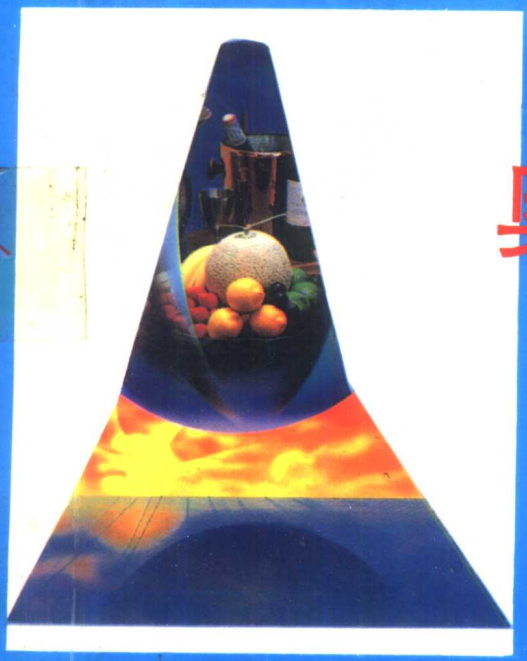
李生乐
李 勇 编著

DOS 多任务 de 奥妙及实例剖析



·電子工業出版社·

DOS 多任务的奥妙及实例剖析

李生乐　李　勇　　编

电子工业出版社
1994

(京)新登字 055 号

内 容 简 介

本书主要讲述 DOS 开发者始终未公开的 DOS 多任务技术，通过对 DOS 实用程序 PRINT.COM 的详细剖析，揭示了多任务程序的编写要点、策略以及对 TSR 程序编写的奥秘，尤其对于理解 WINDOWS 和市面上的许多实时通信软件有重要的参考作用。为便于读者阅读，书中还详细介绍了 DOS 多任务有关的软、硬件知识和一些未公开的 DOS 功能调用。

本书适于计算机软硬件开发人员参考使用，也可作为高年级本科生和研究生的教学参考资料。

DOS 多任务的奥妙及实例剖析
李生乐 李 勇 编
责任编辑：王雨寒 向 元
电子工业出版社出版(北京万寿路)
电子工业出版社发行 各地新华书店经销
双青印刷厂印刷
开本：787×1092 1/16 印张：27.625 字数：642 千字
1994 年 4 月第 1 版 1994 年 4 月第 1 次印刷
印数：1～5000 册 定价：29 元
ISBN7－5053－2539－6/TP.754

绪　论

本书是经过全面剖析 PRINT.COM 后所整理的详细资料合成本。DOS 是在一个 CPU 运行下的单用户操作系统，但是 DOS 的功能调用 INT 21H 是不可再入式的功能调用，DOS 的进程管理是单任务性的管理。如果想在 DOS 系统支持下编写多任务性的应用程序，这对于程序开发者来说有一定的难度。尽管 DOS 是单任务性的进程管理系统，但利用 CPU 的时间片，是可以完成多任务性的功能的，关键是这种功能的应用程序怎样设计。在 DOS 状态下的时间片太多了，比如一个录入员在录入数据，一个办公人员在编辑一篇文章，一程序员在编写一个程序，.....等等，这些在键盘操作的时间等待状态下，时间片非常多。有人统计这种状态下 CPU 有 80% 的空闲时刻片。DOS 本身系统开销小，占用资源少，是 DOS 的一大特点，在 DOS 支持下的 WINDOWS 3.0 及 3.1 版本的系统，是非常著名而且成功的在一个 DOS 环境下的多任务系统。PRINT 和 WINDOWS 的多任务管理是大同小异的，WINDOWS 管理大体分为三部分：(一)用户(USES)部分，这一部分负责窗口管理；(二)核心(KERNEL)部分，这一部分负责系统服务，多任务、多资源(基本内存、扩展内存、扩充内存、外设、硬盘等)管理；(三)图形设备接口(GDI)部分，这一部分有图形设备接口，从应用程序传到设备驱动程序、字符操作位图(BITMAP)进行输出。上面三部分中的第二部分——核心部分，其 DOS 再入技术思想和 PRINT 中的技术思想是基本类似的。掌握了这种技术，用户就可以编写自己的 DOS 系统下的多任务性系统。大家知道，尽管在 PC 机开发出了一些多用户操作系统，如 XENIX 等，由于 DOS 的可扩充性及 DOS 的应用软件非常之多，DOS 的寿命还难以估算，有关专家声言 DOS 还没有后继者，特别是 WINDOW 3.0 及 3.1 的出现，更加延长了 DOS 的寿命。WINDOWS NT 能否代替 DOS 使人们还在拭目以待。

在 DOS 下完成多任务性的系统设计涉及到许多软、硬件知识及一些 DOS 未公开的功能调用，为了使本书自成系统，编者把有关软、硬件知识一一介绍，那些 DOS 未公开的中断调用，在附录二中给出。实践证明，使用未公开的调用确实能完成一些已公开的调用难以完成的功能，且兼容性也好。相反，若用户自己费尽心思编写一些难度较大的程序模块反而兼容性不好。

本书第一章叙述了中断系统。在第一节中详细介绍了各种类型的中断，及 DOS 中断与 XENIX 中断的区别。这些中断包括内中断，外中断(硬中断)，软中断。第二节介绍了 8259A 中断控制器及其应用：8259A 芯片的功能，硬中断的过程，中断命令字 ICW 的初始化，8259A 的各种工作方式，最后是 8259A 的应用举例，从软件编程上可以清楚地看到怎样应用 8259A 芯片。因为在 DOS 多任务性的系统中少不了利用 8259A ，检查 8259A 的状态等一系列对 8259A 的操作。因此，本章是硬件知识的基础。

第二章详细描述了时钟系统。第一节介绍了可编程定时器/计数器的作用，以 8254 芯片为例叙述了其逻辑功能及工作方式，对其三个通道(蜂鸣器、内存刷新、时钟定时中断)通过程序举例作了详细地介绍。从硬件角度描述了可编程定时器/计数器的作用。第二节对时钟中断作了详细地介绍，实际上是对可编程定时器/计数器的通道——时钟定时中断通道的加细，这种加细通是过系统的时钟定时中断服务例程 ISR 来说明其在系统中的作用。当然用户也可以加载自己的时钟定时中断服务例程 ISR。这一部分实在是太有必要搞清楚了，特别是对 DOS

用户的程序开发者而言，DOS 的多任务性系统设计是离不开这个中断的，请读者要特别清楚地掌握这个中断服务例程的编写。

第三章叙述了有关的中断。第一节介绍了硬中断事件发生时应采取的策略。第二节是对硬中断的加细，这种加细也是通过一个硬中断编程举例来说明的，这个硬中断是一个实时通信接收程序，其中断号为 INT 0CH 或 INT 0BH，当然这种中断类型还有键盘中断 INT 9 等等。对于类似这种串型口发生的中断，本例子用一个环形队列 FIFO 来管理数据，对通信规程的控制、管理也是通过这个环形队列 FIFO 来实施。本例子还包括对系统键盘环形缓冲区的队列 FIFO 进行操作的子程序，通过这些例子使读者能举一反三、融会贯通。本节详细叙述了硬中断的的全过程：即中断处理两大过程。本章程序中还有 INT 21H 的子功能 4BH 调用举例，相信这些例子对读者会有益处。第三节对 PRINT 中用到的有关 DOS 未公布的中断调用作了详细地介绍，这些中断有 INT 28H，INT 2FH，INT 21H 的 34H 子功能，INT 21H 与 52H 子功能等等。有关其他的未公布的中断功能在附录 B 中有详细介绍。请对此感兴趣的读者阅读附录 B。

第四章介绍了设备驱动程序。关于设备驱动程序的书不少，也很厚，但与本书所写的设备驱动程序部分有所不同。本书是针对怎样利用设备驱动程序而写的，通过详细分析 DOS 的设备驱动程序的有关程序及参考大量的资料，彻底搞清楚了怎样利用设备驱动程序对外设操作。DOS 程序开发者都知道，对设备如屏幕、打印机、磁盘的操作一是通过 DOS 的功能调用，二是通过 ROM 中的 BIOS，三是自己编写程序，再就是利用设备驱动程序。对于前三种，特别是前两种，大家用的很多，但对于最后一种大家就不怎么用，甚至不太会用，本章正是为解决这个问题而写的，请大家注意这一技巧。应该在今后的工作中对这一技术的应用引起重视。因为利用设备驱动程序，一是兼容性好，二是速度快，三是避开了 DOS 的直接调用，即避开了 DOS 的再入问题，国外有很多软件都利用设备驱动程序。PRINT 正是利用设备驱动程序对设备操作，我国的 DOS 应用程序开发者应重视此问题。一般对外设操作通常是用户程序→DOS→DOS 设备驱动程序→ROM BIOS，或者是用户程序调 BIOS，后一种操作不能充分利用 DOS 的资源，例如文件操作不利用 DOS 而直接利用 BIOS 就不能形成磁盘文件。设备驱动程序起到一个桥梁作用。对怎样编写设备驱动程序，本章有详细的说明，以便于读者编写自己的设备驱动程序。

第五章详细介绍了程序段前缀 PSP 的结构及其作用。第一节详细叙述了 PSP 的结构，用汇编风格给出了其数据结构，且对各个字段(FIELDS)作了详细的介绍。第二节对 PSP 应用作了各种举例，读者可举一反三，读完本章后就会对 PSP 中的各个字段应用自如。本节列举了 PSP 的获取、PSP 的设置，这二个功能可以构成上下文切换(Context Switching)、各种参数的获取、TSR 内存释放及其他应用等等。这一章非常实用，PSP 是(且在 DOS 多任务系统设计中是)进程资源的管理一张重要的参数表，是各个进程的一个重要标志。请读者重视。

第六章介绍程序的可重定位形式。.EXE 和.COM 文件结构不同，加载方式也不同，看了本章后读者会清楚这两种不同的可重定位方式。这对于编.COM 文件及.EXE 文件的 TSR 非常有用。理解.EXE 型文件的重定位问题，对于程序的二次开发，特别是对于程序的加密、运行程序的压缩等特别有用，可以这样说，不会利用.EXE 型文件的重定位技术是设计不出优秀的软件加密系统的。

第七章对多任务(Multi-tasking)程序的编写要点、策略以及对 TSR 的程序编写作了详细

介绍。介绍了基于过程的(Process based)和基于线索的(Thead of execution)多任务实现的概念,介绍了多任务内核(Multi-tasking kernal)算法。TSR 程序为大家所熟悉,我国著名的 CCDOS 就是一个人人皆知的 TSR 型实用程序,还有不少 TSR 系统,例 WINDOWS,记事本,调试器,保护方式下 DOS 扩展器多任务系统,网络支持系统等等,但多任务性的 TSR 编写要点却鲜为人知。本书通过全面地剖析 PRINT 及阅读大量的国内外资料,把编者所理解的一些要点毫无保留地献给读者。第一节叙述 DOS 的单任务性,当然这种单任务性是由单 CPU 所造成的。但是单 CPU 运行多任务靠时间片,因为在多 CPU 实行多任务时,当各个 CPU 都不空闲时,就靠时间片支持完成再来的任务。第二节介绍了终止并驻留 TSR 实用程序。第一部分介绍了中断服务例程 ISR ,这和第一章不重复,因为 TSR 中大都是中断服务例程为主体部分。第二部分通过一个 TSR 举例来说明中断服务程序在 TSR 中的一些重要作用,这个例子是一个非常好、非常实用的工具软件,也是作者多年来爱不释手的工具。通过这个例子,读者可以从中得到益处,也为以后理解 PRINT 程序打下基础。这个例子确实是能看得见的例子,读者只不过要花点时间录入磁盘编译连接即可。这个软件被命名为 MONITOR,因为利用它可以动态观看内存及 CS:IP 值。第三部分介绍了常驻内存实用程序,这种实用程序与中断服务例程型的 ISR 有区别。第三节是本章的重点,着重介绍多任务性的 TSR 程序编要点(这是编者的理解)。很不幸的是,TSR 程序的编写标准直到现在还未产生,但只要遵循这些要点就能设计出多任务性的 TSR 系统。这一节还介绍了定时中断 INT 1CH、INT 8、DOS 的三个堆栈区(3.0 以上版本),DOS 的重入问题,PSP 上下文切换(Context Switching),DOS 的忙标志、中断状态控制的检查(结合第一章看更清楚),磁盘 ROM BIOS 中断 INT 13H,DOS 的错误危机处理,其他中断:INT 17H、INT 15H、INT 14H,扩展内存管理等等.这些都是在多任务性系统中要考虑的问题,一个都不能遗漏,否则系统将出现意想不到的问题。第四节讨论了 TSR 的执行问题,包括终止并驻留、激活、挂起、休眠等问题。激发的方法:利用 INT 16H,INT 15H,INT 1CH,INT 8H,INT 9,INT 28H,键盘上的右左 SHIFT 键,CTRL 键,ALT 键等激发方式。当然也可以用 RS232 口的中断源来激活,例如 MODEM 线路、鼠标及软件条件激发(病毒程序就是一个 TSR 型程序)等等。第五节讨论了一个非常有用但都被人们忽略开发的问题:TSR 的撤离问题,一般地撤离仅是编写一个对自己的 TSR 撤离问题,但对通用的 TSR 撤离问题,编者给出了一个模型,提供了一个成功的程序 RTSR,当然这个程序也不一定是完善的。编者经常看到有些用户对 TSR 的撤离竟然用热启动或冷启动,再次启动、引导机器,这对计算机资源、时间实在是太浪费了。有了这种软件,只需按一下键便即刻释放用户想要解放的 TSR 。编者在这一节对这个软件的编写也毫不保留地给出了自己的理解及体会。RTSR 程序举例是编者在本节给读者的一个很好的礼物。

第八章是对 PRINT.COM 的总体分析及程序框图。尽管有些资料、书籍中也有些零散地分析说明,但就其内部实质性、关键的技术没有论述。这一章对各个子程序的框图及详细的流程加以说明。

第九章论述的是本书的实质性问题——PRINT 的全面剖析。以上所有章节都是为本章做准备而写的,对那些 DOS 开发高手可直接阅读上一章和本章即可。本章是对 PRINT.COM (DOS V3.3 版本)进行了全面地剖析,且是通过还原出源程序、调通后整理而成,对各个子程序都有详细的注解,各个模块间的关系,子程序的功能,甚至每条指令的功能有注解。这些注解是编者花费了多年的心血分析而成,现毫无保留地献给广大读者。这些模块包含:可重定位

部分，前台时间片分配，后台时间片分配，PSP 上下文切换，设备驱动程序的利用（包括设备输入，设备输出，屏幕 CON ，打印 PRN 等），后台的文件管理设备输出，后台的进程管理，时钟中断，INT 28H 时间片的利用，外设的管理，包括 INT 14H，INT 15H，INT 17H，INT 13H 等等，DOS 的错误捕捉，INT 2FH，DOS 的错误处理，后台的模块功能包括后台的设备操作及后台的文件操作等若干模块。当然还有一些外设的管理，在 PRINT 中没有处理。WINDOWS 的成功之处正是对 DOS 所不能管理的设备进行极大限度地管理，WINDOWS 系统本身也占用不少的资源，这些资源包括扩充内存，扩展内存等等。用户在设计自己的 TSR 时要用到这些硬件资源。

附录 A 列出了 DOS 的错误信息，DOS 的程序开发者应该重视这部分，通过这些标志，用户可以合理地处理 DOS 中出现的错误，也可合理地处理用户所设计的系统中出现的错误，以便不会在出错的情况下使系统出现异常，甚至崩溃。也可以这样说，系统即使是出现 DOS 中的错误，用户的系统也能正确地处理，使系统仍正常无误运行。这一点对多任务性的 TSR 设计是非常重要的。

附录 B 是有关 DOS 中未公开的中断详细列表，它是编者分析 DOS. BIOS 后及查阅了大量的中外文资料整理而成的，大部分中断已在 DOS 系统验证过。在术语方面，编者力求准确性、专业性，如 FILE HANDLE，以前大都译为文件句柄或文件把柄，但其实际含义就是 DOS 的一个文件逻辑指针，因此可把它叫做文件指针；LIST OF LISTS 有人把它译为表之表，表指针等，但其实际的含义是 DOS 的多重表的指针表，等等。有些术语可能与其他资料不一样，但看了本附录后，能使用各个功能调用，也就达到了目的。

附录 C 是系统的重要数据总结表，包括 BIOS 信息，CMOS 信息，DOSS 内存链结构，CRT 控制器信息，磁盘参数表等等。

附录 D 是本书的参考书目，读者如果需要详细阅读其他有关内容，可以到这些参考书中查阅。

前　言

尽管有关 DOS 的技术资料很多，甚至包含有 DOS 内核的源程序详解，但是有关 DOS 多任务的系统论述之书还很少见，DOS 多任务这个问题是一个专题，本书就是涉及这个专题的一本书。

在 DOS 系统上实现多任务功能，其奥妙始终未有公开，可是在 DOS 系统上实现多任务的软件并不少见。早期的有 PRINT，近期的有 WINDOWS，还有各个公司自己的多任务软件系统，特别是用于实时通信方面。PRINT 软件是一个 DOS 多任务软件经典的实例。DOS 的开发者直到现在始终没有公开这个秘密。尽管 PRINT 同 DOS 一并涌向市场。众所周知，DOS 调用是不可再入式的中断调用，不提供多任务的功能，但多任务处理的软件在 DOS 系统中确实有之。本书正是为揭开这个奥妙而抛砖引玉，使广大读者通过本书能更好地在 DOS 操作系统下编写自己的多任务软件，尽管多任务操作系统已在 PC 机上出现，但是 DOS 仍有极强的生命力，因为 DOS 用户的覆盖面广，DOS 的应用软件很多，且都成熟。因此我们下决心编写此书。

在 DOS 操作系统编写多任务程序时，涉及到许多软件、硬件、系统等问题，为了使本书更系统，且自成体系，我们把有关编写多任务系统的内容尽可能全面地收集进来一一介绍，特别是对 PRINT 程序。本书分析的 PRINT 程序是 DOS3.3 版所附属的。PRINT 中涉及的有关知识、编写多任务系统程序的要点、系统资源标志、及未公布的中断功能调用等，都与编写多任务程序有着重要的关系。它们包括可执行文件结构、中断系统、时钟系统、程序段前缀 PSP 、设备驱动程序、TSR 编写要点、系统中断服务程序(ISR)等，为了使读者减少查阅大量资料的麻烦，我们把所分析、搜集的一些 DOS 未公开的资料编入本书以给读者提供方便。我们还根据多年积累的资料整理和对 DOS 操作系统、BIOS 及一些相关软件的分析，较全面地把所有中断功能调用也写在该书中，目地在于不仅使该书为编写 DOS 操作下的多任务程序提供方便，而且也能使其成为一本 PC 机系列的中断功能调用速查手册，更加方便 PC 机的开发者。

为了使读者更详细地分析本书中的某些章节，我们把参考书目也列在本书后面，以便使读者去寻找另一些感兴趣的知识。本书中前面的章节完全是帮助后面分析 PRINT 源程序而写的，如果读者对有些章节很了解则可以跳过去，读后面的部分。阅读后面的内容有困惑时，也可到前面章节查阅有关的知识。这种关联也是我们写书时所考虑到的。

书中除 PRINT.ASM 程序详解外，还有我们在工作中开发的一些程序，例如利用时钟中断查看内存、TSR 动态撤离程序等。有些程序给出了源程序，有些程序则只是作为介绍，没有给出源程序。特别指出的是，编者在调通 PRINT.ASM 程序后，才着手此书的编写工作，并把这一技术用于前后台实时通信。这个实时通信程序，可以用于网络，也可以用于 PC 机接 Fax 或 PC 机接 PC 机的点对点通信，或者用于终端接电传机等等。这种技术还可用于实时控制。除这些外，我们还编写了 TSR 程序中利用硬件资源的软件，例如对扩展内存、扩充内存的管理等等。

读者可从本书得到下述问题的解答：

1. 多任务进程编写的内核技术。

2. 后台进程与前台进程的切换。
3. 后台进程如何执行磁盘的输入及输出操作。
4. 后台进程与前台进程时间片的分配。
5. 后台进程的激活、挂起、休眠。
6. 后台进程的动态撤离。
7. 后台进程的输入输出。
8. 后台进程应考虑的其他中断。
9. 后台进程的再入问题。
10. 后台进程如何解决 DOS 的再入问题。
11. 后台进程对 DOS 错误危机的处理。
12. 利用 INT 1CH 管理前台进程的时间片。
13. 利用前台进程的间隙执行 INT 28H。
14. 程序段前缀的一般应用及在 TSR 中的应用。
15. 设备驱动程序的应用。

由于我们的水平有限，书中难免有错漏之处，敬请广大读者批评指正。

编　者

1993 年于北京

目　录

第一章　中断系统

所谓中断就是由于特殊的原因而改变程序的流程，中断保证 CPU 在运行过程中随机地对外界发生的请求予以响应，在完成实时性的任务后立即返回被中断的断点，继续执行被挂起的例程。

为响应中断的速度，现代计算机采用了多级向量中断技术。向量中断的基本思想是：对应每一中断类型，在内存的特定位置上存放一个中断向量，该向量含有这种类型中断服务程序的入口地址。PC 机的中断向量表在 0 段的 0－400H 中，对于不同类型的中断赋予不同的优先级，即当几个中断同时提出请求时，CPU 优先响应级别最高的中断。为有效地管理外部硬件中断，系统通常使用一个中断控制器负责处理各级中断请求。这种处理机包括如下功能：

- 对不同类型的中断源赋予固定优先级或循环优先级；
- 随机选择对某些中断级的开放或禁止；
- 当执行某级中断例程时，所有同级或低级的中断均被屏蔽，直到当前的例程运行完毕。但是，高一级的中断请求能中断正在执行的低级例程，这样就实现了多级中断的嵌套。除外部的硬中断外，系统还能响应内部中断和软中断。

DOS 的功能中断服务程序是以软中断形式表现的，这些中断均可供系统或应用程序调用，这些中断功能包括：

- 程序结束处理；
- Ctrl-C 中止处理；
- 严重错误处理；
- 绝对磁盘读写处理；
- 系统功能调用(INT 21H)；
- 假脱机打印处理。

前 3 种属于非直接调用的向量，但应用程序可以利用系统功能调用中的若干功能建立用户自身控制的向量来取代 DOS 提供的向量，以便接受管理系统的控制权。DOS 服务功能中，最重要的中断向量是系统功能调用，它有近百个子功能供系统或应用程序调用，这些子功能请参阅本书的附录 B。

1.1　中断类型

80X86 机的中断类型分为 3 类：

- 内中断(中断源是内部硬件)；
- 外中断，也称硬中断(中断源是外部的硬件)；
- 软中断(中断源是中断指令)。

8086、80X86 为 CPU 微处理器，上述 3 类中断是用一字节编码，共有 256(0－0FFH)优先级中断。因为每个中断是段间调用，所以每个中断向量占 4 字节地址，这 4 字节是段地址：偏移

量，系统安排在最低端的 0 地址至 3FFH 为中断向量表，存放 256 个中断向量地址。各类中断的约定如图 1.1 所示：

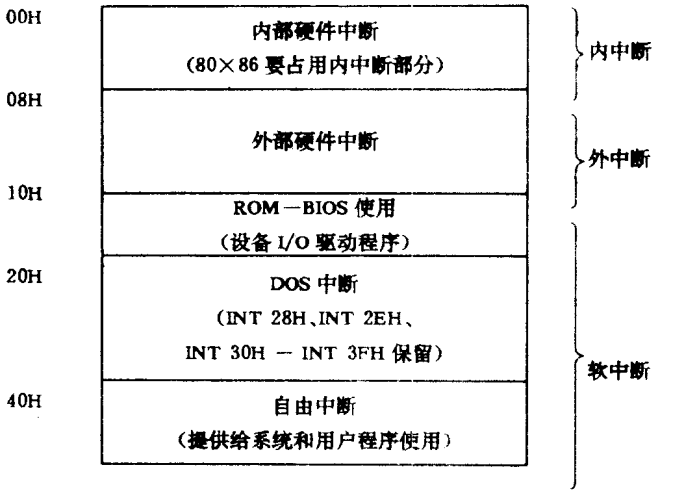


图 1.1　3 种中断类型的内存向量表图

1.1.1　内中断

内中断即在系统运行程序时，因硬件出错(内存奇偶校验错，突然掉电等错误)，或某些特殊事件发生(如除数为 0 、运算溢出或单步跟踪等)所引起的中断。

几个内中断：

INT 0　　除数为零中断

INT 1　　标志寄存器的 TF＝1 且执行一条指令后，执行 INT 1

INT 2　　不可屏蔽中断(NMI)，处理硬件错误

INT 3　　断点中断，DEBUG 中用

INT 4　　执行指令 INT0，当有溢出时(of＝1)INT 0 指令执行后，执行 INT 4

表 1.1 列出中断的向量地址。

表 1.1　中断向量一览表

中断号	向量地址		
00H	0:00H—03H	除数为零	
01H	0:04H—07H	单步跟踪	
02H	0:08H—0BH	不可屏蔽中断	
03H	0:0CH—0FH	断点中断	
04H	0:10H—13H	益出中断	
05H	0:14H—17H	PRINT SCREEN	

中断号	向量地址	中断源	备　注
		BOUND 超界☆	BOUND☆
06H	0:18H－1BH	保留	非法操作码☆
07H	0:1CH－1FH	保留	处理扩展非法☆
08H	0:20H－23H	时钟中断	双精度错☆
09H	0:24H－27H	键盘中断	段溢出☆
0AH	0:28H－2BH	保留	非法任务段☆
0BH	0:2CH－2FH	通信口 COM2	段不存在☆
0CH	0:30H－33H	通信口 COM1	堆栈段溢出☆
0DH	0:34H－37H	硬盘	存储保护错☆
0EH	0:38H－3BH	软磁盘	保留
0FH	0:3CH－3FH	打印机	
⋮	⋮	(请参考附录 B)	
1FH	0:7CH－7FH		

☆表示 80X86 定义的内部中断侵占了 8088/8086 的外部中断区域，但 AT 机在 DOS 操作系统下没有使用，仍与 XT 机兼容，只在 XENIX 操作系统状态下使用。由此可见 XENIX 和 DOS 的中断号不一样。

1.1.2 外中断

外中断是由外部设备控制器提出实时中断请求而引起的。这些中断是可屏蔽的，或者由中断控制器设置屏蔽参数禁止指定某些中断，或者直接使用禁止中断指令 CLI(关中断)，以禁止 CPU 响应所有外中断。

图 1.2 是 80X86 硬中断原理图，8086、80386 的中断原理和 80286 基本是相同的，从图中可看出 80286 有 3 种中断信号。

1. NMI：不可屏蔽中断。中断源是硬件故障。例如电压波动、掉电，内存奇偶错等等，一旦发生这些错误，80X86 就无条件地产生 2 号中断(INT 2)。

2. INTR：外部中断。中断源为外部设备，当外部设备输入到 INTR 脚上一个信号，且当控制标识 IF＝1 时，80X86 就开始执行一种称为中断响应周期的读总线周期，从外部的中断控制电路输入一字节的中断类型码而发生中断，如果 IF＝0(既 CLI 指令)时，INTR 引脚上的信号无效。

3. ERROR：中断源为数值运算处理器 80287 产生的异常中断。

连接到中断控制器的中断请求线，是按系统设置的优先级依次与外设控制器相连的，因此，软件无法将其修改，AT 机上使用两片 8259A 中断控制器芯片能支持 16 级外中断。图 1.3 描述了 XT 机和 AT 机外中断请求线的设置。

图 1.4 是 8259 的级联方式，在 CPU 上应用，

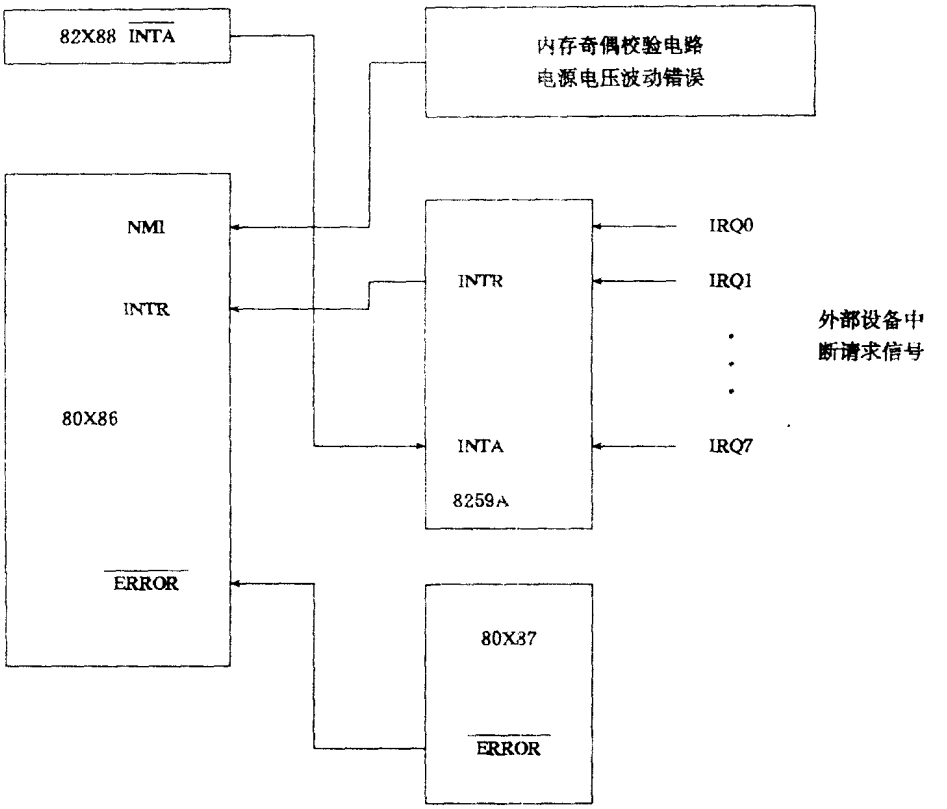


图1.2 硬中断电路原理图

一般地说，中断控制器 8259 在上电初始化期间设置的初始化状态包括：

1. 全嵌套方式。指中断优先级次序为 0—7，在处理某级中断时，屏蔽同级或低级中断，但可响应高一级中断请求。

2. 非自动结束方式。中断处理完之后要向 8259 发中断结束命令 EOI 。

3. 中断请求为边沿触发方式，当这类外部中断请求产生之后，处理机一般要经过中断判优、中断响应、中断处理及中断返回等过程。

80X86 的 8 级外中断向量一览表

中断号	中断源
INT 08H	定时器
INT 09H	键　盘
INT 0AH	保　留
INT 0BH	串行口 2
INT 0CH	串行口 1
INT 0DH	硬　盘
INT 0EH	软　盘
INT 0FH	打印机并行口

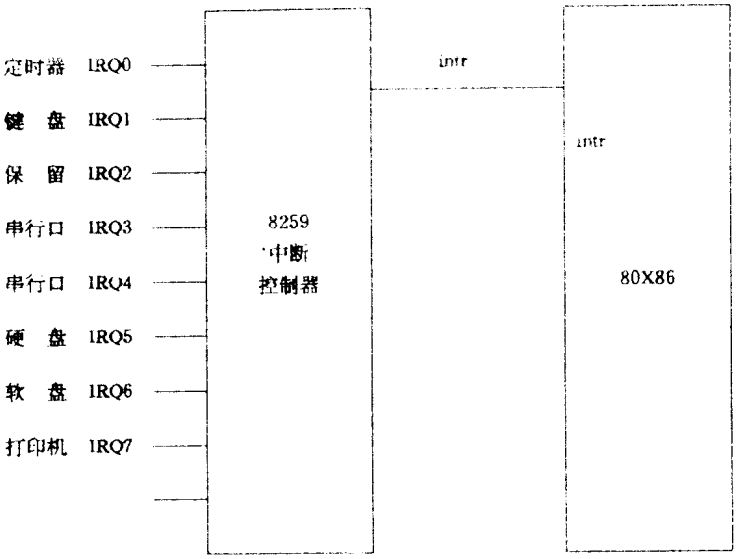


图1.3 80X86中断请求设置

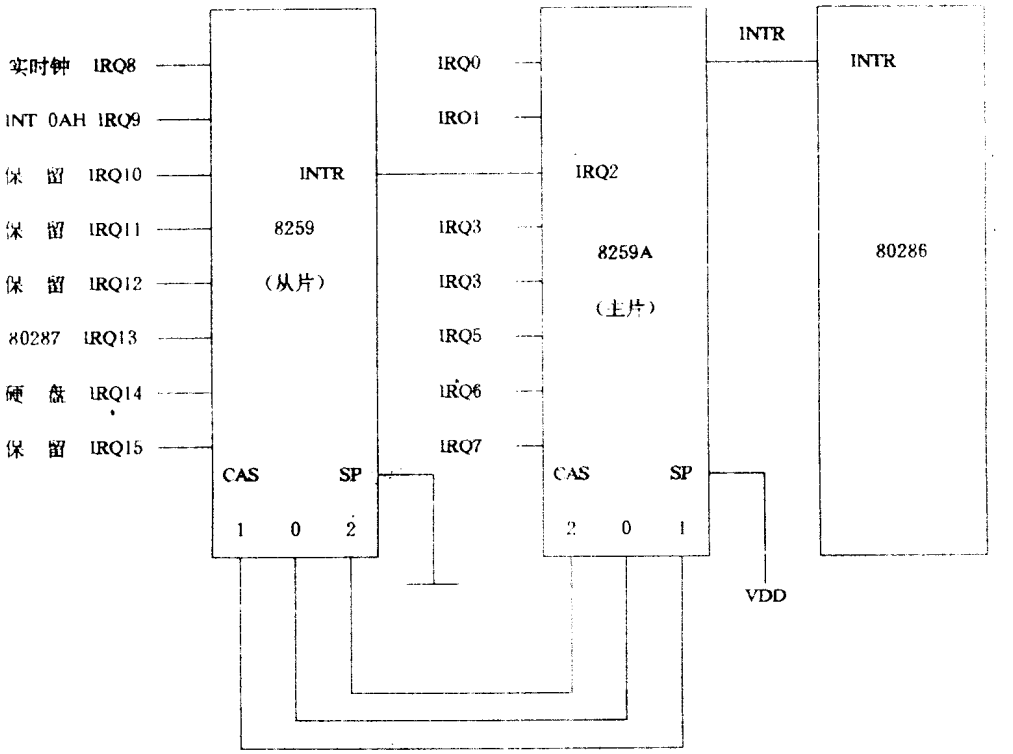


图1.4 80X86中断请求设置(级联方式)

1.1.3 软中断

软中断是指在程序中执行一条 INT 指令(或称软件自陷中断指令)后进入中断例程。

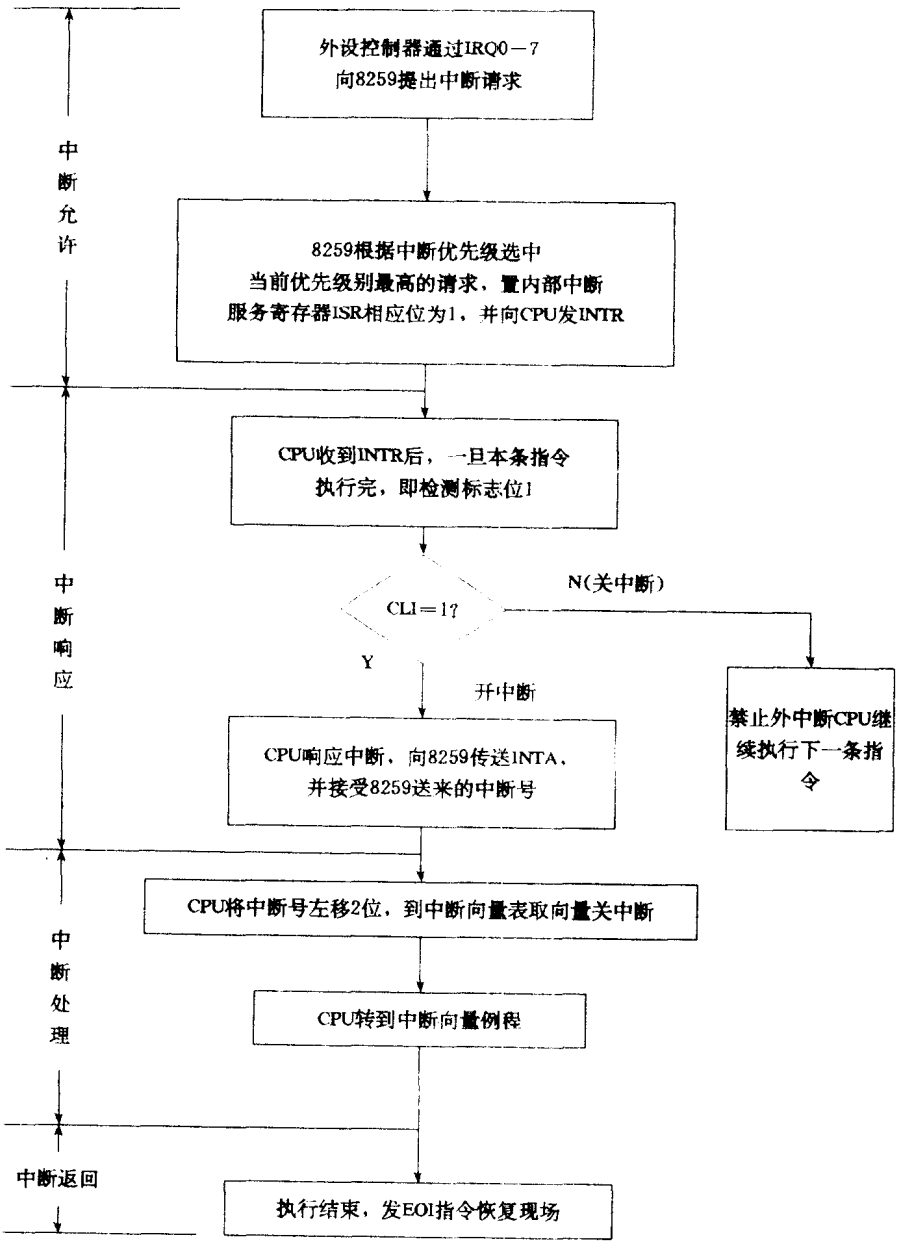


图1.5　外中断处理流程

中断指令形式是：

INT n

此指令为两字节指令，前一个字节是指令的操作码(0CDH)，后一个字节是中断类型码或称中断号。此类中断不受中断允许标志的影响。CPU 在接受 INT N 指令时，立即根据类型码 n 在中断向量表中找到该中断的入口地址，然后自动进入软件中断服务例程。

80X86 机系统中使用的软中断大至分 3 个部分：