DOS 6.x 高级编程

肖力 编著

人民邮电出版社

DOS 6.x 高级编程

肖 力 编著

人民邮电出版社

内 容 提 要

本书介绍在 386 和 MS－DOS 6.0 以上的微机系统上充分利用硬软件资源进行高级编程的各种技术。

全书分以下四部分内容：第一部分介绍硬件基础和 DOS 知识，分别讲解了 80386 的硬件特性、硬件调试手段、DOS 6 和 EMS 提供的调用。第二部分介绍 386 的实模式与保护模式互相切换技术和如何使用扩充内存和扩展内存技术。第三部分介绍直接存取、访问 CD－ROM 和使用 VESA 技术，讲解查询和事件驱动方式的鼠标程序、设备驱动程序和 TSR 程序的结构和设计方法。第四部分介绍 Windows 中的 DOS 程序与 Windows 通信的方法。

本书从基础知识讲到高级编程，内容丰富、叙述详尽，有大量实例 程序可供读者剖析和调用。

本书可供微机软件开发人员参考也可作为大专院校计算机和应用专业高年级师生和研究生的教学参考书。

DOS 6.x 高级编程

肖 力 编著

责任编辑 吕晓春

*

人民邮电出版社出版发行

北京朝阳门内南竹杆胡同 111 号

北京顺义兴华印刷厂印刷

新华书店总店科技发行所经销

*

开本：787×1092 1/16 1996 年 4 月 第一版

印张：42.25 1996 年 4 月 北京第 1 次印刷

字数：1064 千字 印数：1－4 000 册

ISBN 7-115-05981-0/TP・268

定价：45.00 元

前　言

随着计算机技术的发展，计算机的应用越来越广泛。它不仅已深入到工业、农业、商业、金融、医学、国防、科研等各领域，而且已进入人们的日常生活。可以预见，计算机技术将在我国国民经济建设和人民生活当中发挥越来越大的作用。

人们在使用计算机时，除了要了解计算机系统的基本硬件结构外，还必须熟练掌握计算机的操作系统。操作系统是系统软件的核心和最基本部分。它是整个计算机系统的控制管理中心。操作系统统一控制、管理计算机硬件和软件；其中也包括对其它系统软件，如编辑程序、编译程序、连接程序等的管理。随着计算机体系结构和使用方式的发展，各种操作系统也随之产生和发展。目前在微型计算机上使用的操作系统有很多种，MS－DOS 是美国微软公司为 IBM PC 机研制的磁盘操作系统，从 1981 年开发了 MS－DOS 1.00 版本以来，至今已发展到 MS－DOS 6.22 版。

为了帮助计算机应用和开发人员更充分地利用计算机的硬件资源，我们编写了这本《DOS 6.x 高级编程》。它是从计算机硬件和 DOS 基础知识讲起，一直讲到高级编程。我们希望它作为微型计算机软件开发人员和大专院校计算机和应用专业高年级师生的参考书，能对大家的学习、工作与研究有所帮助。

对于本书中的疏漏和差错，敬请读者指正，我们不胜感谢！

编者

1996 年 1 月

目　录

第一章

PC 硬件概述

在一本关于软件的书中，第一章就来谈论硬件，这似乎有点不合逻辑。但是，为了有效地对PC 编程，就必须懂得有关的硬件知识。在下面的几节中，我们将回顾 8086 系列处理器的一些基本的结构特点，并介绍 PC 机的硬件。如果你经常使用的高级语言隐藏了这些细节的话，那么你就会觉得开始的几节特别有帮助。

1.1 基本寻址

8086 型处理器把内存组织成 8 位的字节。如果涉及的数大于 8 位，8086 就把最低有效字节存放在最低内存地址。尽管这听上去很合理，但是在读内存卸出时常常会令人迷惑，因为数值似乎是倒序的。例如，计算机这样存放 B800H 这个字：先存放 00H，然后是 B8H。

Intel 系列处理器采用分段的内存寻址技术。简单地说，一个段就是一个内存区，计算机可以处理多个段。在实址模式(DOS 运行的模式)中，每个段是严格的 64KB 长，一共可以有 65536 个段。段与段之间可以重叠，一个段可以从上一个段的 16 字节后开始。这就是为什么 DOS 不能直接访问大于 1MB 内存的原因(65,536 * 16＝1048576 即 1MB)。8086 和 8088(用于 XT 级机器上的处理器)只能寻址 1MB，286、386 和 486 处理器可以支持更大的内存，但 DOS 不能直接访问它们。

段的编号从 0000H 到 FFFFH。既然每个段都是 64KB 长，我们就可以用一个称为偏移量的值来定位要寻址的字节。一个完整的 8086 地址总是包括一个段值和一个偏移量。如果段值是 0040H，偏移量是 0102H，那么地址就被写作 0040:0102。因为相邻段之间间隔 16 个字节(10H)，所以 0000:0010 与 0001:0000 表示同一地址。同样，0040:0000 与 0000:0400 是同一地址，0020:0200 也表示这一地址。计算机总是倒序地存放地址。例如，0040:1234 在计算机的内存(以 16 进制表示)中是这样的：

34 12 40 00

程序通过寄存器和常数的组合来形成地址。鉴于此，我们将在考查了处理器的寄存器之后再详细地讨论寻址问题。

1.2 内存类型

用户程序可以使用几种不同类型的内存。就像我们刚看到的，XT 级计算机只能寻址 1MB 内存，这种内存称为常规内存。XT 可以有 640KB 的 RAM，而 I/O 设备、BIOS ROM 和其它系统资源使用其余的 384KB 内存。

当程序需要的内存空间开始超出这个范围时，Lotus、Intel 和 Microsoft 一起建立了扩充内存规范(EMS)。你可以看到 EMS 通常和它后面的版本号一起出现，像 EMS 3.2 或 EMS 4.0(目前最新的版本)。绝大多数的 EMS 系统包括一个或多个 EMS 卡和一个称之为驱动程序的软件。程序可以通过驱动程序请求使用 EMS 内存，可是这种内存在某一时刻只能访问 64KB。EMS 驱动程序把这 64KB 块放在 1MB 地址空间的高端，通常就在紧邻着 BIOS 的下面。通过调用驱动程序，程序可以把 EMS 内存 16KB 的"页"映射到这个 64KB 的块中。

即使 EMS 可以处理的内存多达 32MB，但在某一时刻，程序只能访问有限数量的 16KB 的页。尽管这不太方便，但是扩充内存是 XT 级机器可以使用的唯一的一类附加内存。如果程序使用其它类型的附加内存，那它们就不能在基于 8088/8086 的机器上运行。

EMS 也可以由磁盘来模拟，这不需要额外的硬件，但是与硬件的 EMS 系统相比速度太慢。

在 286，386 和 486 上，1MB 的内存并不是上限。286 可以寻址多达 16MB；386 和 486 可以寻址多达 4096MB(4GB)。从技术上讲，这种扩展内存只能在保护模式下访问。但 BIOS 提供了一个功能，数据可以在任意的两个地址之间相互传递，包括扩展内存中的地址。

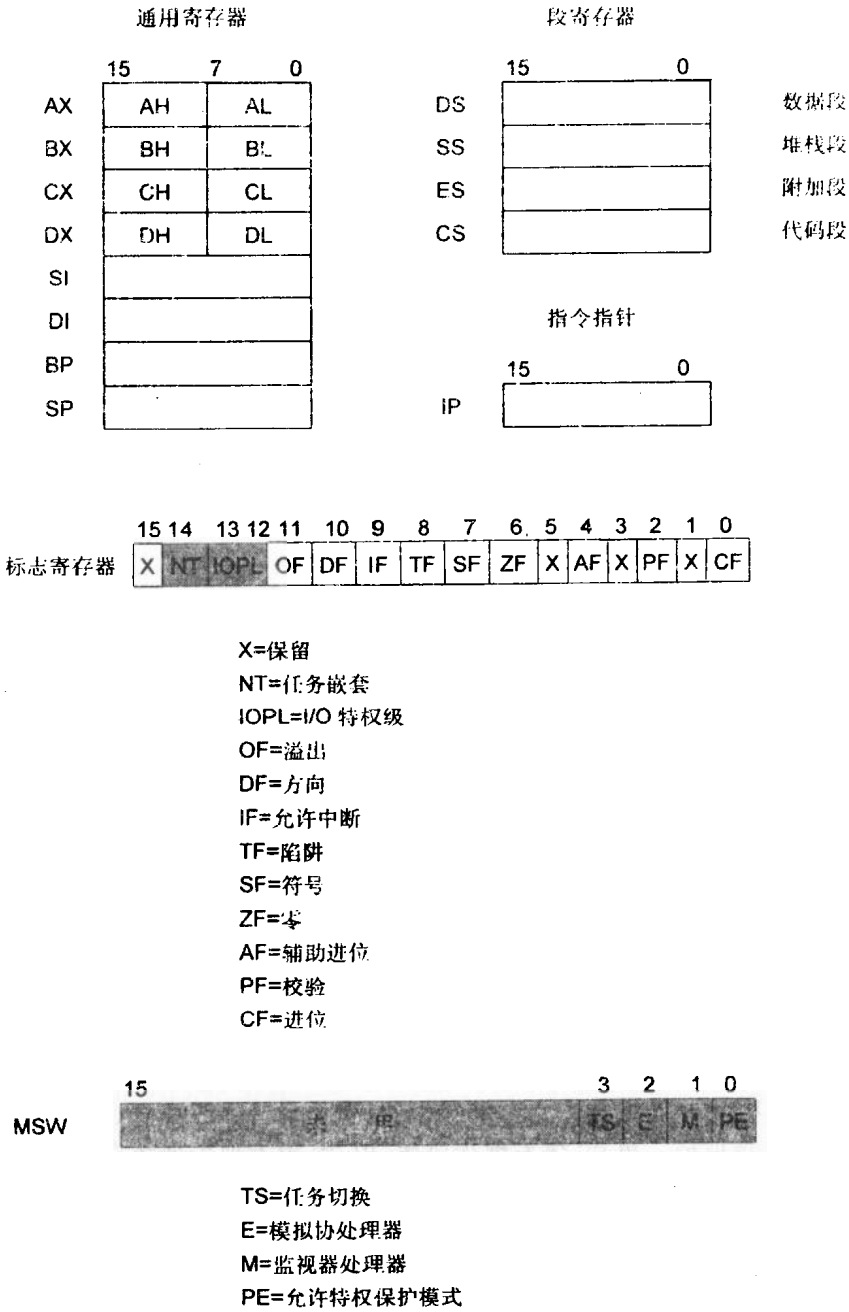
扩展内存规范(XMS)驱动程序用一个巧妙的寻址技术(将在后面讨论)允许 DOS 下的应用程序直接访问扩展内存的将近 64K 的区域。重要的是要注意在保护模式下，常规内存和扩展内存并没有区别，扩展内存只不过是从 1MB 地址开始。用软件可将扩展内存转换成扩充内存，在 286 上这一点不是十分有效，并且不能模拟所有的 EMS 特性。可是，386 和 486 有强大的内存管理功能，EMS 可以完整有效地用软件来实现。

1.3 寄 存 器

8086 有四个通用寄存器、一个标志寄存器、四个段寄存器、二个索引寄存器、一个堆栈段寄存器和一个堆栈指针寄存器、一个基址寄存器和一个指令指针寄存器。图 1－1 给出了这些寄存器以及 8086 中的其它寄存器；图 1－2 给出了在 386 和 486 中的寄存器。注意 286、386 和 486 有多种专用的寄存器。虽然它们在 DOS 下不是十分有用，但当我们讨论 DOS 的保护模式扩展时将会用到它们。

每个通用寄存器都是 16 位的。也可以把其中的任何一个用作二个 8 位寄存器。例如，AH 和 AL 事实上就是 AX。AX、BX、CX 和 DX 是通用寄存器：

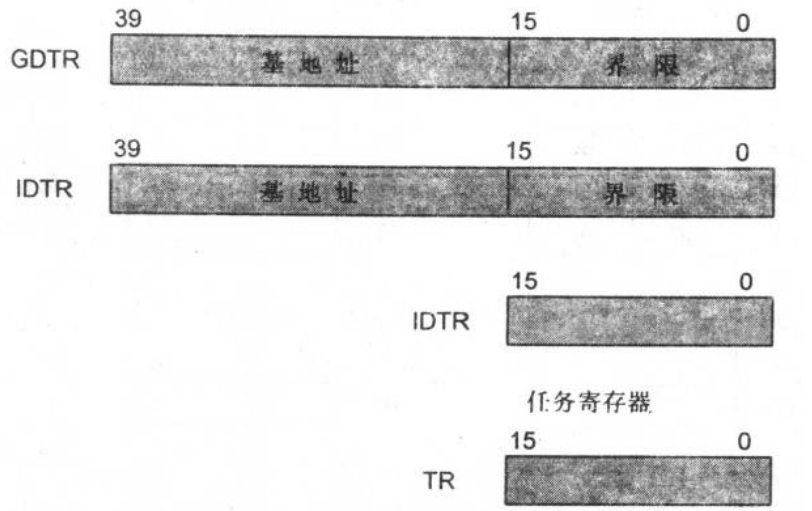
- AX 有时也称为累加器，用于算术运算。
- BX 是唯一能在实址模式中参与构成地址的通用寄存器。
- CX 用于循环或重复操作的计数。



注：带阴影的寄存器和域仅在 286 上存在

图 1－1 8088/8086/80286 寄存器(a)

• DX 指定 I/O 地址，特别是大于 0FFH 的 I/O 地址。它还在乘法和除法指令中与 AX 一起使用。AX 寄存器的两个 8 位寄存器。在图 1－1 中 AX 是 8001H，这就意味着 AH＝80H，AL＝01H。386 和 486 有同样的通用寄存器，只不过那些寄存器是 32 位的。32 位的寄存器都以字母 E 开头。像图 1－2 当中，EAX 是 72018001H，那么 AX＝8001H，



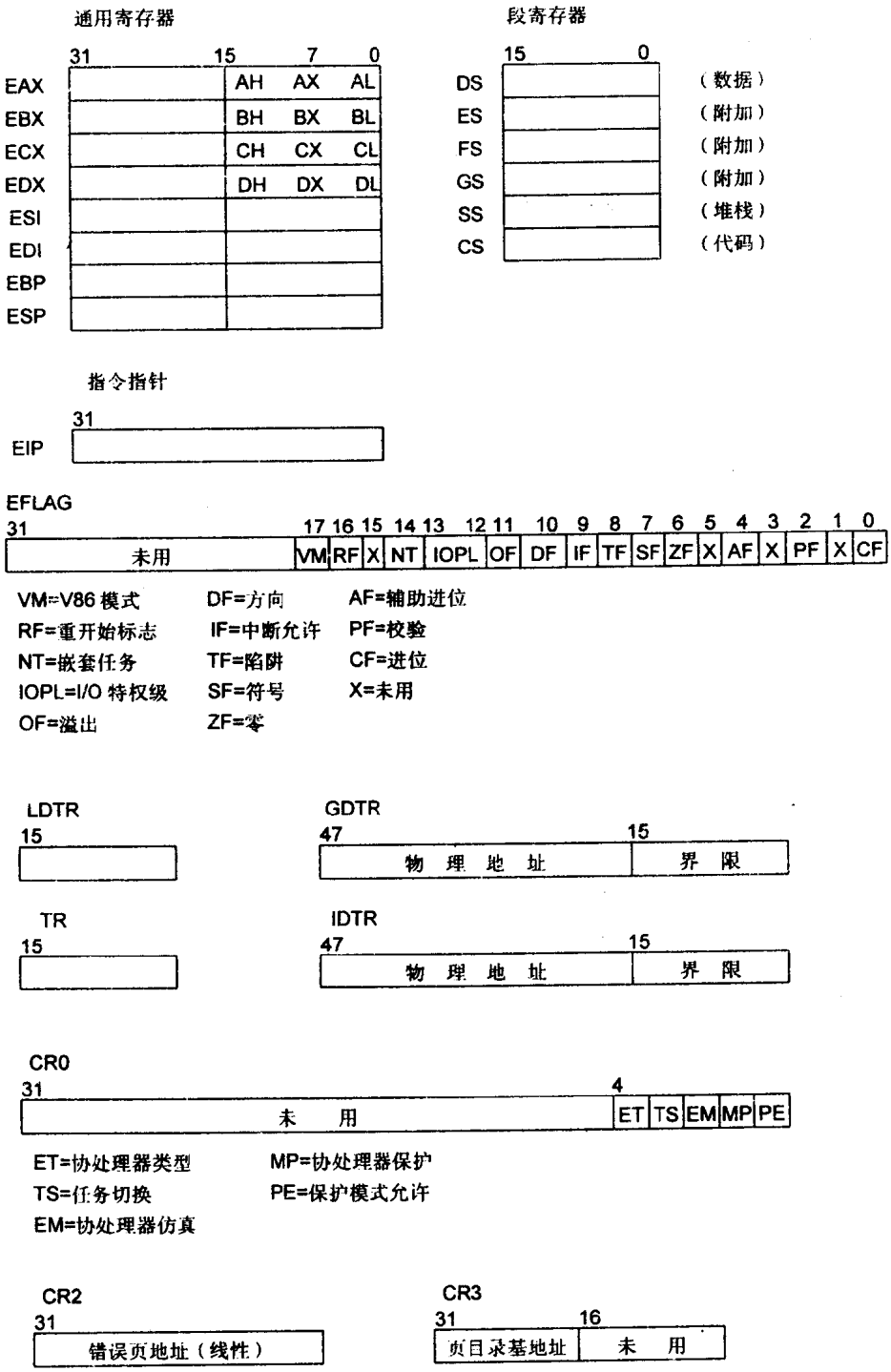
注：带阴影的寄存器和域仅在 286 上存在

图 1-1 8088/8086/80286 寄存器(b)

AH=80H，AL=01H。不能直接访问 32 位寄存器的高 16 位。

理论上讲，通用寄存器可以用于任何操作。但是每个寄存器都有某些特定功能。有时对于某种特定的操作必须使用某个寄存器，有时使用某个寄存器会比用其它寄存器更有效。

- 段寄存器(CS、DS、SS 和 ES)用来指出当前代码段(CS)、数据段(DS)和堆栈段(SS)，ES 是个附加段，用于段间数据传输。这些段像下面描述的那样参与地址计算。386 和 486 处理器还有二个附加段寄存器：FS 和 GS。
- 二个索引寄存器：SI 和 DI。主要是参与形成地址。与通用寄存器不同，SI 和 DI 是不能分成二半的。在 386 和 486 上，可以使用 ESI 和 EDI 来形成 32 位地址，或用 SI 和 DI 形成 16 位地址。
- 8086 系列提供一个堆栈供 CPU 和程序员使用。堆栈作为数据临时存储区，在子程序调用和中断功能中起重要作用。堆栈段寄存器(SS)和栈指针(SP，或 386/486 中的 ESP)给出栈顶地址。用 SS 寄存器作段值，用 SP 寄存器作偏移量，就能确定当前堆栈的栈顶。
- 基址寄存器 BP(或 386/486 中的 EBP)指向栈中的项。在许多方面，它与 SI、DI 寄存器相似，但我们对寻址方式稍加观察就能看出两者之间有明显的差别。
- 指令指针寄存器(IP)给出下一条指令在当前代码段中的偏移量。尽管跳转指令或子程序调用会修改 IP，但用户不能直接修改它。当然，在 386 和 486 中，32 位方式时使用 EIP。
- 标志寄存器包含一系列确定处理器状态的标志位。某一位可以表示上次算术运算是否溢出，另一位可以控制处理器对外部事件(或称为中断)的响应。程序通常用特定的指令设置或清除某一位来改变这个寄存器。



注：图中不包含调试和测试寄存器

图 1－2　80386/80486 寄存器