

计算机等级考试辅导（二级）

FoxBASE 数据库管理系统

谭浩强 主编

齐英杰 薛镭 徐霞 杨斌 徐士良 编著

侯炳辉 审

清华大学出版社



计算机等级考试辅导

(二级)

FoxBASE 数据库管理系统

谭浩强
齐英志
侯炳辉

薛雷 徐霞 杨斌 徐士良

主编
编著
审

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书是为帮助读者准备参加各类计算机等级考试而编写的辅导材料,内容包括计算机的基本知识和 FoxBASE 数据库管理系统,基本上覆盖了全国和各地方计算机等级考试的考试大纲要求。

本书的每一章都包括:本章的内容要点、学习难点、例题分析和思考题四部分。针对考试中容易出现的问题作了重点讲解和分析。

本书可作为考生的考前辅导教材,也可作为各类人员学习计算机基本知识和 FoxBASE 数据库管理系统的辅导教材,也可供自学参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

计算机等级考试辅导(二级): FoxBASE 数据库管理系统/谭浩强主编;齐英杰等编著. —北京:清华大学出版社,1995

ISBN 7-302-01858-8

I. 计… I. ①谭… ②齐… II. ①电子计算机-关系数据库-数据库管理系统, FoxBASE②电子计算机-等级-考试-自学参考资料 IV. TP311.13

中国版本图书馆 CIP 数据核字(95)第 06562 号

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:昌平环球印刷厂

发行者:新华书店总店科技发行所

开 本: 787×1092 1/16 印张: 15.5 字数: 367 千字

版 次: 1995 年 5 月第 1 版, 1995 年 5 月第 1 次印刷

书 号: ISBN 7-302-01858-8/TP·838

印 数: 00001—10000

定 价: 13.50 元

前 言

随着我国计算机普及第二次高潮的兴起,各行各业的人们都在努力地学习计算机知识,推动计算机应用。继“中国计算机软件专业技术资格和水平考试”之后,1994 年国家教委考试中心又推出了面向社会的“全国计算机等级考试”。许多省市还开展了面向高校非计算机专业学生的计算机水平测试。这些计算机统一考试的一个共同目的是,力求提供一个统一的、客观的要求和标准,以衡量应试者掌握计算机知识的程度,供社会上各用人单位参考。

经过许多专家的研讨,认为计算机初学者学习计算机知识应当按以下四个层次循序渐进地进行,即: 1. 计算机基本知识和初步的操作使用; 2. 程序设计,能用一种高级语言或数据库语言编制程序; 3. 进一步学习软硬件知识,具有计算机应用软件的初步开发能力(偏软的方向)或计算机应用系统的初步分析和设计能力(偏硬的方向); 4. 结合各个专业应用领域的需要,深入学习有关的计算机知识,深入开展计算机应用。

目前,全国和地方的计算机等级考试基本上是按照以上层次划分考试等级的,有的设置了三个考试等级,有的设置了前二个等级。考试的形式多数采用“标准题”(即“客观题”)形式,有的还有上机测试。

为了帮助以上各种考试的应试者复习有关考试内容,熟悉考试形式,检查自己掌握的程度,根据广大读者的要求,我们编写了这套“计算机等级考试辅导”丛书。该丛书包括一级、二级考试的内容,其中二级又包括 BASIC、FORTRAN、PASCAL、C、FoxBASE 五本。每一本书的各章的内容大体上都包含:本章内容要点、学习难点、例题分析、思考题等几部分。当然,这套丛书不是教材,它不可能包罗考试中可能遇到的全部内容和所有形式,只是为读者复习提供的一种辅导材料。

本丛书是邀请京津地区高等学校中具有较丰富教学经验的教师编写的。本书第一章由徐士良编写,第二章、第八章由齐英杰编写,第三章由徐霞编写,第四章、第五章由杨斌编写,第六章、第七章由薛镭编写,侯炳辉审阅了第二至第八章。由于时间较紧,可能会有不尽如人意之处。我们准备在第一版出版之后,根据广大读者的意见和等级考试的发展,对本书作必要的修改和补充,使之不断完善。

主编 谭浩强

1995 年 2 月

目 录

第一章 计算机基本知识	1
1.1 本章要点	1
1.1.1 微型机的硬件与软件系统.....	1
1.1.2 计算机中数的表示及编码.....	3
1.1.3 DOS 操作系统概述	7
1.1.4 常用 DOS 命令的使用及常见错误分析.....	11
1.1.5 微型机的系统维护	23
1.1.6 计算机病毒及其防治	24
1.2 本章难点.....	25
1.2.1 计算机中数的表示及运算	25
1.2.2 磁盘文件的树状目录结构	27
1.2.3 DOS 命令的正确使用	28
1.2.4 输入输出改向	30
1.2.5 几种打印方式的比较	30
1.3 例题分析.....	31
1.3.1 选择题	31
1.3.2 填空题	35
1.4 思考题.....	38
1.4.1 选择题	38
1.4.2 填空题	40
第二章 数据库基本概念	42
2.1 本章要点.....	42
2.1.1 数据库系统	42
2.1.2 数据库管理系统	43
2.1.3 数据库应用系统	45
2.1.4 关系数据库	46
2.1.5 FoxBASE 关系数据库	47
2.2 本章难点.....	51
2.2.1 数据库的分级结构	51
2.2.2 关系数据库的特殊运算	51
2.3 例题分析.....	53
2.4 思考题.....	55

• ■ •

第三章 FoxBASE + 基本语法规则	57
3.1 本章要点	57
3.1.1 基本术语	57
3.1.2 数据类型	58
3.1.3 函数	60
3.1.4 FoxBASE+各种文件类型的特点	73
3.1.5 FoxBASE+命令的语法规则及书写格式	76
3.2 本章难点	79
3.2.1 表达式类型及其运算	79
3.2.2 数组	82
3.2.3 数据库文件	85
3.3 例题分析	88
3.4 思考题	94
第四章 FoxBASE + 简单程序设计	95
4.1 本章要点	95
4.1.1 命令文件的建立与执行	95
4.1.2 简单输入输出	98
4.1.3 顺序结构程序设计	101
4.1.4 分支结构程序设计	102
4.1.5 循环结构程序设计	103
4.1.6 过程调用	106
4.1.7 程序调试	111
4.2 本章难点	114
4.2.1 命令文件的建立与执行	114
4.2.2 简单输入输出	115
4.2.3 屏幕格式设计	119
4.2.4 菜单程序设计	121
4.2.5 过程之间的数据传递	124
4.2.6 程序调试	126
4.3 例题分析	127
4.4 思考题	133
第五章 库文件操作及其应用初步	134
5.1 本章要点	134
5.1.1 库文件结构的操作	134
5.1.2 工作区的有关规定	136
5.1.3 移动记录指针	137

5.1.4	排序与索引	139
5.1.5	查询、筛选与统计	142
5.1.6	多工作区操作	146
5.2	本章难点	148
5.2.1	FoxBASE+的索引原理	148
5.2.2	模糊查询	148
5.2.3	记录指针分析	149
5.2.4	多用户环境下数据库的操作	150
5.3	例题分析	155
5.4	思考题	158
第六章	数据输入输出及数据库维护	160
6.1	本章要点	160
6.1.1	数据输入	160
6.1.2	数据输出	165
6.1.3	删除数据	174
6.1.4	编辑数据	178
6.2	本章难点	181
6.2.1	有关全屏命令	181
6.2.2	记录指针与 INSERT 命令	182
6.2.3	由其它文件向库文件输入数据	183
6.2.4	CLEAR ALL 与 CLOSE ALL 的区别	183
6.2.5	有关数据复制的若干问题	183
6.2.6	几个有关删除的命令之间的关系	184
6.2.7	全屏幕编辑修改命令	184
6.2.8	替换命令	185
6.3	例题分析	185
6.4	思考题	188
第七章	数据库系统管理操作命令	189
7.1	本章要点	189
7.1.1	FoxBASE+的仿 DOS 命令	189
7.1.2	管理打印机的操作	191
7.1.3	有关内存变量的操作	193
7.1.4	系统环境保护及屏幕信息收集	196
7.1.5	功能键及系统状态设置	198
7.1.6	FoxBASE+的系统配置	201
7.2	本章难点	206

7.2.1	仿 DOS 命令的作用	206
7.2.2	仿 DOS 命令的特点	207
7.2.3	打印机的有关设置	208
7.2.4	输出系统当前状态	209
7.2.5	有关 FoxBASE+ 系统的内存配置	209
7.3	例题分析	211
7.4	思考题	212
第八章	FoxBASE + 程序设计及上机操作	214
8.1	程序设计基本方法	214
8.1.1	程序设计的概念和步骤	214
8.1.2	程序流程图	215
8.1.3	程序控制结构	215
8.1.4	程序设计风格	217
8.2	FoxBASE+ 系统程序设计的特点	218
8.3	程序的测试与调试	220
8.3.1	程序中常见的错误类型	220
8.3.2	测试方法	221
8.4	例题分析	224
8.4.1	程序设计上机试题常见类型及解题方法	224
8.4.2	例题分析	225
8.5	思考题	230
附录 A	FoxBASE + 2.10 命令集(字母顺序)	232
附录 B	FoxBASE + 2.10 函数集(字母顺序)	238

第一章 计算机基本知识

1.1 本章要点

1.1.1 微型机的硬件与软件系统

一个完整的计算机系统由硬件系统与软件系统两大部分组成。

一、微型机的硬件系统

1. 中央处理器(又称 CPU)

它是由运算器和控制器两部分组成的。其中运算器负责数据的算术运算和逻辑运算;控制器负责提供控制信号,协调并控制输入输出操作以及对内存的访问。根据 CPU 能同时处理的数据位数,常见的微机有 8 位机、16 位机、32 位机,目前又推出了 64 位的高档微机。CPU 的品质决定了微机系统的档次。

2. 存储器

它用于存放程序或数据信息。存储器中的存储单元是以字节为单位进行编址的。存储器的容量是存储器中所包含的字节数,通常用 KB 或 MB 表示,有时也用 K 或 M 来表示。其中 $1\text{KB}=1024$ 字节, $1\text{MB}=1024\text{KB}$ 。

存储器又分为内存与外存。

内存储器按其工作方式又可以分为随机读写存储器 RAM 和只读存储器 ROM。RAM 中的数据可以随机地读出和写入,断电后, RAM 中的信息会丢失。ROM 中的数据只能读出而不能写入新的内容,断电后, ROM 中的信息不会丢失。在一般的 PC 机中,内存容量为 640KB,它包括了 RAM 和 ROM,档次较高的微机其内存容量还可以扩充。

内存与 CPU 一起构成了计算机的主机部分。

外存储器又称为辅助存储器。外存的容量一般要比内存大得多,而且可以移动,便于不同系统之间的信息交流。在微机中,最常见的外存储器有磁盘,包括硬盘和软盘。

硬盘的容量要比软盘大得多,目前一般微机所配置的硬盘容量在 60MB 以上。

目前微机上常用的软盘有:容量为 360K 的 5 吋双面双密度(DS,DD)软盘、容量为 1.2M 的 5 吋双面高密度(DS,HD)软盘与容量为 1.44M 的 3 吋双面高密度软盘。5 吋软盘的一侧有一个写保护口,当贴上保护纸后只能读而不能写入。同样,在 3 吋软盘上有一个写保护孔。

在第一次使用磁盘时,必须首先对磁盘格式化。

3. 输入设备

它是外界向计算机传送信息的装置。在微机系统中,最常用的输入设备是键盘。

磁盘机(即磁盘驱动器)也是一种输入设备,它将磁盘上的信息传送到主机中。

4. 输出设备

它的作用是将计算机中的数据信息传送到外部媒介,并转化成某种为人们所认识的表示形式。在微机系统中,最常用的输出设备有显示器和打印机。

显示器和键盘构成了微机系统的控制台。

磁盘机也是一种输出设备,它负责将主机中的信息传送到磁盘上保存起来。

二、微型机的软件系统

通常,软件包括计算机运行所需的各种程序和数据,以及有关的文档。

硬件是软件的物质基础,但硬件能否发挥其作用又取决于软件。

软件一般可以分为系统软件、应用软件和支持软件。

1. 系统软件

系统软件通常是指管理、监控和维护计算机资源(包括硬件和软件)的一种软件。最常用的系统软件有:

- 操作系统;
- 各种语言处理程序,如汇编程序、编译程序及解释程序等;
- 数据库管理系统,如 FoxBASE 等。

2. 应用软件

应用软件是指利用计算机及系统软件为解决各种实际问题而编制的、具有专门用途的软件。常见的应用软件有:

- 各种字处理软件,如,汉字编辑软件 WORDSTAR、汉字字表编辑软件 CCED 及桌面印刷系统 WPS 等;
- 各种用于科学计算的软件包;
- 计算机辅助制造、辅助设计、辅助教学等软件;
- 各种图形软件等。

3. 支持软件

支持软件是指在计算机硬件与系统软件的基础上,用于支援其他软件研制和开发的软件。它的目的是方便用户编制应用软件。

三、微型机的分类及主要性能指标

1. 微型机的分类

- 按字长分
分为 8 位机、16 位机、32 位机和 64 位机。
- 按结构分
分为单片机、单板机、多芯片机与多板机。
- 按用途分
分为工业过程控制机和数据处理机。

2. 微型机主要性能指标

- 字长

- 时钟频率
- 内存容量
- 外部设备配置
- 软件配置

四、微型机的应用领域

微型机的应用领域很广泛,且应用水平也越来越深入。主要应用有以下几个方面:

- 科学技术计算;
- 数据处理;
- 过程控制;
- 计算机通信;
- 计算机辅助设计、辅助制造、辅助测试与辅助教学等。

1.1.2 计算机中数的表示及编码

计算机中的所有信息都是以二进制形式表示的。为了阅读和书写方便,一般又用八进制或十六进制表示。

一、计算机常用计数制

1. 二进制数

在二进制数中,基数为 2。因此在二进制数中出现的数字字符只有两个:0 与 1。每一位计数的原则为“逢二进一”。

要将十进制整数转换为二进制整数可以采用“除 2 取余”法:将十进制数除以 2,得到一个商数和余数,再将商数除以 2,又得到一个商数和余数。这个过程一直做下去,直到商数为 0 为止,每次得到的余数即为对应二进制数的各位数字,但必须注意:最先得到的余数为二进制数的最低位,最后得到的余数为二进制数的最高位。

例如,将十进制整数 69 转换为二进制整数,过程如下:

2 69	余数为 1, 即 $a_0=1$
2 34	余数为 0, 即 $a_1=0$
2 17	余数为 1, 即 $a_2=1$
2 8	余数为 0, 即 $a_3=0$
2 4	余数为 0, 即 $a_4=0$
2 2	余数为 0, 即 $a_5=0$
2 1	余数为 1, 即 $a_6=1$
0	商为 0, 结束

因此, $(69)_{10} = (1000101)_2$

要将十进制纯小数转换为二进制小数可以采用“乘 2 取整”法:用 2 乘十进制小数,其整数部分为 a_{-1} ,再用 2 乘余下的纯小数部分,其整数部分为 a_{-2} 。这个过程一直做下去,直

到余下的纯小数为 0 或满足所要求的精度为止。最后可得到对应的二进制小数 $(0.a_{-1}a_{-2}\dots a_{-n})_2$ 。必须注意,一个十进制小数不一定能准确地转换为二进制小数。

例如,将十进制小数 0.357 转换为二进制小数,过程如下:

$$\begin{array}{rcl}
 0.357 & & \\
 \times 2 & & \\
 \hline
 0.714 & \text{整数为 0, 即 } a_{-1}=0 & \\
 \\
 0.714 & & \\
 \times 2 & & \\
 \hline
 1.428 & \text{整数为 1, 即 } a_{-2}=1 & \\
 \\
 0.428 & & \\
 \times 2 & & \\
 \hline
 0.856 & \text{整数为 0, 即 } a_{-3}=0 & \\
 \\
 0.856 & & \\
 \times 2 & & \\
 \hline
 1.712 & \text{整数为 1, 即 } a_{-4}=1 & \\
 \\
 0.712 & & \\
 \times 2 & & \\
 \hline
 1.424 & \text{整数为 1, 即 } a_{-5}=1 & \\
 \\
 \vdots & &
 \end{array}$$

上述过程可以一直做下去,但是不能穷尽。如果要求只取到二进制小数点后第五位为止,则

$$(0.357)_{10} \approx (0.01011)_2$$

对于一般的十进制数,可以分别将整数部分转换与小数部分转换后再组合起来。例如:

$$(69.357)_{10} \approx (1000101.01011)_2$$

2. 八进制数

在八进制数中,基数为 8。因此,在八进制数中出现的数字字符有 8 个:0,1,2,3,4,5,6,7。每一位计数的原则为“逢八进一”。

与二进制数类似,将十进制整数转换为八进制整数可以采用“除 8 取余”法;十进制小数转换为八进制小数可以采用“乘 8 取整”法。

二进制数转换为八进制数采用如下方法:从小数点开始,向前每三位一组构成一位八进制整数位;从小数点开始向后每三位一组构成一位八进制小数位,若最后不足三位二进制位,则应添 0 补足三位。例如:

$$\begin{aligned}
 (69.357)_{10} &\approx (1000101.01011)_2 \\
 &= (1,000,101.010,110)_2 \\
 &= (105.26)_8
 \end{aligned}$$

3. 十六进制数

在十六进制数中,基数为 16。因此,在十六进制数中出现的数字字符有 16 个:0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,其中 A、B、C、D、E、F 分别表示值 10,11,12,13,14,15。十六进制数中每一位计数原则为“逢十六进一”。

将十进制数转换为十六进制数的方法与二进制数、八进制数类似,其整数部分采用“除 16 取余”法,小数部分采用“乘 16 取整”法。

二进制数转换为十六进制数采用如下方法:从小数点开始向前每四位一组构成一位十六进制整数位;从小数点开始向后每四位一组构成一位十六进制小数位,若最后不足四位二进制位,则应添 0 补足四位。例如:

$$\begin{aligned}
 (69.357)_{10} &\approx (1000101.01011)_2 \\
 &= (0100,0101.0101,1000)_2 \\
 &= (45.58)_{16}
 \end{aligned}$$

同样,也可以很方便地将十六进制数转换为二进制数。

二、计算机中数的表示

数有正有负。在计算机中表示一个数时,总是用最高位表示数的符号,其中“0”表示正,“1”表示负。

在计算机中,小数点位置固定的数称为定点数。通常,计算机中的定点数有两种表示法:

- 小数点默认为在一个二进制数最后一位的后面。这种定点数称为定点整数。
- 小数点默认为在二进制数的最高位(即符号位)后面。这种定点数称为定点小数。

在计算机中,既有整数部分又有小数部分的数称为浮点数。通常,浮点数表示为:

$$P = \pm S \times 2^{\pm N}$$

其中 P, S, N 均为二进制数。S 称为 P 的尾数,表示成定点小数;N 称为 P 的阶码,表示成定点整数。

由此可知,浮点数表示中的各部分又都是定点数。不管是定点整数还是定点小数,由于它们的小数点位置固定,因此在运算时可以不考虑小数点的位置,只要区分是定点整数还是定点小数就可以了。

在计算机中,对于有符号的定点数又有三种表示法:原码、反码与补码。

1. 原码

原码表示中,符号位在最高位,“0”表示正,“1”表示负,其数值部分按一般二进制形式表示。下面是原码表示的例:

$$\begin{aligned}
 (+50)_{10} &= (+0110010)_2 = (00110010)_{\text{原}} \\
 (-50)_{10} &= (-0110010)_2 = (10110010)_{\text{原}} \\
 (+127)_{10} &= (+1111111)_2 = (01111111)_{\text{原}}
 \end{aligned}$$

$$(-127)_{10} = (-1111111)_2 = (11111111)_{\text{原}}$$

在用原码表示时,两个异号数相加时很不方便。

2. 反码

正数的反码是原码本身,负数的反码为该数原码除符号位外各位求反(即“0”变为“1”,“1”变为“0”)。例如:

$$(+50)_{10} = (00110010)_{\text{原}} = (00110010)_{\text{反}}$$

$$(-50)_{10} = (10110010)_{\text{原}} = (11001101)_{\text{反}}$$

$$(+127)_{10} = (01111111)_{\text{原}} = (01111111)_{\text{反}}$$

$$(-127)_{10} = (11111111)_{\text{原}} = (10000000)_{\text{反}}$$

可以验证:一个数的反码的反码是原码本身。

3. 补码

正数的补码是原码本身,负数的补码是该数的反码最后一位加1。例如:

$$(+50)_{10} = (00110010)_{\text{原}} = (00110010)_{\text{补}}$$

$$(-50)_{10} = (10110010)_{\text{原}} = (11001110)_{\text{补}}$$

$$(+127)_{10} = (01111111)_{\text{原}} = (01111111)_{\text{补}}$$

$$(-127)_{10} = (11111111)_{\text{原}} = (10000001)_{\text{补}}$$

同样可以验证:一个数的补码的补码还是原码本身。

在计算机中,加减法运算可以统一化成补码的加法运算,其符号位参与一起运算,结果为补码形式,这是十分方便的。

在上述三种表示中,反码只是作为求补码过程的一种中间形式。为了运算方便,计算机中所有的定点数一般都化为补码表示。

最后必须指出:当数的绝对值超过所用的二进制位数允许表示的最大值时,就会发生溢出,从而造成运算错误。上述的例子中,均用八位二进制位表示一个有符号定点数,因此,用补码表示的数的范围是: $-128 \sim +127$, 如果所要表示的数超过这个范围,就要溢出。如果用十六位二进制位表示数,则用补码表示的数的范围是: $-32768 \sim 32767$ 。

三、字符编码与汉字编码

1. 字符编码

目前微型机中普遍采用的字符编码是 ASCII 码。它是用七位二进制数对 127 个字符进行编码,其中前 32 个是一些不可打印的控制符号。

在 ASCII 编码中,十个数字符号、26 个大写英文字母、26 个小写英文字母是分别连续编码的。因此,如果知道了一个数字符号的 ASCII 码,就可推出其它数字符号的 ASCII 码。同样,对于英文大写字母与小写字母的 ASCII 码也是如此。

2. 汉字编码及字模信息

汉字有两种编码:国标码与机内码。

国标码是“中华人民共和国国家标准信息交换汉字编码”,代号为“GB2312-80”。在国标码的字符集中,收集了一级汉字 3755 个,二级汉字 3008 个,图形符号 682 个,共 7445 个。一个汉字对应一个“区位码”,由四位数字组成,前两位数字为区码(0~94),后两位数字

字为位码(0~94)。

机内码是指汉字在计算机中的编码。由于区位码中的区码及位码与基本字符的 ASCII 码有冲突,因此,汉字的机内码与区位码稍有区别。

汉字的机内码占两个字节,分别称为机内码的高位与低位。它们与区位码的关系如下:

机内码高位=区码+A0H

机内码低位=位码+A0H

例如,汉字“啊”的区位码为“1601”,区码为 16,位码为 01,如果分别用十六进制表示,则区码为 10H,位码为 01H。因此,该汉字的机内码为 B0A1H,其中高位为 B0H,低位为 A1H。

汉字字库是由所有汉字的字模信息构成的。一个汉字字模信息占若干字节,究竟占多少个字节由汉字的字形决定。例如,如果用 16×16 点阵表示一个汉字,则一个汉字占 16 行,每行有 16 个点,在存储时用两个字节存放一行上 16 个点的信息,对应位为“0”表示该点为“白”,“1”表示该点为“黑”。因此,一个 16×16 点阵的汉字占 32 个字节。同样,32×32 点阵的一个汉字要用 128 个字节来存储。其它规格的汉字可以此类推。

在输出一个汉字时,首先根据该汉字的机内码找出对应字模信息在汉字库中的位置,然后取出这些字模信息作为图形在屏幕上显示或在打印机上打印输出。

1.1.3 DOS 操作系统概述

一、DOS 系统的组成及功能

这里所说的 DOS 操作系统是指美国 Microsoft 公司为 IBM PC 微机及兼容机开发的单用户、单任务磁盘操作系统,即 MS-DOS。

MS-DOS 主要由三个程序模块组成,它们分别是:输入输出系统(IBMIO.COM)、文件管理系统(IBMDOS.COM)与命令处理程序(COMMAND.COM)。其中输入输出系统还包括了直接与计算机硬件打交道的软件模块 BIOS。

IBMIO.COM 与 IBMDOS.COM 是 DOS 系统的两个重要模块,它们是以隐含的方式存放在 DOS 系统盘上,一般用户看不到这两个文件。

在模块 COMMAND.COM 中,包括了 DOS 中所有内部命令的处理子程序。除此之外,它还具有以下功能:

- (1) 对用户输入的 DOS 内部命令进行解释并执行;
- (2) 对错误中断和键盘中断进行处理;
- (3) 负责将外部命令的命令文件装入内存并执行。

DOS 命令分为内部命令及外部命令。内部命令包括在模块 COMMAND.COM 中,因此是常驻内存的。外部命令存放在磁盘中,只有当要执行外部命令时,才将相应的命令文件调入内存并执行,执行完后,内存中就不再保留该命令文件。

二、DOS 启动

DOS 的启动方式有三种:加电冷启动、按主机箱面板上的“RESET”键及按组合控制键 Ctrl-Alt-Del 热启动。

其中前两种启动过程是一样的,而后一种启动不经过系统自检。

DOS 系统的启动过程如图 1.1 所示。

三、DOS 键盘的使用

在 DOS 状态下,除了一般的字符键外,还定义了一些具有特定功能的键,主要有以下两种。

1. 组合控制键

组合控制键为用户提供了对系统运行进行某种干预的机会,它一般由两个或三个键同时动作组合而成。

DOS 常用的组合控制键如下:

Ctrl-Alt-Del	热启动 DOS 系统。
^ C 或 Ctrl-Break	终止当前操作。
^ P 或 Ctrl-PrtSc	将标准输出同时送到屏幕和打印机。
Shift-PtrSc	在打印机上产生屏幕的硬拷贝。
^ S 或 Ctrl-NumLock	暂停标准输出设备的输出。
^ H 或 BackSpace	退格并删除一个字符。
^ J 或 Ctrl-Enter	结束物理行但不结束逻辑行。

2. 功能键

DOS 功能键又称 DOS 编辑键,用于对 DOS 命令的编辑修改。

F1	复制输入缓冲区中当前位置上的一个字符到屏幕。
F2<字符>	将输入缓冲区中直到指定字符为止的内容复制到屏幕。
F3	将输入缓冲区中当前位置后的所有字符复制到屏幕。
F4<字符>	将输入缓冲区中除当前位置到指定字符之间的所有字符外,以后的字符复制到屏幕。
F5	输入的下一个命令行只存放在输入缓冲区中,而不执行这个命令。

四、DOS 磁盘文件

1. 文件与文件名

互相有关联的信息的集合称为文件。每个文件必须有一个名字,称为文件名。

文件名一般由文件标识符和文件扩展名组成,即文件名的一般形式为:

文件标识符. 扩展名

文件扩展名又称为后缀。

文件标识符可以由 1 至 8 个 ASCII 码字符组成,这些字符可以是:

- 10 个数字符号 0~9;
- 26 个英文字母(大小写等价);
- 特殊符号: \$, #, @, -, ! 等。DOS 规定,在文件标识符中不能出现字符 <, >, \ 和空格等。如果文件标识符中的字符个数超过 8 个,则超过部分无效。

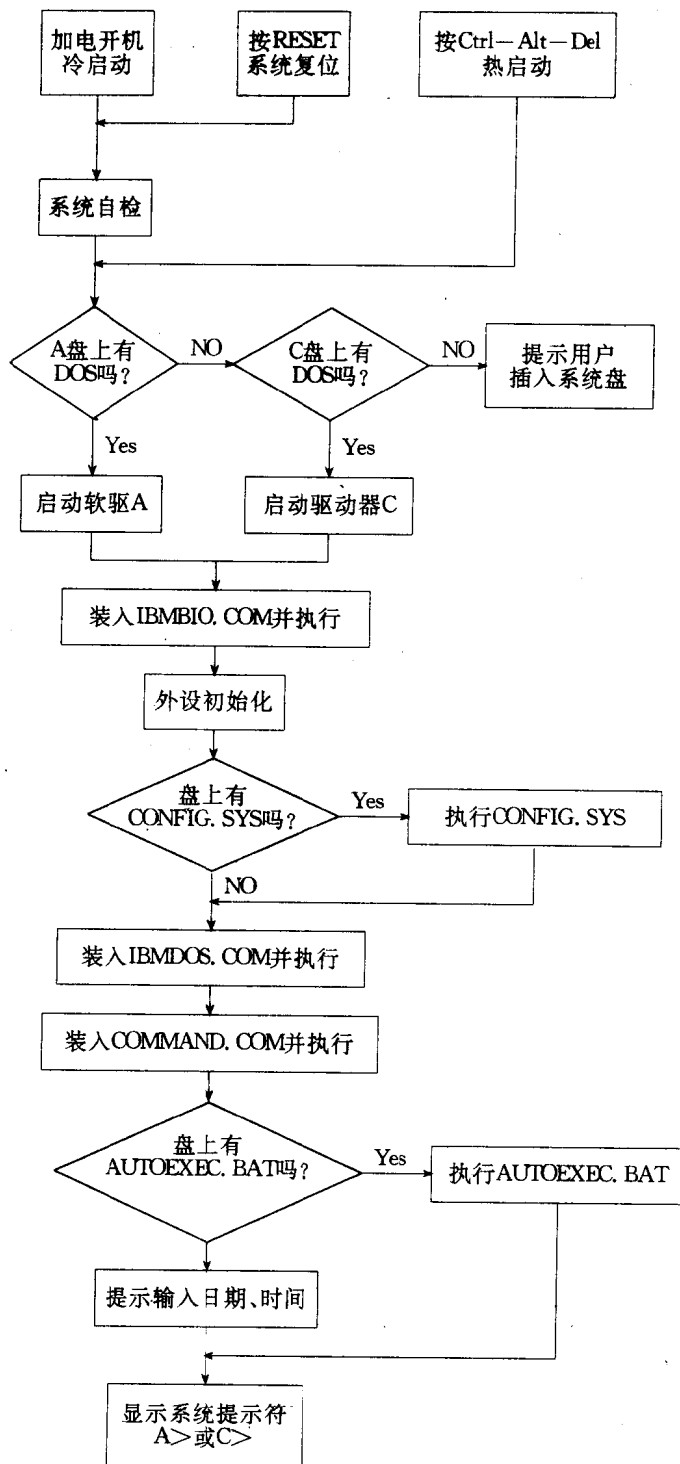


图 1.1 DOS 系统启动过程