备输入汉字时，码元个数达到这个数目且无重码时，将自动输入唯一对应的字或词，有重码时仍需要选择输入。若确定不自动输入，则键完编码后必须按空格键或大写字母键才能输入对应的字或词。

vii. 确定编码词典是否分为若干个分词典

若编码词典较小（容量不超过 64KB），则只生成一个词典，命名为 DICT.TBL。

若编码词典较大（容量超过64KB，但不超过360KB），则词典生成程序会自动产生若干个分词典。每个分词典的容量以64KB为限。分词典的个数取决于源文件的大小，最多可以有6个。各分词典按生成的先后顺序依次命名为DICT.TBL，DICT.2ND，DICT.3RD，DICT.4TH，DICT.5TH和DICT.6TH。为了维护方便，也可把一个大型源文件分割成几个小文件，然后将每个小文件分别生成为上述分词典。为了使这些分别生成的分词典能够代表同一输入方案，必须在输入生成的方案名称时选择相同的方案名，并且在确定编码表是否还有续表时给出各分词典之间的链接关系。这样，词典生成程序才能知道这些分词典是代表同一输入方案，并会按生成顺序依次给它们命名：DICT.TBL，DICT.2ND，DICT.3RD，DICT.4TH，DICT.5TH，DICT.6TH。这样生成的分词典，系统将它们看作一个整体，用户使用时不必考虑是否有分词典。

viii. 上述步骤完成后，开始生成词典。生成完毕，自动返回操作系统。

③把由词典生成程序生成的编码词典的文件名换名为系统认可的词典文件名。

由词典生成程序生成的编码词典的文件名都是相同的，即DICT.*，不能被系统认可，因此要换名为系统认可的词典文件名DICTn.*，其中，n=1，2，…，19，n为输入方案编号。例如，DICT19.TBL，DICT19.2ND。要注意，不能换名为系统已有的输入方案。

④把汉字输入方案装入功能键。

新生成的编码词典换名后，便成为系统中的一种新的汉字输入方案。用METHODS命令可看到这个新方案的配置情况。由于新方案尚未装入功能键，因此还不能成为系统常驻的汉字输入方案，为此，必须用METHODS命令修改汉字输入方案配置，将新方案装入功能键。

当键入 METHODS 命令后，屏幕上显示出当前磁盘上的所有汉字输入方案的名称、编号、是字编码还是词编码、最大码长、码元类型以及当前对应的功能键。系统允许的方案编号为 1–19，其中 1–5 对应于功能键 Alt–F1 到 Alt–F5，其它作为备份。可以通过修改配置，使备份方案对应到功能键 Alt–F1 到 Alt–F5，成为常驻的汉字输入方案。在输入汉字时，用户便可通过切换功能键来选用这种汉字输入方案。

例如，当键入 METHODS 命令后，屏幕显示如下：

汉字输入配置情况:

功能键	输入方案	方案编号	词／字码	最大码长	码无类型
Alt–F1	对话	1	字		26 个字母：a...z
Alt–F2	拼音	2	词		26 个字母：a...z
Alt–F3	英中	3	词		26 个字母：a...z
Alt–F4	双拼	4	字		26 个字母：a...z
Alt–F5	区位	5	字	4	10 个数字：0...9
	国标	6	字	4	十六进制数(0...9, a...f)
	电报	7	字	4	小键盘并按下 NUM LOCK(0...9, +–.)
	八笔	8	字		小键盘并按下 NUM LOCK(0...9, +–.)
	五笔	9	词	4	26 个字母加上 10 个数字
	仓颉	10	字		26 个字母：a...z
	首尾	11	字		26 个字母：a...z

如要修改配置请按 C

否则按＜ESC＞退回系统

这时，如果要改变配置，需按 C 键。然后，根据屏幕提示，给出要装入的方案编号（1–19）和对应于该方案的功能键号 Alt–F（1–5）。若要将方案 11 装入功能键 Alt–F5，则只要给

出方案编号 11 和功能键号 5 即可。修改配置后的屏幕显示如下:

汉字输入配置情况:

功能键	输入方案	方案编号	词/字码	最大码长	码无类型
Alt-F1	对话	1	字		26 个字母: a...z
Alt-F2	拼音	2	词		26 个字母: a...z
Alt-F3	英中	3	词		26 个字母: a...z
Alt-F4	双拼	4	字		26 个字母: a...z
Alt-F5	首尾	5	字		26 个字母: a...z
	国标	6	字	4	十六进制数(0...9, a...f)
	电报	7	字	4	小键盘并按下 NUM LOCK(0...9, +-.)
	八笔	8	字		小键盘并按下 NUM LOCK(0...9, +-.)
	五笔	9	词	4	26 个字母加上 10 个数字
	仓颉	10	字		26 个字母: a...z
	区位	11	字	4	10 个数字: 0...9

如要修改配置请按 C

否则按 <ESC> 退回系统

此后，当输入汉字时，按功能键 Alt-F5 后，便可切换为首尾码汉字输入方案。

(3) 编码词典的修改

修改系统中已有的编码词典，可采用下列两种方法:

①现场修改编码词典

联想式汉字系统提供一种简单的修改词典的方法，只要按 Alt-0 键，便能在现场随时修改系统中的所有词典，包括编码词典、联想字词典和联想词组词典。这种方法适用于现场随时对词典作局部修改，用户可在系统已有的词典中加入自己常用的人

名、地名、单位名称、专业术语等，使词典在使用过程中不断完善，随着时间的推移面变得越来越适合用户的需要。

现场修改编码词典的步骤如下:

i. 按 Alt–0 键，系统自动调用词典修改程序，进入现场修改词典状态。

ii. 确定是修改编码词典，还是修改联想字词典或联想词组词典。

iii. 确定在编码词典中是增加一个词或字，还是减少一个词或字。增加或减少词还是字，取决于相应的汉字输入方案是词编码还是字编码。

iv. 输入要增加或减少的词或字的编码。编码的码元类型必须符合要修改的汉字输入方案的码元类型。

v. 输入要增加或减少的词或字。在输入过程中，可切换汉字输入方案，用常驻的任何汉字输入方案输入汉字，但输入的词之间不得有 ASCII 字符。

vi. 完成上述步骤后，新增加的词或字马上就可以使用。

vii. 若要保存编码词典的修改结果，关机前至少要切换一次汉字输入方案，或者按一次 Alt–0 键，才能将修改后的编码词典存入磁盘中。

例如，要在英文输入方案中增加一个词“新加坡”，它所对应的输入编码为 singapore，则可通过上述方法修改英文输入方案的编码词典。

②修改编码词典源文件，用词典生成程序重新生成编码词典。

这种方法适用于需要对编码词典作大量修改的场合。修改过程是：首先，用中文字处理软件或编辑程序修改编码词典源文件。如果编码词典源文件留有备份，则可直接在源文件基础上进行修改；否则，必须先调用系统提供的通用词典还原程序把要修改的编码词典还原为源文件，再去修改源文件。然后，再调用通

用的词典生成程序重新生成编码词典。编码词典的生成方法详见本节 1. (2)。

通用词典还源程序能够把系统中的所有词典还原为源文件，包括编码词典、联想字词典和联想词组词典。词典还原程序把编码词典还原成源文件的步骤如下:

i. 输入要还原的词典的文件名。必须使用系统规定的词典文件名。

ii. 决定还原后源文件是否保存。若不保存，则可以立即在屏幕上看到还原后的源文件，但只显示一次。若保存，则继续下一步骤。

iii. 输入还原后的词典文件名。词典还原程序将词典还原为源文件，并存入磁盘中。

如果输入方案有多个分词典，则需多次使用词典还原程序将每个分词典分别还原，分别存入磁盘，从而得到完整的词典源文件。

2. 万用输入法

零壹中文系统的万用输入法的基本设计思想是: 用户通过中文字处理软件或编辑程序建立一个数据文件，在数据文件中输入用户自定义的输入编码规则，即输入码及其对应的汉字或词组。该数据文件经中文系统处理后，便在系统中建立了一种新的汉字输入法，用户以后就可以使用这种新的汉字输入法。

(1) 数据文件的建立

用户首先必须用中文字处理软件或编辑程序建立一个数据文件，数据文件由输入码及其对应的汉字或词组成。数据文件名随用户喜欢命名。

输入码由键盘符号组成。每个输入码可对应多个字或词。对应于同一输入码的各字或词之间用分隔码#; 分开。每一输入码及其对应的字或词输入完毕后加上结束码###。例如，建立台北市街道名称的数据文件，每个街道名称的输入码由街道名称所含

各字的第一个注音符号构成，于是，数据文件中输入码及其对应的字或词如下所示：

T2X　承德路#；成都路##

BVR　仁孝街#；仁贤街#；

　　仁兴街#；仁信街##

(2) 数据文件的处理

当数据文件建立完成之后，必须先利用中文系统提供的万用输入法编译程序来处理这个数据文件。该编译程序要求用户给出输入法的名称（例如，街道名称）、码的最大长度、引导码。引导码主要用作分隔码和结束码，系统的原始设定为#；和##，当用户在自己的数据文件中使用的分隔码和结束码与系统的原始设定不同时，需给出用户所使用的引导码，然后再进行编译。

(3) 索引文件载入中文系统

当处理完数据文件后，产生三个索引文件。它们是数据文件的目标文件。把这三个索引文件拷贝到适当的子目录中，并指定寻找它们的路径，从而在中文系统中载入了一种新的输入法。此后，用户便可以在中文系统中使用这种新的输入法。

10.6　汉字打印驱动程序的自动生成

目前使用的打印机有许多不同的型号，它们的驱动命令各异，很难有一个通用于所有打印机的汉字打印驱动程序，因而必须为每种型号的打印机配备专用的汉字打印驱动程序。这就给操作系统的汉化工作带来沉重的负担。

有的中文操作系统在安装打印驱动程序的过程中，以菜单的形式显示出系统配置的各种打印机型号，用户根据自己实际配置的打印机型号来选择，从而在系统中装入合适的汉字打印驱动程序。

有的中文操作系统还为用户提供了通用汉字打印驱动程序的

生成程序，它通过菜单式提问，让用户给出有关打印机的特性参数，根据用户提供的这些参数自动生成所需要的汉字打印驱动程序。当然，这种生成程序只能适用于一类打印机，例如，目前流行的 Esc 序列控制图形类打印机，而对于极少数型号特殊、不标准的打印机是不适用的。

CCDOS 4.0 为 24 针打印机提供了 24×24 点阵汉字打印驱动程序的生成程序。下面，以 M2024 打印机为例来说明它的使用方法。

24×24 点汉字打印驱动程序的生成程序提出的问题及用户针对 M2024 打印机所做的回答如下:

(1) 打印机名称? M2024

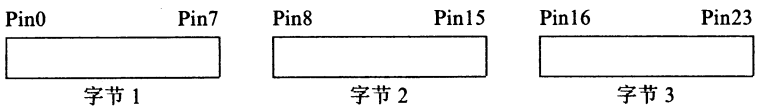
输入打印机名称，实际上是给出所生成的中文打印驱动程序的文件名，因此要符合操作系统的规定，不能多于 8 个字符。

(2) 打印机映象关系:

当打印机按图形方式打印汉字时，图形数据是以字节方式传送，若干个字节对应打印头的一列 24 针。

①八位有效 (y/n)? y

有的打印机规定，每一字节的 8 位都有效，故一列需 3 个字节，如下所示:



②六位有效 (y/n)? n

有的打印机规定，每一字节 8 位中只有 6 位有效，故一列 24 针对应 4 个字节。

③高位对第一针 (y/n)? y

打印机 24 针最上边一针为第一针。八位有效中又分为字节高位对应打印头第一针和字节低位对应打印头第一针 (Pin0)。

六位有效中又有高6位有效和低6位有效之分。

④打印机宽度？2176

每种打印机一行所能接收的图形数据列数不同。例如，M2024打印机为2176列／行，TH3070打印机为2448列／行。

综上所述，M2024打印机的映象关系是：图形数据的每字节的8位均有效，且每一字节高位对应打印头一列24针的最上面那一针；每行最多能接收2176列图形数据。

(3) 打印机位映象控制：

一般打印机打印图形有专门的图形打印控制命令，格式如下：

若干字节　　若干字节

控制码序列用于表示图形打印控制命令的功能。数据描述序列用于指明当前行打印图形数据的列数，描述分为两类：十六进制数描述和特征码描述。

①是ESC序列 (y／n)？y

当控制码序列以ASCII字符ESC（码值为十六进制1B，十进制27）开头时，称为ESC序列。

②控制码序列？G

输入控制码序列时，不包括ESC码。当控制码序列是多个字符时，回答时必须用加号“+”分隔。

③列数是十六进制数 (y／n)？y

十六进制数描述，是指用若干个字节十六进制数表示列数，而且有先高字节后低字节（即先送的字节表示高位）和先低字节后高字节之分。例如，TH3070打印机图形打印控制命令为ESC序列，八位有效，且为十六进制数描述，先高字节后低字节，命令格式是：

ESC+〃I〃+CHR(04)+CHR(100)

它表示后面的 3×(4×256+100)=3×1100 个字节，即 1100 列为图形数据。

④列数是十进制特征码 (y／n)？n

特征码描述，是指用若干个十进制特征码 (0–9) 表示列数，并规定为先高字节后低字节方式。例如，TH1351 打印机图形打印控制命令为 ESC 序列，六位有效，且为特征码描述，分四次传送，命令格式是:

ESC+〃j〃+〃0100〃

它表示后面的 4×100 个字节，即 100 列为图形数据。

⑤分几字节送打印机 (1–9)？2

⑥先高字节后低字节 (y／n)？y

⑦把打印机准备好，然后按任意键，打印机是打印是一个黑方块码 (y／n)？y

此问帮助用户检验控制序列输入正确与否。如果按任一键后，打印机没有打印出一个黑方块，说明输入有错误，需要重新回答上述打印机位映象控制的问题，直到打印机打印出一个黑方块为止，说明上述回答正确，并继续往下做。

综上所述，M2024 打印机的图形方式打印控制命令格式如下:

ESC+〃G〃+n1+n2

其中，n1，n2 表示该命令后面(n1×256+n2)×3 字节为图形数据 (0≤n≤255)，此为单向打印。

(4) 打印机走纸控制:

打印机走纸控制的提问帮助用户确定打印机走纸命令的结构。下述每个问题的含义与打印机位映象控制的问题相仿，只不过数据描述序列表示走纸量，回答方法基本相同，就不一一赘述了。

各打印机所规定的走纸单位 (即最小走纸量) 不同，而且同一打印机也可能有多种走纸命令，并规定了不同的走纸单位，常

见的有 1／6，1／8，1／48，1／170，1／180 英寸等。建议用户选择较精细的走纸命令。

①是 ESC 序列（y／n)？y

②控制码序列？J

③列数是十六进制数（y／n)？y

④列数是十进制特征码（y／n)？n

⑤分几字节送打印机（1–9)？1

⑥先高字节后低字节（y／n)？y

⑦打印机走纸是否引起回车换行（y／n)？n

有的走纸命令引起打印机缓冲区数据打印且回车换行；有的则不引起缓冲区数据打印，需再送回车和换行命令；还有的引起缓冲区数据打印后只执行回车或换行命令之一。本程序只对前两种有效。

⑧把打印机准备好，然后按任意键，打印机走纸了吗(y／n)?

此问帮助用户检验步纸控制命令输入正确否。如果打印机未走纸，说明输入有错误；如果正确，继续往下做。

⑨把打印机准备好，然后按任意键，行之间接合好（Y)，走纸多（A)，走纸少（L)?

此问帮助用户确定正确的走纸量（即行间距)。按任意键后，打印机打印出两个纵向相邻的黑方块。若它们相离，则回答 A；若它们相叠，则回答 L；系统根据用户回答自动调整行间距，直至它们刚好相接，此时回答 Y。

综上所述，M2024 打印机有多个走纸控制命令，其中走纸控制命令

ESC+〃J〃+n　　(1≤n≤255)

使打印机走纸 n／120 英寸，且发送这一命令后并不引起缓冲区数据打印和走纸，需要再送回车和换行命令。

至此，用户对汉字打印驱动程序的生成程序提出的问题全部

回答完毕。汉字打印驱程序的生成程序根据上述回答信息，自动生成汉字打印驱动程序 M2024.COM，并存入磁盘中。

第十一章　基于中文操作系统的软件汉化

基于中文操作系统的软件汉化，是指在中文操作系统支持下的程序语言、数据库管理系统、通用应用软件、以及应用程序的汉化。

基于中文操作系统的软件汉化与基于西文操作系统的软件汉化的主要区别是：基于西文操作系统的软件汉化，需要在程序语言、数据库管理系统、通用应用软件这一层上或在应用程序这一层上重新建立汉字处理功能；而基于中文操作系统的软件汉化，由于中文操作系统的汉字处理功能可以传递给它所支持的高层软件，因而程序语言、数据库管理系统、通用应用软件、甚至应用程序的汉化，只需在继承中文操作系统汉字处理功能的基础上补充增加一些无法从中文操作系统获得的汉字处理功能。

本章将从增加汉字处理功能、考虑中文环境下的特殊性问题、提示信息汉化三方面，来讨论基于中文操作系统软件汉化的基本方法。

11.1　增加汉字处理功能

基于中文操作系统的软件汉化依赖于中文操作系统的汉字处理功能。由于操作系统的汉化通常是用汉字处理模块代替西文处理模块来实现的，这样做并未改变它的系统功能调用界面，高层软件可以利用原有的系统功能调用来调用中文操作系统的汉字处理模块，因此，程序语言、数据库管理系统、通用应用软件，甚至应用程序，在中文操作系统的支持下很容易获得汉字处理功能。鉴于上述原因，基于中文操作系统的软件汉化，是一种常用

的、简单的、切实可行的软件汉化方法。

尽管如此，由于西文软件是为处理西文设计的，根本未考虑汉字的特点和汉字处理的特点，更未考虑中国人的习惯，因此，当把西文软件搬到中文操作系统环境中运行时，原有的处理西文的功能对处理汉字无能为力。为了弥补这种“先天不足”，一方面，对程序语言、数据库管理系统、通用应用软件实行汉化，在继承中文操作系统汉字处理功能的基础上，补充增加无法从中文操作系统获得的汉字处理功能；另一方面，在应用程序中利用程序语言、数据库管理系统、通用应用软件原有的命令、语句、函数继承中文程序语言、中文数据库管理系统、中文通用应用软件、甚至中文操作系统的汉字处理功能，并利用外部接口命令、语句、函数调用汉字处理模块，用以补充增加无法从低层中文软件获得的汉字处理功能。

鉴于计算机处理汉字实际上是处理汉字代码，汉字内码与西文字符代码的差异是西文软件“先天不足”和“水土不服”的重要根源。因此，汉字内码的选择和设计，对基于中文操作系统的软件汉化影响很大。对于不同形式的汉字内码，存在的问题和产生的现象也不一样，因此解决问题的办法也不相同。

下面，我们以目前使用较为广泛的高位均为 1 的双字节汉字内码为例，来说明基于中文操作系统的软件汉化是如何增加汉字处理功能的。

11.1.1 阻止高位为 1 字节的通行

用高位均为 1 的双字节表示汉字内码，容易区分汉字和西文字符，但也给软件汉化带来一些困难，这些困难主要是由高位为 1 和双字节这两个因素引起的。

鉴于有些西文软件在某些场合下不允许高位为 1 的字节通行，从而也阻碍了高位为 1 的汉字内码的通行。下面，以几个例

子说明之。

1.含有汉字的标识符

当西文软件的解释程序或由编译程序产生的目标程序进行词法分析时，要判断变量名、常量名、文件名、过程名是否为标识符。在西文软件中，标识符通常定义为以字母开头的字母和数字组成的字符串。为了检查名字是否符合标识符的定义，要逐个识别名字中的每个字符，看它的首字符是否为字母，非首字符是否为字母或数字。一般有以下两种识别方法:

(1)临界值比较法

在西文软件的解释程序或由编译程序产生的目标程序中含有与下列关系式相对应的指令:

30≤X≤39	X 为数字
41≤X≤5A	X 为大写字母
61≤X≤7A	X 为小写字母

用以在拼单词时把当前读到 X 中的字符与各种类型字符的编码临界值直接进行比较。上面的数字、大写字母、小写字母编码临界值均为十六进制 ASCII 码。

(2)映象值判别法

在西文软件中设置一个字符与字符类型映象值对照表，表中每个西文字符都对应一个映象值，每种字符类型各有一个映象值，比如，数字为 1，大写字母为 2，小写字母为 3，等等。每当拼单词时，程序首先根据当前读到的字符，在字符与字符类型映象值对照表中查找对应的映象值；然后再根据映象值来判断该字符的类型。

在中文软件中，标识符通常定义为以汉字或字母开头的汉字、字母、数字组成的字符串。也就是说，标识符的首字符可以为汉字或字母，非首字符可以为汉字、字母或数字。然而，在西文软件中，无论是采用临界值比较法，还是采用映象值判别法，都限制标识符的首字符只能是字母，非首字符只能是字母或数

字。由于高位均为 1 的双字节汉字内码的码值为 A1—FE，显然在数字、大写字母、小写字母的 ASCII 码值范围之外，因此，西文软件的原有词法分析程序将汉字内码视为非法字符，汉字内码不能通过词法检查。

对于采用临界值比较法的西文软件，只要在原有词法分析程序基础上，增加当前读到的字符与汉字编码临界值的比较操作，即可使汉字内码畅通无阻。其对应的关系式如下：

A1≤X≤FE

下面，通过剖析一个实例，来说明如何针对临界值比较法进行软件汉化。

在 dBASEⅢPLUS 基于中文 DOS 的汉化过程中，我们首先根据英文 dBASEⅢPLUS 对字段名的定义拟定字段名的汉字处理功能描述，即中文 dBASEⅢPLUS 对字段名的定义。英文 dBASEⅢPLUS 规定：字段名是以字母开头的字母、数字、下划线组成的字符串。中文 dBADE ⅢPLUS 把字段名扩充定义为：字段名是以汉字或字母开头的汉字、字母、数字、下划线组成的字符串。然后，根据字段名的汉字处理功能描述设计测试用例，通过执行 dBASEⅢPLUS 与字段名有关的命令，来检验这些测试用例是否符合上述字段名的汉字处理功能描述。

当执行 CREATE 命令建立数据库文件时，我们发现，字段名中无法输入汉字。出现的现象是：当向字段名中输入汉字时，光标不动，响铃，汉字根本显示不出来。

根据上述软件汉化发现“错误”阶段的测试结果，我们采用猜测试探查找法去查找“错误”。我们从上述出错现象着手，对出错原因进行猜想：程序中对字段名的词法分析可能采用了临界值比较法，阻碍汉字的输入。基于这种情况，程序中要有两处把当前读到的字符与编码临界值进行比较的指令。第一处要判断字段名的首字符是否属于大写字母 A–Z 或小写字母 a–z 之间的字符，第二处要判断字段名的非首字符是否属于数字 0–9、大写字母

A–Z、小写字母 a–z 或下划线_之间的字符。经过查找反汇编代码，我们在 DBA.LOD 文件中找到了这样的指令，如下:

```
1E3E:765E 3D4100      CMP     AX,0041
1E3E:7661 7C0F        JL      7672
1E3E:7663 3D5B00      CMP     AX,005B
1E3E:7666 7C0D        JL      7675
1E3E:7668 3D6100      CMP     AX,0061
1E3E:766B 7C05        JL      7672
1E3E:766D 3D7B00      CMP     AX,007B
1E3E:7670 7C03        JL      7675
1E3E:7672 33C0        XOR     AX,AX
1E3E:7674 CB          RETF
1E3E:7675 B80100      MOV     AX,0001
1E3E:7678 CB          RETF
1E3E:7679 59          POP     CX
1E3E:767A 5B          POP     BX
1E3E:767B 58          POP     AX
1E3E:767C 50          PUSH    AX
1E3E:767D 53          PUSH    BX
1E3E:767E 51          PUSH    CX
1E3E:767F 3D4100      CMP     AX,0041
1E3E:7682 7C09        JL      768D
1E3E:7684 3D5A00      CMP     AX,005A
1E3E:7687 7704        JA      768D
1E3E:7689 B80100      MOV     AX,0001
1E3E:768C CB          REFF
1E3E:768D 33C0        XOR     AX,AX
1E3E:768F CB          RETF
1E3E:7690 59          POP     CX
1E3E:7691 5B          POP     BX
1E3E:7692 58          POP     AX
1E3E:7693 50          PUSH    AX
1E3E:7694 53          PUSH    BX
```

```
1E3E:7695 51        PUSH  CX
1E3E:7696 3D6100    CMP   AX,0061
1E3E:7699 7C09      JL    76A4
1E3E:769B 3D7A00    CMP   AX,007A
1E3E:769E 7704      JA    76A4
1E3E:76A0 B80100    MOV   AX,0001
1E3E:76A3 CB        RETF
1E3E:76A4 33C0      XOR   AX,AX
1E3E:76A6 CB        RETF
1E3E:76A7 59        POP   CX
1E3E:76A8 5B        POP   BX
1E3E:76A9 58        POP   AX
1E3E:76AA 50        PUSH  AX
1E3E:76AB 53        PUSH  BX
1E3E:76AC 51        PUSH  CX
1E3E:76AD 3D3000    CMP   AX,0030
1E3E:76B0 7C19      JL    76CB
1E3E:76B2 3D3A00    CMP   AX,003A
1E3E:76B5 7C17      JL    76CE
1E3E:76B7 3D4100    CMP   AX,0041
1E3E:76BA 7C0F      JL    76CB
1E3E:76BC 3D5B00    CMP   AX,005B
1E3E:76BF 7C0D      JL    76CE
1E3E:76C1 3D6100    CMP   AX,0061
1E3E:76C4 7C05      JL    76CB
1E3E:76C6 3D7B00    CMP   AX,007B
1E3E:76C9 7C03      JL    76CE
1E3E:76CB 33C0      XOR   AX,AX
1E3E:76CD CB        RETF
1E3E:76CE B80100    MOV   AX,0001
1E3E:76D1 CB        RETF
```

经过认真分析和试验，我们知道，1E3E:765E-1E3E:7678一段程序判断字段名的首字符是否属于大写字母 A-Z 或小写字

母 a-z 之间的字符。1E3E:76AD-1E3E:76D1 一段程序判断字段名的首字符是否属于数字 0-9、大写字母 A-Z、小写字母 a-z 之间的字符（下划线在别处判断）。第一段程序的含义是：当首字符 AX 的 ASCII 码大于或等于大写字母 A 的 ASCII 码 41 且小于大写字母 Z 后面的[的 ASCII 码 5B，或者，大于或等于小写字母 a 的 ASCII 码 61 且小于小写字母 z 后面的{的 ASCII 码 7B 时，转向 7675；否则，转向 7672。第二段程序的含义是：当非首字符 AX 的 ASCII 码大于或等于数字 0 的 ASCII 码 30 且小于数字 9 后面的：的 ASCII 码 3A，或者，大于或等于 41 且小于 5B，或者，大于或等于 61 且小于 7B 时，转向 76CE；否则，转向 76CB。

在找到出错位置并查出出错原因后，要通过修改程序来纠正“错误”，并用测试用例反复执行程序，检验是否符合汉字处理功能描述，直到正确为止。

为了允许字段名中含有汉字，一种简便的纠正“错误”方法是：把第一段程序中的

```
1E3E:766D 3D7B00      CMP     AX,007B
```

改为

```
1E3E:766D 3DFF00      CMP     AX,00FF
```

把第二段程序中的

```
1E3E:76C6 3D7B00      CMP     AX,007B
```

改为

```
1E3E76C6 3DFF00       CMP     AX,00FF
```

这样，就把小写字母的码值范围下限 7A 扩充到 FE，其中包括汉字的码值 A1-FE，从而在建立数据库文件时字段名中就能输入汉字，并在屏幕上显示出来。

这是一种简单但不精确的纠正“错误”方法。原因是汉字内码的码值是 A1-FE，而上述这种方法使字段名中也允许含有码值为 7B-A0 的字符，例如，{{，显然是一个非法的字段名。

下面，给出一个精确的但较为复杂的纠正“错误”方法。它把第一段程序改为：

```
1E3E:765E 3D4100      CMP     AX,0041
1E3E:7661 7C2A        JL      768D
1E3E:7663 3D5B00      CMP     AX,005B
1E3E:7666 7C21        JL      7689
1E3E:7668 3D6100      CMP     AX,0061
1E3E:766B 7C20        JL      768D
1E3E:766D 3D7B00      CMP     AX,007B
1E3E:7670 7C17        JL      7689
1E3E:7672 3DA100      CMP     AX,00A1
1E3E:7675 7C16        JL      768D
1E3E:7677 EB10        JMP     7689
```

它把第二段程序改为：

```
1E3E:76AD 3D3000      CMP     AX,0030
1E3E:76B0 7CDB        JL      768D
1E3E:76B2 3D3A00      CMP     AX,003A
1E3E:76B5 7CD2        JL      7689
1E3E:76B7 3D4100      CMP     AX,0041
1E3E:76BA 7CD1        JL      768D
1E3E:76BC 3B5B00      CMP     AX,005B
1E3E:76BF 7CC8        JL      7689
1E3E:76C1 3D6100      CMP     AX,0061
1E3E:76C4 7CC7        JL      768D
1E3E:76C6 3D7B00      CMP     AX,007B
1E3E:76C9 7CBE        JL      7689
1E3E:76CB 3DA100      CMP     AX,00A1
1E3E:76CE 7CBD        JL      768D
1E3E:76D0 EBB7        JMP     7689
```

在第一段程序中，增加了对首字符 AX 是否汉字码值的判断，指令如下：

```
1E3E:7672 3DA100      CMP     AX,00A1
1E3E:7C16 7C16        JL      768D
1E3E:7677 EB10        JMP     7689
```

它覆盖了原来的指令:

```
1E3E:7672 33C0        XOR     AX,AX
1E3E:7674 CB          RETF
1E3E:7675 B80100      MOV     AX,0001
1E3E:7678 CB          RETF
```

同样，在第二段程序中，增加了对非首字符 AX 是否汉字码值的判断，指令如下:

```
1E3E:76CB 3DA100      CMP     AX,00A1
1E3E:76CE 7CBD        JL      768D
1E3E:76D0 EBB7        JMP     7689
```

它覆盖了原来的指令:

```
1E3E:76CB 33C0        XOR     AX,AX
1E3E:76CD CB          RETF
1E3E:76CE B80100      MOV     AX,0001
1E3E:76D1 CB          RETF
```

恰巧，在两段程序之间有两条指令

```
1E3E:7689 B80100      MOV     AX,0001
1E3E:768C CB          RETF
```

与第一段程序中被覆盖的指令

```
1E3E:7675 B80100      MOV     AX,0001
1E3E:7678 CB          RETF
```

和第二段程序中被覆盖的指令

```
1E3E:76CE B80100      MOV     AX,0001
1E3E:76D1 CB          RETF
```

相同；而且，在两段程序之间还有两条指令

```
1E3E:768D 33C0        XOR     AX,AX
1E3E:768F CB          RETF
```

与第一段程序中被覆盖的指令

```
1E3E:7672 33C0        XOR     AX,AX
1E3E:7674 CB          RETF
```

和第二段程序中被覆盖的指令

```
1E3E:76CB 33C0        XOR     AX,AX
1E3E:76CD CB          RETF
```

相同。正是利用了这种机会，只要将原来转向 7672 和 76CB 的指令均改为转向 7689，并将原来转向 7675 和 76CE 的指令均改为转向 768D，便可腾出位置用来容纳判断 AX 是否汉字码值的指令。

至此为止，能不能说字段名的汉化已经完成。回答是否定的。这是因为，上述汉化仅仅使在用 CREATE 命令建立数据库文件时字段中能输入汉字，但不能保证所有使用字段名的地方均符合字段名的汉字处理功能描述，因此，尚需执行所有与字段名有关的命令，检验测试用例是否符合字段名的汉字处理功能描述。

果然如此,虽然在建立数据库文件时定义了含有汉字的字段名,但在使用 LIST 和 DISPLAY 命令显示字段内容时，仍不允许含有汉字的字段名跟在 LIST 和 DISPLAY 之后。例如，

```
. list 姓名
syntax error
      ?
list  姓名
. display  姓名
syntax error
      ?
display  姓名
```

根据测试结果,查找反汇编代码，我们在 DBA.LOD 文件中找到了下列指令:

```
1E3E:7612 3C41      CMP     AL,41
1E3E:7614 7C38      JL      764E
1E3E:7616 3C5B      CMP     AL,5B
1E3E:7618 7C0A      JL      7624
1E3E:761A 3C61      CMP     AL,61
1E3E:761C 7C30      JL      764E
1E3E:761E 3C7B      CMP     AL,7B
```

```
1E3E:7620 732C        JNB     764E
1E3E:7622 2C20        SUB     AL,20
1E3E:7624 46          INC     SI
1E3E:7625 41          INC     CX
1E3E:7626 3B4E10      CMP     CX,[BP+10]
1E3E:7629 7D23        JGE     764E
1E3E:762B AA          STOSB
1E3E:762C 8A04        MOV     AL,[SI]
1E3E:762E 3C30        CMP     AL,30
1E3E:7630 7C1C        JL      764E
1E3E:7632 3C3A        CMP     AL,3A
1E3E:7634 7CEE        JL      7624
1E3E:7636 3C41        CMP     AL,41
1E3E:7638 7C3A        JL      764E
1E3E:763A 3C5B        CMP     AL,5B
1E3E:763C 7CE6        JL      7624
1E3E:763E 3C61        CMP     AL,61
1E3E:7640 7C08        JL      764A
1E3E:7642 3C7B        CMP     AL,7B
1E3E:7644 7308        JNB     764E
1E3E:7646 2C20        SUB     AL,20
1E3E:7648 EBDA        JMP     7624
1E3E:764A 3C5F        CMP     AL,5F
1E3E:764C 74D6        JZ      7624
1E3E:764E 32C0        XOR     AL,AL
```

经过分析和试验，我们了解到这段程序的含义：若字段名的首字符为大写字母，则转向 7624；若字段名的首字符为小写字母，则减去 20（转换为大写字母），接着执行 7624；否则，转向 764E。若字段名的非首字符为数字或大写字母或下划线，则转向 7624；若字段名的非首字符为小写字母，则减去 20（转换为大写字母），转向 7624；否则，转向 764E。

为了使字段名中允许含有汉字，我们把上述程序修改为下列

程序:

```
1E3E:7612 3C41      CMP     AL,41
1E3E:7614 7C38      JL      764E
1E3E:7616 3C5B      CMP     AL,5B
1E3E:7618 7210      JL      762A
1E3E:761A 3C61      CMP     AL,61
1E3E:761C 7C30      JL      764E
1E3E:761E 3C7B      CMP     AL,7B
1E3E:7620 7304      JNB     7626
1E3E:7622 2C20      SUB     AL,20
1E3E:7624 EB04      JMP     762A
1E3E:7626 3CA1      CMP     AL,A1
1E3E:7628 7224      JB      764E
1E3E:762A 46        INC     SI
1E3E:762B 41        INC     CX
1E3E:762C 3B4E10    CMP     CX.[BP+10]
1E3E:762F 7D1D      JGE     764E
1E3E:7631 AA        STOSB
1E3E:7632 8A04      MOV     AL,[SI]
1E3E:7634 3C30      CMP     AL,30
1E3E:7636 7C16      JL      764E
1E3E:7638 3C3A      CMP     AL,3A
1E3E:763A 7CEE      JL      762A
1E3E:763C 3C41      CMP     AL,41
1E3E:763E 7C0E      JL      764E
1E3E:7640 3C58      CMP     AL,5B
1E3E:7642 7CE6      JL      762A
1E3E:7644 3C61      CMP     AL,61
1E3E:7646 7C02      JL      764A
1E3E:7648 EBD4      JMP     761E
1E3E:764A 3C5F      CMP     AL,5F
1E3E:764C 74DC      JZ      762A
1E3E:764E 32C0      XOR     AL,AL
```

为了增加对 AL 是否汉字码值的判断，我们在程序中加入下列三条指令:

```
1E3E:7624 EB04        JMP     762A
1E3E:7626 3CA1        CMP     AL,A1
1E3E:7628 7224        JB      764E
```

为了给它们腾出位置，把原来程序中 1E3E:7624−1E3E:7640 的指令下串到 1E3E:762A−1E3E:7646 的位置(保持原来的指令)，因此，原来程序中转向 7624 的指令均改为转向 762A。同时，由于指令下串，覆盖了原来程序中的下列指令:

```
1E3E:7642 3C7B        CMP     AL,7B
1E3E:7644 7308        JNB     764E
1E3E:7646 2C20        SUB     AL,20
```

这三条指令恰好与原来程序中的下列三条指令相同:

```
1E3E:761E 3C7B        CMP     AL,7B
1E3E:7620 732C        JNB     764E
1E3E:7622 2C20        SUB     AL,20
```

利用这种机会，可使字符判断和非首字符判断公用这三条指令，并把原来程序中的指令

```
1E3E:7648 EBDA        JMP     7624
```

改为:

```
1E3E:7648 EBD4        JMP     761E
```

同时，为了适应判断 AL 是否汉字编码的三条指令的加入，把原来程序的指令

```
1E3E:7620 732C        JNB     764E
```

改为:

```
1E3E:7620 7304        JNB     7626
```

对于采用映象值判别法的西文软件，词法分析程序要么拒绝映射高位为 1 的字节，要么把高位为 1 的字节映射到西文字符与字符类型映象值对照表以外的内存单元，由于这些内存单元可能作其它变量或常量用，会出现随机值，因而会引起混乱。解决这

个问题有以下两种方法:

①扩充字符与字符类型映象值对照表。使它包括高位为1字节的映象值。由于我们把汉字视为小写字母，因此将码值为A1−FE的汉字内码字节的映象值定义为小写字母的映象值。这种方法不必修改词法分析程序，因而逻辑关系非常清楚，但需要增加原对照表的长度，有时会极大地浪费磁盘空间。

②修改词法分析程序，使它增加对汉字码值的判断，并将汉字内码字节映射为大写字母或小写字母的映象值。也就是说，每当读到一个字节时，程序判断它的码值是否在A1−FE之间，若是，则根据任一字母（例如，小写字母z）在字符与字符类型映象值对照表中查找对应的映象值。

2.含有汉字的字符串或文本文件

当输入输出字符串或文本文件时，西文软件的解释程序或由编译程序产生的目标程序往往要检查输入输出的字符是否可打印字符。它拒绝非可打印字符的输入输出。例如，当输入非可打印字符时，响铃警告，屏幕正文区的光标不动，或出现出错信息(比如，在输入非可打印字符的位置显示问号?)。

在采用ASCII字符的系统中，可打印字符的码值为十六进制20−7E。这样，在西文软件的解释程序或由编译程序产生的目标程序中必然含有与下列关系式相对应的指令:

20≤X≤7E

用以检查当前输入输出的字符X是否可打印字符。汉字的码值为A1−FE，显然无法通过这样的检查。

为了解决这个问题，简便的方法是把检查当前输入输出的字符X是否可打印字符的指令放宽为与下列关系式相对应的指令:

20≤X≤FE

这种方法是不精确的，其中含有码值为7F−A0的非打印字符。精确的方法是：增加检查当前输入输出的字节X是否汉字

内码字节的指令，其对应的关系式如下：

A1≤X≤EF

下面，通过剖析两个实例，来说明如何针对可打印字符的检查进行软件汉化。

在 dBASEⅢPLUS 基于中文 DOS 的汉化过程中，我们首先根据英文 dBASEⅢPLUS 对文本文件和备注字段的定义拟定它们的汉字处理功能描述。英文 dBASEⅢPLUS 规定：文本文件和备注字段中包含可打印的 ASCII 字符。中文 dBASEⅢPLUS 把文本文件和备注字段扩充定义为：文本文件和备注字段中包含汉字和可打印的 ASCII 字符。然后，根据文本文件和备注字段的汉字处理功能描述设计测试用例，通过执行 dBASEⅢPLUS 与文本文件和备注字段有关的命令，来检验这些测试用例是否符合上述汉字处理功能描述。

当执行 MODIFY COMMAND 建立文本文件时，键入的汉字（例如，姓名，汉字内码为 D0D5 C3FB）在屏幕上显示出来，按 Ctrl-End 键退出全屏幕编辑并保存编辑的内容。但是，当再次执行 MODIFY COMMAND 命令进入全屏幕编辑时，屏幕上原来显示的汉字变成了其它字符（例如，姓名变成了 PUC{（编码为 50 55 43 7B），显然汉字内码字节的高位 1 被滤掉了。根据这一猜测，试探出在文件 DBA.OVL 文件中存在一条这样的指令：

AND BYTE PTR [SI], 7F

把它改为

AND BYTE PTR [SI], FF

便解决了这一问题。

类似地，当再次进入备注字段编辑屏幕时，原来在备注字段输入并显示的汉字变成了其它字符，例如，姓名变成了 PUC{，显然汉字内码字节的高位 1 被滤掉了。根据这一猜测，试探出在 DBA.LOD 文件中存在一条这样的指令：

AND WORD PTR [BP−04],007F

把它改为

AND WORD PTR [BP−04],00FF

便解决了这一问题。

11.1.2 高位为 1 的字节已被派作它用

鉴于有些西文软件在某些场合下已把高位为 1 的字节派作别的用场，致使它们与高位为 1 的汉字内码发生冲突，从而阻碍了中文操作系统汉字处理功能的传递。下面，以几个例子说明之。

1.高位 1 用作反相显示标志

在菜单提示中，常常采用高亮度光标来选择菜单中的项目，高亮度光标往往是用字符的反相显示来形成的。反相显示的字符和背景颜色与正常显示相反。若正常显示是黑底白字，则反相显示为白底黑字。其它一些提示也常用反相显示来提醒用户注意。

由于有的软件把高位 1 用作字符的反相显示标志，而汉字内码高位恰为 1，因此汉字被显示为反相的 ASCII 字符。

解决这一问题的最简单办法是：一方面，修改形成反相标志的程序，从程序中删除 ASCII 字符设置反相标志（高位置为 1）的指令；另一方面，修改根据反相标志产生反相显示的程序，从程序中删除实现反相显示的指令。

2.高位为 1 的字节用于存储压缩计数

为了实现存储优化，有些西文软件采用了存储压缩技术。在存储字符时，对若干个连续相同的字符只用两个字节表示，第一个字节容纳这种字符的码值，第二个字节表示这种字符连续的个数，并将该计数字节的高位置为 1。当取出字符时，根据高位为 1 的计数字节，取出若干个连续相同的字符，这种字符就是从计数字节前面一个字节中取出的。

由于用于存储压缩的计数字节以高位 1 为标志，这样就与高位为 1 的汉字内码字节发生冲突，导致把汉字内码字节误认为是

用于存储压缩的计数字节，从而使含有汉字的变量名、文件名、字符串等在屏幕上非但没显示汉字，反而显示出许多连续相同的字符。

为了解决这一问题，只有忍痛割爱，牺牲这一存储优化功能。一方面，在存储字符时，不必判断是否有若干个连续相同的字符，因而不必构造用于存储压缩的计数字节。另一方面，当取出字符时，不必判断是否有高位为 1 的计数字节，因而也不必解释并展开若干个连续相同的字符。这样，高位 1 就专供汉字内码使用，从而不会发生上述因高位 1 标志的冲突而发生的混乱情况。

11.1.3 两个高位为 1 字节配对的二义性问题

由于用两个相邻的高位为 1 的字节合在一起表示一个汉字内码，因此，当对含有汉字的字符串进行操作时，常常会产生二义性问题，即某一高位为 1 的字节是与紧靠它前面的一个高位为 1 的字节配对构成一个汉字内码，还是与紧靠它后面的一个高位为 1 的字节配对构成一个汉字内码。软件汉化必须考虑这种问题，消除高位均为 1 的双字节汉字内码的二义性。下面，以几个例子说明之。

1.查找高位均为 1 的双字节汉字内码

由于任何两个高位为 1 的字节均可组合成一个汉字内码，因此在对汉字进行编辑时很容易产生二义性。例如，在汉字串“汉化”中查找“夯”字。“汉字”二字基于 GB2312 的汉字内码为 BABABBAF，“夯”字基于 GB2321 的汉字内码 BABB。本来，在汉字串“汉字”中不存在“夯”字，但由于高位为 1 的双字节汉字内码的二义性，当西文软件用“夯”字的汉字内码 BABB 去查找 BABABBAF 时，会把其中的 BABB 误为“夯”字的汉字内码，从而产生错误的结果。由此可见，如果要查找的汉字的汉字内码恰好与某一汉字的第二个字节及紧靠它后面的汉字的第一个字节

的代码相同，那么会出现因代码交叉而造成的查找错误。如果查找的不是单个汉字，而是汉字串，则发生这种错误的概率也就很少。

解决这一问题的办法是：当查找到匹配的字节串后，检查前面的高位为1的字节是否偶数个。若是，则表示查到所需要的汉字；否则，继续向后寻找匹配的字节串，直到找到匹配的字节串并且前面的高位为1字节为偶数个为止。

2.用西文字符替换一个汉字的第一字节

当修改一汉字串，用一个西文字符替换一个汉字的左半部分时，后面的汉字串就会发生混乱。这是因为，在输入或输出缓冲区中，当用单字节的西文字符替换双字节的汉字内码的第一个字节时，后面便出现了奇数个高位为1的字节，使它们两两匹配错误，从而导致汉字串的显示或打印混乱。

解决这一问题的办法是：让输入或输出缓冲区中的字符串始终保持汉字内码字节配对，不出现奇数个高位为1的字节。为此，当用西文字符替换一个汉字的第一字节时，把它的第二个字节充为空格，从而避免了奇数个高位为1字节的出现。

3.光标位于一个汉字的右半部时字符串的增删改

原则上，一般不允许用户在光标位于一个汉字的右半部时进行字符串的插入、删除或替换等操作，否则会引起后面的汉字串或前后局部范围内的汉字显示或打印出现混乱。例如，当插入一个西文字符时，后面的汉字串出现混乱；当插入一汉字时，或进行删除时，或用一西文字符或一汉字进行替换时，光标前后局部范围内的汉字会出现混乱。出现混乱的原因，要么是出现了奇数个高位为1的字节，要么出现了孤立的汉字内码字节，要么汉字内码字节两两匹配错误。

上述错误是由于当光标位于汉字的右半部时进行增删改操作引起的，尽管一般不允许用户这样做，但出错处理程序应尽量避免出现混乱，以免使用户不知所措。然而，由于西文软件是为处

理西文字符而设计的，未考虑对上述双字节汉字内码引起的错误做出错处理，因此，在软件汉化时，应充分考虑对上述问题的处理，防止奇数个高位为 1 字节的出现，并把有关引起混乱的字节充为空格，例如，在一个汉字的两个字节之间插入一个西文字符时，可把这个汉字的两个字节均充为空格。

4.显示优化引起的奇数个高位为 1 的字节

为了实现显示优化，在有些西文软件中，每当用一个字符去替换屏幕中光标所在位置上的字符时，都要比较两个字符是否相同，只有当两个字符不相同时才更新并显示这个字符。

由于当用一个汉字串去覆盖屏幕上原有的汉字串时，可能会存在以一个高位为 1 的字节去修改相同的字节的情况，而这个字节的修改未使屏幕接受这一字节，因此在汉字显示过程，导致了汉字串中高位为 1 的字节个数为奇数，从而引起汉字显示的混乱。

为了解决这一问题，对于高位为 1 的字节，无论修改字节和被修改字节是否相同，都要求它重新显示。

5.以单字节为单位处理汉字

由于西文软件是为处理西文字符设计的，字符串的插入、删除、替换等操作运算往往是以单字节为单位进行处理。而汉字内码是由双字节组成的，因此西文软件对汉字串的某些操作运算有时是不适应的。例如，当插入汉字时，因单字节再显示会引起汉字内码字节两两匹配错误，而使汉字显示混乱。

解决这一问题的办法是：根据汉字内码字节的奇偶计数值，遇到一个汉字的第一个字节时，光标停留在当前位置上，等待第二个字节；待下一个字节出现时，把这两个高位为 1 的字节拼为汉字内码，以双字节为单位进行再显示，把这个汉字显示在当前光标处；显示完后，光标后移两个字节。

11.1.4 汉字绕回问题

在以显示为基础的字处理或数据输入过程中，西文软件常采用绕回技术。当超过显示宽度的限制时，光标会自动从当前行的最后一个位置跳到下一行的第一个位置。

由于西文软件是为处理西文字符设计的，因而对每行最后一个位置不应出现半个汉字的禁则现象未予考虑，从而导致了边界显示的混乱。下面，以几个例子说明之。

1.数据输入中的汉字绕回问题

当向数据库中输入数据时，数据受字段宽度的限制,不接受超过字段宽度限制的字符。每当用户键入一个字符时，程序都要进行边界检查，一旦到达右边界，程序将光标自动移到下一字段的第一个位置上。（当字段未填满时，可按回车键跳到下一字段。）

当光标位于一个字段的最后一个位置上时，若键入一个汉字，则由于一个汉字由两个字节组成，边界检查程序只能把该汉字的第一个字节放在该字段的最后一个位置上，然后强送一个回车符，把该汉字的第二个字节移到下一字段的第一个位置上，从而把一个汉字分为两半在两个字段中显示。鉴于汉字显示程序在接受一个汉字的第一个字节时，将光标停留在当前位置上，等待它的第二个字节，这与边界检查程序发生矛盾，从而会使边界显示混乱。

为了解决这一问题，要修改字段边界检查程序。当输入的字节个数等于（字段宽度−1）时，不允许继续输入汉字，只好牺牲一个字节。

2. 字处理中的汉字绕回问题

当用字处理软件或编辑程序连续键入文字时，字处理软件或编辑程序一般会自动调整输入换行。由于西文字处理软件或编辑程序是为处理西文字符设计的，以英文单词为单位换行，未考虑汉字换行问题，因此会在一行右端出现半个汉字的禁则现象。

为了解决这一问题，中文字处理软件或编辑程序应考虑以单个汉字为单位换行，当光标到达行尾的前一个位置上时，若继续

输入汉字，则在行尾空-位置，把这一汉字绕回到下一行的第一个位置上显示。

11.1.5 汉字的比较和排序

依照西文软件中字符串比较和排序的概念和做法，含有汉字的字符串的比较，实质上是比较汉字和西文字符的码值；含有汉字的字符串的排序，实质上是按汉字和西文字符码值的大小排序。

汉字的码值取决于汉字字符集及其内码形式。西文字符的码值取决于西文字符集及其代码体系。汉字的码值和西文字符的码值之间的关系取决于汉字字符集与西文字符集之间的关系。

显然，这种利用西文软件原有的字符串比较和排序功能，以汉字和西文字符的码值为依据进行含有汉字的字符串的比较和排序是不适宜的。一般说来，除等于和不等于比较有意义外，其它比较（小于、大于、小于或等于、大于或等于）没有多大意义。由于汉字内码排列顺序杂乱无章，而且没有统一的形式，因此，汉字内码不宜作序值用。

为了解决这一问题，可通过对程序语言、数据库管理系统、通用应用软件的汉化增加汉字比较功能和汉字排序功能。用户亦可在应用程序中利用程序语言、数据库管理系统、通用应用软件的外部接口命令、语句或函数（例如，CALL，RUN 等）调用中文系统提供的标准的汉字排序模块，按汉字的字音、字形或字义等属性对汉字排序，例如，拼音序、部首序、笔画序等。用户还可根据自己的需要编写自己的汉字排序模块，按自己定义的属性值排列顺序对汉字排序，例如，省市地名、单位、姓名等（见 6.5.1 节 6.）。

11.1.6 汉字的输入输出

根据软件功能的传递性，程序语言、数据库管理系统、通用

应用软件的绝大多数汉字输入输出功能是从中文操作系统传递来的。这是通过它们的解释程序或由编译程序产生的目标程序调用中文操作系统的汉字输入输出模块来实现的。它们的输入输出语句、命令、函数被解释程序解释为或被编译程序编译为中文操作系统的系统功能调用。当执行输入输出命令、语句、函数时，解释程序或由编译程序产生的目标程序调用中文操作系统的汉字输入输出模块，根据对输入输出信息类型的判断，截取与过滤输入输出信息流，或转入原西文字符处理流程，或转入增加的汉字处理流程。由此可见，基于中文操作系统的程序语言、数据库管理系统、通用应用软件在汉字输入输出功能方面的汉化工作，仅仅在于排除阻碍汉字输入输出的障碍，或补充增加汉字输入输出功能。

除此之外，用户亦可以在应用程序中，一方面，利用程序语言、数据库管理系统、通用应用软件原有的命令、语句、函数(例如，PRINT,LPRINT,INPUT,READ,WRITE,CHR $ (),ESC序列等)，继承中文程序语言、中文数据库管理系统、中文通用应用软件、甚至中文操作系统的汉字输入输出功能，例如，汉字打印字形变换；另一方面，利用程序语言、数据库管理系统、通用应用软件的外部接口命令、语句、函数（例如，CALL,RUN等)，调用中文系统提供的标准的汉字输入输出模块，例如，中文多字体打印程序、汉字报表打印程序等。用户亦可根据自己的需要编写自己的汉字输入输出模块，例如，建立 dBASE 通用多字体报表系统 dGMRS 用以在 dBASE 中打印多字体报表（见 1.3.3 节)。

11.1.7 其它形式汉字内码存在的问题

上面，我们较详细地例举了高位为 1 的双字节汉字内码增加汉字处理功能存在的问题。对于不同形式的汉字内码，存在的问题不一样，因此，基于中文操作系统的软件汉化方法也不尽相

同。例如，高位分别为 1 和 0 的双字节汉字内码，虽然不存在高位均为 1 的双字节内码那种两个高位为 1 字节配对的二义性问题。但是，无论是第一个字节高位为 1 第二个字节高位为 0 的双字节汉字内码，还是第一个字节高位为 0 第二个字节高位为 1 的汉字内码，由于汉字内码中存在一个高位为 0 的字节，这个字节的码值与 ASCII 码相同，因此也会产生另一种二义性问题。下面，以几个例子扼要说明之。

(1)当在含有汉字的字符串中查找 ASCII 字符时，由于在扫描字符串过程中，可能会遇到与该 ASCII 字符码值相同的高位为 0 的汉字内码字节，从而导致错误的结果。解决这一问题的办法是，当查找到匹配的字节后，对于第一个字节高位为 1 第二个字节高位为 0 的双字节汉字内码，检查前一个字节是否高位为 1，对于第一个字节高位为 0 第二个字节高位为 1 的双字节汉字内码，检查后一个字节是否高位为 1。若是，则说明当前找到的字节是汉字内码字节，需要继续向后寻找匹配的字节，直到找到匹配的字节并且不为汉字内码字节为止；否则，表示查找所需要的 ASCII 字符。

(2)有些软件在标识符中的字母不分大小写，即把标识符中的大写字母和小写字母认为是等价的。因此，在查找一个标识符时，要先把小写字母转换为大写字母。若汉字内码的高位为 0 的字节恰好落在小写字母范围之内，则西文软件会把汉字内码字节误为小写字母而转换为大写字母，因而汉字内码字节的码值就会改变。解决这一问题的办法是：修改词法分析程序，当查找一个含有汉字的标识符时，汉字内码字节不转换为大写字母。

(3)汉字内码高位为 0 的字节与回车符 0DH 或换行符 0AH 发生冲突，使某些汉字内码字节当作回车符或换行符用，从而导致某些汉字的丢失。为了解决这一问题，当遇到 0DH 或 0AH 时，要判断它们是汉字内码字节还是回车、换行符，分别予以处理。

11.2 考虑中文环境下的特殊性问题

中文操作系统提供的中文环境与原西文环境有一定的差异，因此，原来在西文环境下运行的西文软件在某些方面可能会不适应中文环境。为了使西文软件的原有功能适应于中文环境，必须对西文软件进行适当的改造。也就是说，基于中文操作系统的软件汉化要考虑在中文环境下的特殊性问题，例如，显示方式、屏幕显示行数、显示字符和打印字符与汉字内码的冲突等问题。下面，通过四个方面的例子，来说明基于中文操作系统的软件汉化是如何考虑中文环境下的特殊性问题的。

11.2.1 汉字的显示方式问题

西文软件解释程序或由编译程序产生的目标程序内部大多都设置了字符显示方式。读、写、显示字符及清除屏幕等操作使用的是西文操作系统提供的系统功能调用，而这些系统功能调用是某些中文操作系统所不支持的。对于汉字的字符显示方式来说，由于汉字显示完全用硬件按字符方式实现，高层软件的显示环境不变，因此，基于这种中文操作系统的高层软件在显示方面基本上无需汉化。然而，对于汉字的图形显示方式来说，由于操作系统在显示方面所提供的系统功能调用发生了变化，因此，必须把高层软件内部在显示方面使用西文操作系统的系统功能调用的地方，改为使用中文操作系统的系统功能调用。例如，基于CCDOS的软件汉化:

(1)若西文软件内部用 INT 10 设置字符显示方式，则应把它改为设置图形显示方式。

(2)若西文软件内部采用将字符及其属性直接填入单色显示适配器的显示缓冲区（B000:0000）的方法，不能简单地把B000:0000 改为 B800:0000（彩色／图形显示适配器的显示缓冲

区)，则必须改为通过 INT 10 调用 CCBIOS 的显示管理模块，用它所提供的汉字显示功能来安排屏幕显示。

(3)若西文软件内部用 INT 10 按字符方式清除屏幕，不能简单地把显示方式置为图形方式，这样清屏操作会变得很慢，而应当在字符显示方式下往屏幕送 80×25 个空格后，再把显示方式转为图形方式。

11.2.2 屏幕显示行数问题

西文软件的屏幕显示一般是按 25 行字符显示方式设计的。由于汉字的显示方式不同，显示器的分辨率和显示适配器上显示缓冲区的容量不同，致使中文操作系统的屏幕显示行数和列数也有所不同。

对于配有高分辨率显示器的微型机来说，中文操作系统提供了 25 行正文显示，从而保持了西文软件原有的显示行数。因此，西文软件在屏幕显示行数方面无需汉化或只做很少的修改。

为了保持西文软件原有的显示行数和每行的显示字数，实现每屏 25 行×40 字的汉字显示，对于以图形方式实现的 16×16 点阵汉字显示，显示器分辨率至少应为 640×400。如果显示器分辨率较低，则应根据中文操作系统提供的屏幕显示行数 (10 行、20 行、24 行等)，协调西文软件的屏幕显示行数。为此，要对西文软件有关屏幕显示行数的地方进行汉化，尤其是全屏幕操作，以适应中文操作系统提供的中文显示环境。例如，CCDOS 采用 CGA 以图形方式显示汉字，每屏只能显示 10 行正文。基于 CCDOS 的软件汉化必须解决下列两个问题:

(1)由于西文软件的屏幕画面是按 25 行设计的，因此 10 行显示导致西文软件的屏幕画面显示不完整。软件汉化必须把显示信息压缩，并安排在 10 行之内。尤其是对于表格，屏幕画面设计太满，显示内容很难安排在 10 行之内，必须合理布局，否则会使屏幕显示面目皆非，有效行数极少，无法投入实用。

(2)由于西文软件规定的屏幕滚动参数（滚动区界限、滚动行数）都是按每屏 25 行考虑的，因此 10 行显示导致屏幕向上翻滚，因要与光标对齐而将有用的显示信息卷到屏幕上边去。软件汉化必须把光标安排在 10 行之内，以 10 行为一页面进行换屏(见 10.3 节 3.)。

11.2.3 中文环境下的西文字符显示问题

西文软件在西文字符显示方面是在西文环境下设计的，当把西文软件搬到中文环境下运行时，西文字符显示有时会出现副作用。

例如，若中文操作系统采用高位均为 1 的双字节汉字内码，则由于汉字内码的码值（A1–FE）占据了 ASCII 扩充字符集中的制表符、背景符等图形字符的码值位置，因此，西文软件中凡是成对显示 ASCII 扩充字符集字符的地方，在中文操作系统上运行时均会误认为汉字，而在屏幕上呈现出不伦不类的汉字。下面，例举几种常见的情况，并讨论软件汉化的方法。

(1)西文软件常常使用单横线“—”画框线，由于它的码值为 C4，而 C4C4 恰是“哪”字的汉字内码，因此，西文软件在中文环境下运行时，两两单横线会变为一个“哪”字。软件汉化的办法是：把 C4 改为 2D（减号），则原来的单实线呈现为单虚线。

(2)西文软件常常使用双横线“＝”画框线，由于它的码值为 CD，而 CDCD 恰是“屯”字的汉字内码，因此，西文软件在中文环境下运行时，两两双横线会变为一个“屯”字。软件汉化的办法是：把 CD 改 3D（等号），则原来的双实线呈现为双虚线。

(3)西文软件常常采用下列制表符来构造窗口或表格：

图形字符	—	=	|	‖	┌	┐	└	┘	╒	╕	╘	╛
码　值	C4	CD	B3	BA	DA	BF	C0	D9	C9	BB	C8	BC

在中文环境下，由于汉字内码占用了上述制表符的码值位

置，因此窗口水平框线两两制表符组成并显示一个汉字。窗口垂直框线尽管当右边为空格时可保持显示原来的图形字符，但当两个窗口的垂直框线相连时，亦会组成并显示一列汉字。软件汉化的办法是，把上述制表符的码值改为合适的 ASCII 字符，建议如下：

制表符	─	=	│	‖	┌	┐	└	┘	▼	▼	▲	▲
ASCII 字符	—	=	¦	¦	◆	◆	◆	◆	▼	▼	▲	▲
	2D	3D	7C	7C	04	04	04	04	1F	1F	1E	1E

(4)西文软件在全屏幕操作中有时用背景符▒来构造背景，由于它的码值为 B0，而 B0B0 恰是“鞍”字的汉字内码，因此，西文软件在中文环境下运行时，两两背景符▒▒会变为一个“鞍”字。软件汉化的办法是：把 B0 改为 20（空格），则原来的阴影背景呈现为空白背景。但是，在某些情况下，例如，在屏幕生成器，空格符是有意义的字符，因此不可把背景符改为空格符，此时应把它改为@，%等图形字符。

(5)西文软件常在屏幕上用◀┘来表示回车键(Enter 或 Return)，图形字符◀┘的码值分别为 11 C4 D9，而 C4 D9 恰是“馁”字的汉字内码，因此，西文软件在中文环境下运行时，会变成◀馁。软件汉化的办法是：把 11 C4 D9 连同旁边的一个空格改为 GB2312 字符集中的两个图形字符←┘，它们的汉字内码分别为 A1FB 和 A9BC（区位码分别为 0191 和 0928）。

11.2.4 中文环境下的西文字符打印问题

西文软件在西文字符打印方面是在西文环境下设计的。当把西文软件搬到中文环境下运行时，西文字符打印有时会出现副作用。

例如，西文软件目前越来越倾向于设置西文字符的多字体打印功能，但是，这种多字体打印功能是建立在西文操作系统基础上的，而中文操作系统一般是不支持的。为了适应中文环境，要

么忍痛割爱，舍掉西文字符的多字体打印功能，要么对软件大幅度汉化。又例如，有的西文软件在中文操作系统下运行时，遇下划线后另起一行打印，从而使打印列错位。把 5F（下划线的码值）改为 20（空格），变通地解决了这一问题。

11.3 提示信息汉化

提示信息汉化的目的在于，为用户提供交互式人机对话的中文界面。本节将简述提示信息的一般存储形式，分别讨论一般形式的提示信息和加密形式的提示信息的汉化方法，介绍用于提高提示信息汉化效率和质量的辅助工具。这些概念和方法不仅适用于基于中文操作系统的软件，而且适用于各层次上的软件。

11.3.1 提示信息的存储形式

西文提示信息一般是以字符串的形式存储在软件的有关文件中的。字符串的开始符、结束符及其它字符没有统一的规定和标准，不同的软件有不同的规定和标准。常用的开始符有字母（码值为 41H−5AH 和 61H−7AH）或%（码值为 25H），结束符有 $（码值为 24H）、NUL（码值为 00H）或 BLANK（码值为 FFH）等，中间包含的字符可为可打印字符（码值为 20H−7EH）、回车符（码值为 0DH）、换行符（码值为 0AH）或响铃符（码值为 07H）等。

确定提示信息的表示形式，是提示信息汉化的先决条件。下面，以寻找提示信息字符串的开始符和结束符为例，来说明如何确定提示信息的存储表示形式。

例如，用调试程序 debug 在 DISKCOPY · COM 文件中寻找提示信息字符的开始符和结束符。在执行 DISKCOPY 命令过程中，我们在屏幕上看到两条提示信息，如下：

Insert source diskette in drive A:

Copy complete

下面，我们利用它们来寻找提示信息字符串的开始符和结束符。寻找过程如下：

(1)执行 debug，调入文件 DISKCOPY · COM；

(2)检索上述两个字符串在内存中的开始地址：

-S CS:100 B10 "Insert source"

508C:074F

-S CS:100 B10 "Copy complete"

508C:07BB

两个字符串的开始地址分别为 508C:074F 和 508C:07BB。

(3)显示两个字符串的内容

-D CS:74F Insert source diskette in drive @ ¨ $

-D CS:7BB Copy complete ¨ $ ¨

由此可见，开始符为字母，结束符为 $。

这种寻找方法找到的开始符和结束符在提示信息字符串中不一定是唯一的，但较常见的是一种软件中只有一种开始符和一种结束符。

11.3.2 提示信息汉化的步骤

提示信息汉化的过程大致可分为以下三个步骤：

1.寻找西文提示信息

寻找西文提示信息有两条途径：一条途径是运行西文软件，记录屏幕上显示的或打印机上打印的西文提示信息，然后通过软件汉化工具在软件的有关文件中检索与该提示信息匹配的字符串；另一条途径是分析并确定提示信息字符串的存储形式，直接在软件的有关文件中逐一提取出所有的西文提示信息。

2.把西文提示信息修改为中文提示信息

用软件汉化工具把西文软件中的西文提示信息替换为中文提示信息。在替换过程中，不要修改字符串的结束符，也不要修改

字符串中间的回车符、换行符、响铃符等非显示字符，否则会引起提示信息的显示混乱。

尤其值得注意的是，有的提示信息还在程序中兼作别用，对于这种提示信息的汉化一定要倍加小心，否则会产生意想不到的结果。

3.把修改后的提示信息记入西文软件中

通过汉化工具把修改后的中文提示信息保存到磁盘文件中，并通过在中文操作系统下运行西文软件来检验屏幕上显示的中文提示信息是否正确。若有错误，再反馈去改正错误。

11.3.3 提示信息汉化的辅助工具

上述常规的提示信息汉化方法是一种串行的提示信息汉化方法，它利用软件汉化的借用工具，逐一地寻找西文提示信息，逐一地把找到的西文提示信息修改为相应的中文提示信息。这种常规的提示信息汉化方法既容易出错，速度又慢，远远满足不了目前软件汉化工作对质量和速度的要求。

为了提高提示信息汉化的生产力和可靠性，目前出现了许多提示信息汉化的辅助工具。这些辅助工具大多采用并行的提示信息汉化方法，自动地寻找西文提示信息，通过全屏幕编辑修改提示信息，从而提高了软件汉化的质量和速度，使提示信息汉化工作并行化、自动化、简单化。

1.并行化

软件汉化的并行化，就是根据软件汉化人员的需要，辅助工具把西文软件的文件划分为若干个能被并行汉化的小文件，便于分工汉化，加快汉化速度。各个小文件汉化完后，辅助工具再把它们按划分文件前的顺序连接起来。

2.自动寻找西文提示信息

自动寻找西文提示信息，就是根据软件汉化人员事先给出的提示信息字符的开始符和结束符的定义，辅助工具自动地顺序地

从文件中逐一提取出所有西文提示信息，供软件汉化人员提示信息汉化用。软件汉化人员通过控制辅助工具，既可以在找完文件中全部西文提示信息后结束，又可以中途结束，保存断点位置；既可以从第一个西文提示信息找起，也可以从上次中途结束保存的断点位置开始；既可以由前向后寻找西文提示信息，又可以由后向前寻找西文提示信息。

3.全屏幕编辑修改提示信息

每当辅助工具找到一个西文提示信息时，便进入全屏幕编辑状态。在全屏幕编辑状态下，辅助工具把当前找到的西文提示信息一式两行显示在屏幕上，第一行供修改时参考对照用，第二行供修改用。软件汉化人员使用规定的各种控制键，对屏幕上显示的第二行西文提示信息进行编辑，把它修改为相应的中文提示信息，经过反复编辑直到软件汉化人员感到满意为止，辅助工具自动地把文件中的西文提示信息修改为中文提示信息。然后，辅助工具继续寻找下一个西文提示信息，继续修改为中文提示信息，直到软件中的所有西文提示信息均汉化完毕为止。

11.3.4 加密形式的提示信息的汉化

有些西文软件对重要的提示信息（例如，版权屏幕）往往采用加密形式给出。这些提示信息在汉化之前首先需要解密，然后再根据加密形式予以汉化。

例如，一个西文字符由两个十六进制数表示，每一个十六进制数在文件中存放的密码及其用于屏幕显示的译码的对应关系如下：

第一个十六进制数：

密码	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
译码	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5

第二个十六进制数:

密码	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
译码	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A

由此可见，如果要把屏幕上西文提示信息修改为中文提示信息，必须首先把中文提示信息的汉字内码翻译成密码形式，然后按这种密码形式修改文件中的西文提示信息，这样屏幕上就可以按译码形式显示中文提示信息。比如，“中国科学院”的汉字内码依次是 D6D0 B9FA BFC6 D1A7 D4BA，把它们翻译为密码形式为：7375 1C5F 1A63 7402 711F，按这种密码形式修改西文提示信息，便可在屏幕上显示出“中国科学院”来。

加密形式的提示信息的汉化，关键在于解密或译码。提示信息的加密形式是多种多样的，要根据软件的有关文件中存放的密码形式和屏幕上显示或打印机上打印的字符，分析加密的形式，经过反复试验，设法找出规律。最好能找出加密函数，例如，原码加 1 成为密码。如果实在找不出加密函数，可找出在文件中存放的密码与用于显示或打印的译码的对应关系，例如，一个西文字符由两个十六进制数表示，第一个十六进制数不变，第二个十六进制数的密码与译码对应关系如下:

密码	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
译码	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6