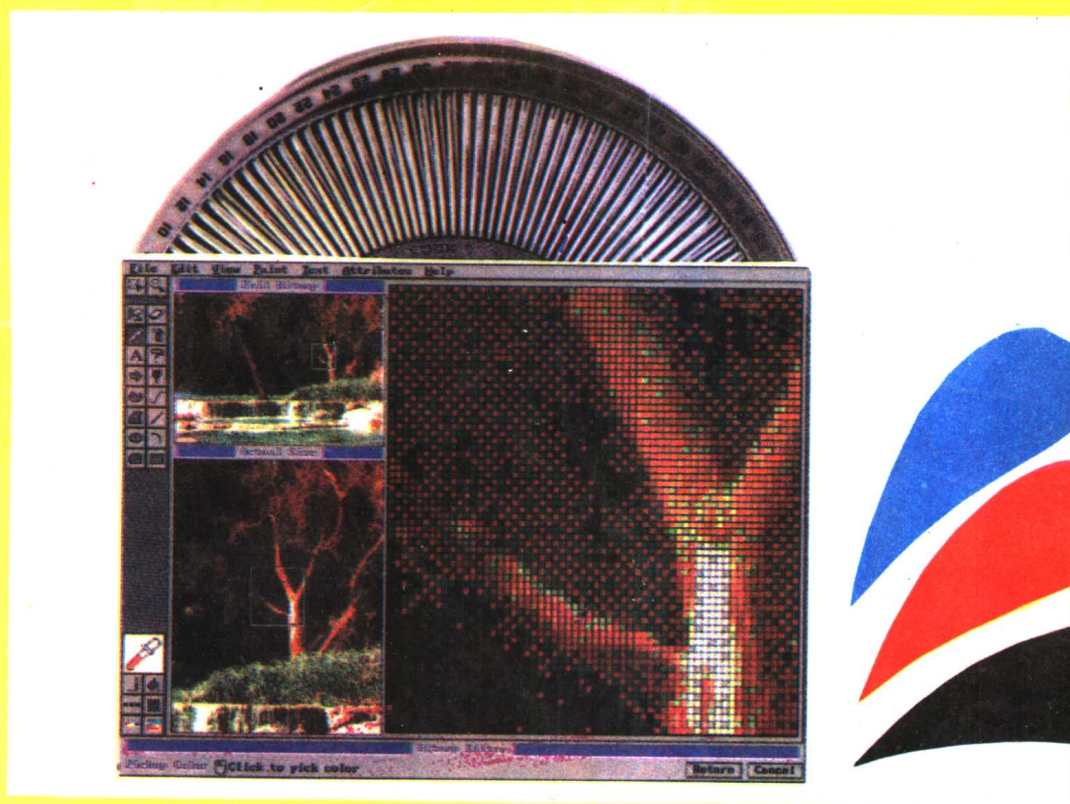


计算机知识普及系列丛书

微机编程 参考手册

王力 编写



学苑出版社

计算机知识普及系列丛书

微机编程参考手册

王 力 编写
肖 雨 审校

学苑出版社

1993.

(京)新登字 151 号

内 容 提 要

本书包括了 IBM PC/XT/AT 及兼容机的三个基本编程服务: DOS 中断、DOS 功能调用和 BIOS 服务程序。除了对微型计算机的编程服务进行详细的说明外, 书中还提供了大量的实例程序, 这些实例程序是以汇编语言、Pascal 语言和 C 语言编写的, 它们都使用了 DOS 和 BIOS 的功能调用。仔细研读这些实例程序, 会更加具体地理解对编程服务的理解并掌握一些实用的编程技巧。

欲购本书的用户, 请直接与北京 8721 信箱联系, 电话 2562329, 邮码 100080。

微机知识普及系列丛书

微机编程参考手册

编 写: 王 力
审 校: 肖 雨
责任编辑: 甄 虎 亮
出 行: 学苑出版社 邮政编码: 100032
社 址: 北京市西城区成方街 33 号
印 刷: 双青印刷厂
开 本: 787×1092 1/16
印 张: 31.25 字 数: 666 千字
印 数: 1~3000 册
版 次: 1993 年 12 月北京第 1 版第 1 次
ISBN7-5077-0821-7/TP·19
本册定价: 19.00 元

学苑版图书印、装错误可随时退换

前 言

目前，微型计算机已在我国得到了广泛的普及。但多数程序员觉得，有关微型机编程服务的内容太多，很难全面地记住它们。因此，他们觉得应当有一本较为全面的参考资料，对有关微型机的编程服务内容有一个完整的介绍。本书就是基于这种目的而编写的。书中包括了 IBM PC/XT/AT 及兼容机的三个基本编程服务：DOS 中断、DOS 功能调用和 BIOS 服务程序。书中的内容不仅连贯一致，而且还可以供读者进行方便的查阅。

除了对微型计算机的编程服务进行详细的说明外，书中还提供了大量的实例程序，这些实例程序是以汇编语言、Pascal 语言和 C 语言编写的，它们都使用了 DOS 和 BIOS 的功能调用。仔细研读这些实例程序，会更加具体地理解对编程服务的理解并掌握一些实用的编程技巧。

本书是集体智慧的结晶，由作者们根据国外的最新技术资料结合自己的实践经验编译而成。由于各方面的因素，难免会有错误之处，衷心希望同志们给予真诚的指导。最后，感谢中科院希望高级电脑技术公司对于本书问世所给予热情支持。感谢所有曾经帮助过我们的同志。

目 录

第一章	引言	1
§ 1.1	关于本书	1
§ 1.2	如何使用本书	1
第二章	BIOS 和 DOS 调用	3
§ 2.1	引言	3
§ 2.2	直接访问硬件	3
§ 2.3	使用 BIOS	4
§ 2.4	使用 DOS	4
§ 2.5	使用高级语言提供的语句	5
§ 2.6	应该使用哪种方法	6
§ 2.7	调用 BIOS 和 DOS	6
§ 2.8	用户程序如何调用 BIOS 和 DOS 服务程序	12
第三章	DOS 中断	13
§ 3.1	引言	13
§ 3.2	如何使用本章	13
§ 3.3	程序段前缀 (PSP)	1
§ 3.4	调用 DOS 中断	17
§ 3.5	DOS 中断目录表	18
第四章	DOS 功能调用	32
§ 4.1	如何使用本章	32
§ 4.2	文件管理: DOS1.X 以上各版本的比较	32
§ 4.3	DOS1.X 文件管理方法	33
§ 4.4	文件控制块格式	34
§ 4.5	磁盘传送地址	37
§ 4.6	DOS2.X 文件管理方法	38
§ 4.7	标准错误代码	39
§ 4.8	使用功能调用	42
§ 4.9	汇编语言使用功能调用	43
§ 4.10	Microsoft C 使用功能调用	43
§ 4.11	Turbo Pascal 使用功能调用	44
§ 4.12	考察和运行程序实例	45
§ 4.13	DOS 功能调用目录	46

第五章	BIOS 中断和功能调用	369
§ 5.1	如何使用本章	369
§ 5.2	BIOS 数据区	369
§ 5.3	调用 BIOS 中断	376
§ 5.4	BIOS 功能调用目录	378
附录	编程服务参照表	490

第一章 引言

§ 1.1 关于本书

对一个程序员而言，最令人扫兴的事情，莫过于当其正在聚精会神地编写一段复杂程序时，忽然记不起一个必须使用的系统功能调用的规则和用法了。IBM PC 机的用户遇到此类问题时，必须仔细查阅 IBM PC 机的有关手册，有时需要连续查找多本手册和资料，甚至需要反复考察晦涩难懂的汇编语言源程序，才能获得有关 IBM PC、XT、AT 及兼容机系统功能的基本信息。即便如此，他们可能仍然需要经过反复试验并对结果进行反复比较，才能掌握和运用一个特定的系统功能调用。

本书正是为这些程序员而编写的，书中包括 IBM PC 机的三个基本编程服务：DOS 中断、DOS 功能调用和 BIOS 服务程序。书中对这些系统服务程序进行了完整的描述并使这些描述不仅具有一致性而且还可以迅速查找到有关的内容。

此外，对于愿意阅读程序而不愿阅读文字描述的读者，书中提供了 200 多个使用 DOS 和 BIOS 调用的程序实例。这些实例程序是使用汇编语言、Microsoft C 语言和 Turbo Pascal 语言这三种最流行的语言编写的。这些实例每一个都是独立和完整的程序，旨在说明如何准确使用 DOS 和 BIOS 所提供的服务功能。

编写本书时，我们设想一个程序员正在编写一个大型复杂的程序，他最需要的并不是程序编制的基本原则和一些不重要的细节，而是能够迅速查找到他需要了解的信息。

因此，本书在编排上力求使读者不必翻阅无关的章节就可以迅速找到所需的内容。我们希望本书能成为 IBM PC 机编程人员的一本快速查询手册。

§ 1.2 如何使用本书

正如读者不会逐页阅读一本字典一样，他们或许不会从本书的首页开始一直阅读到最后（除非读者认为必要）。本书编写的目的是为读者提供一本编程参考手册。正确的方法应该是仅仅翻阅当前感兴趣的部分，或者把书放在方便的地方，以便以后遇到不清楚的问题时可以迅速查找相关的内容。尽管本书篇幅较多，但仅包含五个章节和一个附录，具体编排结构如下：

第一章 引言。这就是读者目前正在阅读的这一章。该章简介全书内容并说明如何使用本书。

第二章 BIOS 和 DOS 调用：它们的功能和如何调用它们。对于以前编程时几乎从未引用过 BIOS 和 DOS 服务程序的人员来说，第二章是一个入门简介。本章讨论了 IBM PC 计算机不同的编程方法，给出中断向量表的概貌，说明了调用 DOS 和 BIOS 服务程序的一般方法。

第三章 DOS 中断调用。这一章详细讨论了所有的 DOS 中断（除中断 21H 将在第四章讨论外）。引言部分还讨论了程序段前缀（PSP）的有关问题。

第四章 DOS 功能调用。这一章详细讨论了 DOS 功能调用，DOS 提供的这个编程服务是通过中断 21H 调用的。引言部分介绍了 DOS2.0 和 DOS1.X 文件管理技术的差异，还讨论了文件控制块 (FCB) 和标准错误代码。

第五章 BIOS 中断和功能调用。这一章详细讨论了 BIOS 所提供的编程服务，这些服务由驻留在 IBM PC 机的只读存储器 (ROM) 中的程序所提供。此外，这章的引言部分还讨论了 BIOS 数据区。

附录 编程服务参照表。这个附录列出本书提供的全部编程服务的参照表。内容是按功能而不是按功能调用号排列的。所以，读者如果完成某种功能 (如显示器 I/O 调用)，但又无法肯定 DOS 或 BIOS 是否提供这种功能时，可查阅这个附录。

因为本书是一本参考手册，所以没有必要从第一页阅读。如果仅仅准备了解 BIOS 调用而不关心 DOS 提供的功能，可以略过前四章，直接阅读第五章。本书内容编排的特点之一是：在讨论某个功能调用的那一节中，将包含有关此调用的全部信息。

为了达到这个目的，有些章节在内容上可能有些重复和交叉，某些问题可能在不同的地方均进行了讨论，这样可避免为了解决一个问题而要查找许多章节，当需要快速查找某些信息时，在一处便可查到所需的全部信息。因为这个特点，如果偶然逐页阅读本书时，读者也许会感到枯燥乏味。

本书的核心章节 (第三章至第五章) 在内容编排上，还注意为读者查找任何有关功能调用及中断的信息提供方便。

首先，功能调用或中断的名字及其功能调用号总是安排在顶角处。这样，如需查找某个功能调用及中断，只要翻看书的顶角处便可找到。编排次序是按中断号的大小顺序排列的，如果一个中断可以提供多种服务，这些服务按功能调用号大小顺序排列。

此外，每个功能调用和中断的说明都采用统一的标准格式，有概要说明、输入参数、输出参数三个部分。输入参数和输出参数的说明简单清晰，一目了然。如果有其它要求或出错情况，也都安排在标准位置处。一旦了解了一个中断或功能调用的内容是如何安排的，就会确切知道应该到哪儿查阅所需的任何中断或功能调用的相应内容。

第二章 BIOS 和 DOS 调用 ——作用和访问方式

§ 2.1 引言

BIOS 和 DOS 是两组系统服务软件的集合，它们使程序能够访问和使用构成 IBM PC 机的硬件。其中包括从键盘读取字符、在显示器上显示信息、读写磁盘、主机向打印机传递信息和许多其它服务。在未详细阐述 BIOS 和 DOS 可以提供什么服务之前先看一看为什么要使用它们以及如何使用它们。

一般来说，程序可以通过四种方式控制 PC 机的硬件（如图 2-1 所示），即直接访问硬件，使用 BIOS 提供的功能，使用 DOS 提供的功能和使用高级语言提供的功能。这四种方式在编程的复杂性和程序的可移植性方面各有利弊。

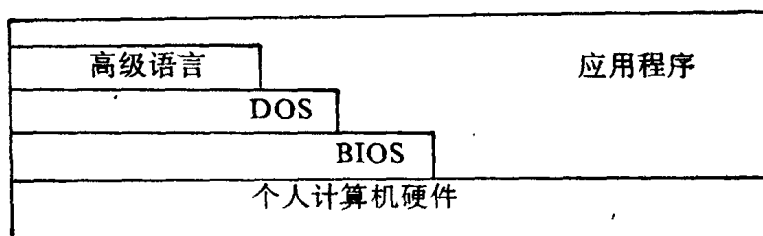


图 2-1 程序控制 PC 机硬件的四种方式

§ 2.2 直接访问硬件

使用 IBM PC 机的最基本方式是利用硬件本身的能力。例如，微处理器（8088 或 80286）通过 I/O 端口与磁盘驱动器、键盘和打印机等其它设备通讯就是如此。I/O 端口是 8 位通道，通过它可以发送和接收数据。每种设备都可以被分配一个或多个 I/O 端口地址，这样每个设备都可单独和微机处理器通讯。一个程序可以通过执行 OUT 指令从 I/O 端口向外发送信息，也可以通过执行 IN 指令从 I/O 端口接收外部信息。

一个使用 IN 和 OUT 指令的程序肯定可以充分利用设备的能力。程序可以通过外设的 I/O 端口向外发送指令，使外设完成指定的操作。然而，这种程序的编写相当繁杂，程序员必须对设备非常熟悉。例如，当访问软盘驱动器时，需要接通过驱动电机、移动磁头到指定位置，对一个扇区进行读写等操作，每个操作至少需要一条复杂的设备驱动指令。

一般来说，如果不是为了得到更高的执行效率和获得 DOS 和 BIOS 不支持的功能，程序不应直接和硬件打交道，直接访问硬件的程序可移植性相当差。在一个厂商生产的机器上调试的程序，在其它厂商生产的兼容机上可能根本无法运行，甚至在同一厂家生产的不同型号的机器上也无法运行。例如在早期的 PC 机上开发的程序可能在 AT 机上根本无

法运行。

§ 2.3 使用 BIOS

BIOS 是一组低级软件程序，这组程序是计算机硬件和其它程序之间的一个缓冲。BIOS 程序本身直接和外设通讯，它为编程人员提供一个简单的接口，避免了和外设打交道而必须编制复杂程序。当计算机加电时，BIOS 服务程序对计算机进行测试以保证硬件工作正常，同时将 DOS 系统从盘上装入内存。

由于 BIOS 在程序和硬件之间形成缓冲，IBM 公司可以引进先进的硬件技术而保持用户软件兼容。这样，即使 PC 机和 AT 机端口地址和外设控制指令不同，一个在 PC 机上运行的程序同样可以在 AT 机上运行。为了在硬件不同时保持软件的兼容性，BIOS 程序必须做相应改动，以保证编程人员看到的接口完全一样。这样，用户程序就不必做任何改动了。

使用 BIOS 而不直接访问硬件将使程序更加简炼。如果用户程序包括大量控制外设的代码，程序所占空间可能超过机器的接收能力。但是，无论是否使用 BIOS，它总是驻留在机器中，仅用几条指令就可以调用用户需要的 BIOS 程序。

BIOS 驻留在系统板上的只读存储器 (ROM) 中，计算机加电后，可以随时调用 BIOS 程序。因为 BIOS 程序提供基本的低层服务，所以调用 BIOS 而不是相应的 DOS 程序可以提高程序的执行效率。

IBM PC 机的 BIOS 已经获得软件版权，这意味着除非得到 IBM 公司的允许，其它厂商在制造自己的计算机时完全复制 IBM 公司的 BIOS 程序代码将是非法的。这样，即使生产兼容机的厂商标榜自己的机器与 IBM PC 兼容，一个有调用 BIOS 语句的程序在兼容机上仍有可能无法运行。

尽管存在在兼容机上无法运行的风险，许多商业上取得成功的软件仍然采用 BIOS 功能调用。这主要是为了得到较高的运行效率，或者需要利用 DOS 不具务的某些功能。这些软件最常用的是 BIOS 提供的显示器控制程序，这些 BIOS 程序允许用户程序改变显示器的显示方式、设置光标和使用其它的 DOS 无法提供的功能。

当程序员编程时，他必须权衡为了使用 BIOS 而获得的较高运行效率和能力，从而冒与其它计算机或运行环境（如 Top View 或 Windows）不兼容的风险是否值得。对用户来说，除非运行效率或使用 DOS 无法提供的功能是至关重要的，否则就应该使用 DOS 调用而不是 BIOS 调用。

§ 2.4 使用 DOS

DOS 是 PC 机上应用最广泛的操作系统。除了能执行从键盘上输入的合法命令外，DOS 还为编程人员提供了丰富的服务程序。

DOS 在更高的层次上提供了与 BIOS 同样的功能。例如，既可以通过调用 BIOS，也可以通过调用 DOS 实现磁盘读写。调用 BIOS 时，必须准确说明读写位置（磁头、磁道和扇区号），才能正确读写信息。但是，在 DOS 系统中已经有文件、目录、子目录等概

念，因此调用 DOS 程序时，不必指出读写信息在盘上的物理位置，只需打开文件，在文件内找到指定位置，就可以进行读写操作。

一般来说，使用 DOS 提供的功能比使用 BIOS 提供的功能更容易。因为使用 DOS 提供的功能时，编程人员不需要对硬件有更多的了解。通过调用 DOS，还可以充分利用操作系统提供的所有功能，如分级文件系统、装载和执行程序、分配内存等。但是，并非 BIOS 的所有功能都能通过调用 DOS 来完成。由于在编写 DOS 时，对可能调用它的情况做了某些假设，因此，有些 BIOS 所提供的功能无法用 DOS 实现。

大多数情况下，并不需要利用 BIOS 调用直接访问硬件，因为用户希望外设工作的方式恰恰是 DOS 调用所提供的方式。磁盘的 I/O 操作就是一个很好的例子。大多数程序进行 I/O 操作时，都希望以文件方式读写数据。如果不使用 DOS 而使用 BIOS 调作进行文件读写，既复杂又容易出错。虽然可以将信息写在指定的位置，但往磁盘上写之前，必须对盘区格式信息非常清楚，才能既保证不覆盖已存在的文件，同时也防止刚刚写入的文件以后被别的程序覆盖。使用 DOS 时，这些工作都由 DOS 系统自动完成。

调用 DOS 功能除接口简单外，程序的可移植性也比使用 BIOS 好，仅仅使用 DOS 功能的程序可以在任何支持 PC-DOS 或 MS-DOS 的计算机上运行。因为许多兼容机厂商生产的 PC 机和 IBM PC 并不是百分之百兼容。如果开发的程序准备作为商品出售，可移植性将十分重要。

但是 DOS 程序的执行效率比 BIOS 低，提供的功能也没有 BIOS 丰富，显示器的 I/O 操作就是一例。DOS 仅提供简单的字符和字符串输出能力，如果使用图形方式、设置字符的属性和颜色，或者需要提高输出显示的速度，就必须使用 BIOS 或者直接访问硬件。

在可能的情况下，尽量使用 DOS 而不是 BIOS，这样程序将具有良好的可移植性。由于 DOS 对复杂操作提供了一个简单接口，程序既容易编写，又方便调试。

§ 2.5 使用高级语言提供的语句

如果用 Pascal 或 C 等高级语言编程，则可以充分利用嵌入在高级语言中的许多操作系统的功能。例如，可以直接使用高级语言提供的语句从键盘接收信息，向显示器写数据和读写磁盘文件。如果高级语言提供的功能完全能满足编程的需要，就没有必要直接使用 DOS 和 BIOS 功能调用。这样程序将具有良好的可移植性（只要把源程序在有编译程序的机器上重新编译一次即可），而且接口比较简单。

但是高级语言提供的 I/O 功能比 DOS 要少，所以有些操作仅仅使用高级语言提供的语句将无法完成。此外，使用高级语言做 I/O 操作将明显增加程序代码长度，例如，即使一个仅使用 REAKLN 和 WRITELN 语句从键盘读入和往显示器写出信息的简单 Pascal 程序，经过编译和连接后产生的程序代码也比调用 DOS 功能从键盘输入和显示器 I/O 操作的程序大 20KB 左右。

§ 2.6 应该使用哪种方法

泛泛地谈论应该使用哪种方法是没有意义的。应该根据编程人员的素质和所编程序的使用要求来考虑这个问题，一个程序员需权衡程序的可移植性、编程的复杂性和目标代码的大小后做出决定。

如果决定使用 BIOS 或 DOS 编程，本书将是一个十分有用的工具。书中详细讨论了 DOS 和 BIOS 功能调用的有关问题，并且对每个 DOS 功能调用都给出了三种语言编写的程序实例。本书的绝大部分内容与 IBM 和 Microsoft 的技术手册完全一致，但是，我们把分散在各种手册中的信息综合在一起，并且补充了实际使用中的大量实例和体会。在内容安排上尽量做到信息编排合理，以便读者能迅速、方便地查找有关信息。

利用高级语言提供的语句编程不在本书讨论的范围内，应该查阅有关高级语言的编译参考手册的书籍。但是在实际编程中，如果读者发现某些功能仅通过高级语言无法实现时，可以浏览本书，看看 DOS 和 BIOS 提供的功能调用是否有助于解决这些问题，最好按先 DOS 后 BIOS 的顺序查阅。

如果 DOS 和 BIOS 提供的功能调用还无法满足编程需要，就只好直接访问硬件。这种情况也不属于本书讨论的范围，应查阅《IBM 技术参考手册》。如果需要更详细的资料，可能还要查阅有关的硬件参考手册。但是在决定直接访问硬件之前，应先从 DOS 功能调用开始，认真查阅本书，也许会找到解决问题的办法。

§ 2.7 调用 BIOS 和 DOS

如果一个程序是由主程序和若干子程序组成，主程序可以通过调用语句调用子程序。但是，如果子程序是分模块编译的，则要把包含各个子程序的目标文件和主程序连接在一起，最后形成一个包含主程序和所有子程序的可执行文件。

BIOS 和 DOS 服务程序的作用与子程序类似，但实现调用时不能用它们的名字，主程序也不必与 BIOS 或 DOS 程序代码连接（见图 2-2），所以形成的代码比较紧凑。另外，调用它们的程序并不需要知道 BIOS 和 DOS 代码实际存储空间的地址。

编程时，调用 BIOS 和 DOS 程序的方法是采用处理器中断。中断是停止 CPU 目前正在处理的工作，而非运行其它程序的信号。中断可以由微处理器本身产生（例如，当试图用 0 作除数时），或者由硬件产生（键盘、磁盘驱动器、系统时钟都可产生中断），也可以由软件产生（当程序执行一条 INT 指令时，就会产生一个中断）。

无论什么时候，产生中断时处理器将做下列工作：

1. 保护现场：CPU 将段寄存器（CS）、指令指示器（IP）和标志寄存器压入堆栈保护起来，以便中断处理程序运行结束后恢复中断前的现场，保证程序继续正确执行。

2. 关中断：如果中断源是硬件，CPU 将清除中断标志位（IF），这时，CPU 在未结束当前的中断服务程序前，对其它的中断请求信号不予响应。中断服务程序完成后，通过重新设置中断标志位，交出中断控制权。

3. 运行中断程序：处理器将控制权交给中断服务程序。该服务程序的地址在中断向量

表中。向量表包含 256 个中断向量，首地址是 RAM 的 0H 字节。开机时，CPU 负责填写这些单元的内容（当 80286 处理器工作在保护方式时，情况则不同，包括中断向量表的地址都将发生变化，但 CPU 工作在保护方式的情况不在本书讨论的范围内）。

中断向量表中每项（又称中断向量）由四个字节组成，高两位字节是每个中断服务程序的段地址（CS），低两位字节是段内偏移量（逻辑地址）。其中段地址是 16 位，实际上代表 20 位的地址码，不过最低 4 位的值始终是“0”而已，这个段地址指向一个段的首地址。偏移量表示相对于段地址偏移的字节数。由于中断表中共有 256 个中断向量，每个中断向量占四个字节，整个中断向量占用 1KB 的空间，从内存的 0H 单元到 3FFH 单元。

产生中断时，根据中断性质和数值选择相应的中断向量。例如硬件中断来自连接 8259 中断控制器的信号线，这些信号线自动映射为中断向量表的特定项。软件中断是通过中断调用号选择向量表的特定项，如果程序中有 INT20H 指令，CPU 将调用地址在中断向量表 20H 处的中断服务程序。

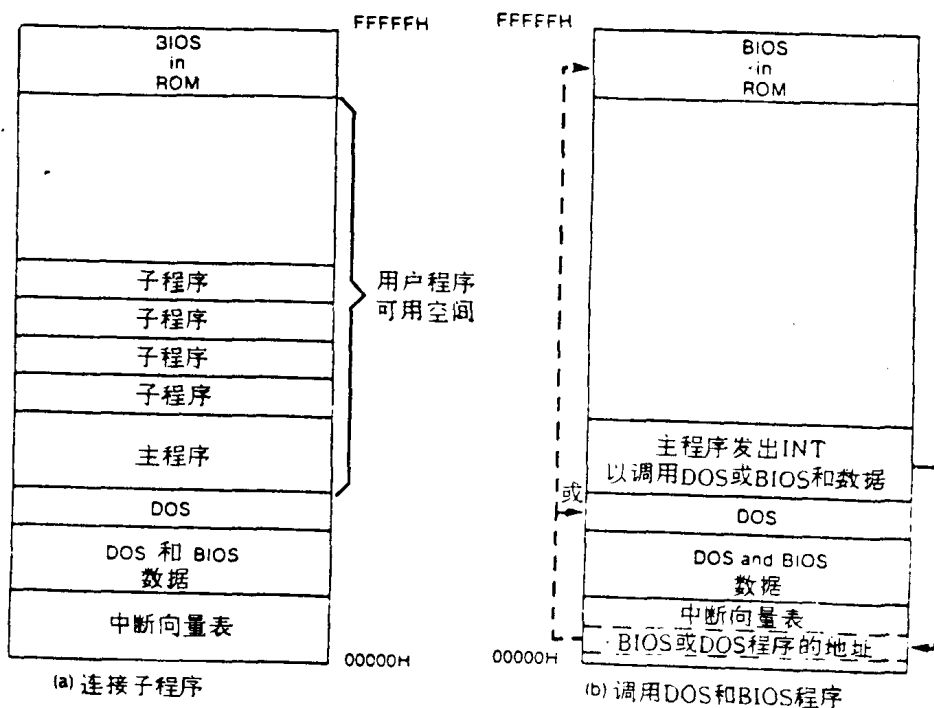


图 2-2 主程序与子程序连接完成某些 OS 服务

图 2-2 通过主程序与特定子程序连接完成某些操作系统的服务将增加程序代码的空间。但是 DOS 和 BIOS 代码已经存储在内存中，用户程序无需把它们再复制到自己的程

序中，所以调用 DOS 或 BIOS 子程序不会增加用户程序空间。

当地址在中断向量表中的程序取得控制权时（这些程序有时称作中断处理程序），通常立即执行相应的操作，或者一旦开始运行即迅速打开中断，以便必要时 CPU 能够响应更高级的中断。

中断服务程序完成以后，执行中断返回指令 IRET，该指令恢复 CS、IP 及标志寄存器，CPU 继续执行原来被中断的程序。

计算机加电时，中断关闭，CPU 执行驻留在 ROM 中的 BIOS 程序。BIOS 程序做的第一件事是填写中断向量表，把 BIOS 中断服务程序的地址写入中断向量表中。这些中断包括处理器中断（如被 0 除）、硬件中断（硬盘、键盘和时钟中断）以及 BIOS 本身的特定软件中断（包括本书所讲座的 BIOS 功能调用）。一旦这些中断发生，便立即转入相应的中断处理程序。

BIOS 仅仅负责填写中断向量表的部分单元。当 BIOS 运行完启动程序（开机、自测程序等）后，将 DOS 程序装入内存。DOS 程序做的第一件事是在中断向量表的剩余单元中填入 DOS 中断处理程序的入口地址。这样，填写中断向量表的工作就完成了。但仍保留一些未用的中断向量（从 78H 到 7FH），留给编程人员建立自己的中断处理程序（Interrupt handler——又称中断句柄）。

BIOS 提供的某些中断的服务程序是计算机运行的最基本软件，如果没有这些软件，计算机根本无法运行。但是，绝大多数的 BIOS 和 DOS 中断向量是指向 BIOS 和 DOS 常用软件服务程序的入口地址。通过中断向量表，就可以调用这些服务程序，而不必知道它们的实际驻留地址，也不必把它们与用户程序连接，这样就减少了用户程序的代码长度。调用中断服务程序时，首先在某些寄存器中填写相应数值，然后执行一条 INT n 指令，其中 n 是中断号。当调用的中断处理程序（BIOS 或 DOS 功能调用）运行结束后，说明运行情况的结果存放在某些寄存器中，并交回控制权。

表 2.1 列出了 BIOS 和 DOS 建立的中断向量，这些中断的绝大部分将在以下三章中详细讨论。许多中断调用具有多重功能（例如，全部 DOS 功能调用都是通过中断 21H 实现的）。

绝对地址	中断向量		说 明
	16 进制	10 进制	
00H	0H	0	0 做除数时处理器发出的中断
04H	1H	1	单步调试时处理器发出的中断
08H	2H	2	非屏蔽中断
0CH	3H	3	调试程序设置断点时处理器发出的中断
10H	4H	4	发生算术溢出时处理器发出的中断
14H	5H	5	调用 BIOS 的屏幕拷贝操作
18-1FH	6-7H	6-7	保留单元
20H	8H	8	每 1/18.2 秒定时器发出的中断
24H	9H	9	按压或释放键盘时产生的中断
28H	0AH	10	保留单元

绝对地址	中断向量		说 明
	16 进制	10 进制	
2CH	0BH	11	通讯设备使用的硬件中断
30H	0CH	12	通讯设备使用的硬件中断
34H	0DH	13	交替打印时硬件产生的中断
38H	0EH	14	软盘驱动器操作结束时产生的硬件中断
3CH	0FH	15	打印机发出警告信号时产生的硬件中断
40H	10H	16	BIOS 的显示 I/O 功能调用
44H	11H	17	BIOS 设备确认调用
48H	12H	18	BIOS 确定内存空间大小的功能调用
4CH	13CH	19	BIOS 的磁盘 I/O 功能调用
50H	14H	20	BIOS 的 RS-232 串行 I/O 功能调用
54H	15H	21	在 PC 和 XT 机上是 BIOS 磁带 I/O 功能调用。在 AT 机上，是 AT 扩充服务功能调用。
58H	16H	22	BIOS 的键盘 I/O 功能调用
5CH	17H	23	BIOS 的打印机 I/O 功能调用
60H	18H	24	ROM 的 BASIC 解释程序功能调用
64H	19H	25	BIOS 的装载引导服务功能调用
68H	1AH	26	BIOS 的日期时钟功能调用
6CH	1BH	27	Ctrl-Break 处理程序功能调用当键入 Ctrl-Break 键时指向可执行的程序入口，初始化时 BIOS 使此向量指向一条 IRET 指令。用户可修改此向量指向自己的程序。
70H	1CH	28	指向每 1/18.2 秒时可执行的服务程序的入口，初始化时此向量指向一条 IRET 指令。用户可修改此向量，使其指向自己的 Ctrl-Break 处理程序。
74H	1DH	29	指向显示控制器初始化参数表。BIOS 使这个向量指向 ROM 驻留表。

绝对地址	中断向量		说 明
	16 进制	10 进制	
78H	1EH	30	指向软盘参数表。BIOS 使这个向量指向 ROM 驻留表，但是 DOS 把它改为指向 DOS 的 RAM 驻留表。
7CH	1FH	31	指向一点阵表。在这个表中，BIOS 可以找到字符集后 128 个字符的点阵(前 128 个字符的点阵在 ROM BIOS 中)。仅在图形方式下才使用这此点阵。在字符方式下，显示适配器的字符发生器产生所有 256 个字符，BIOS 初始化这个向量为 0:0。因此，在图形方式下，用户必须建立一个表，并根据显示后 128 个字符来修改这个向量。
80H	20H	32	终止程序的 DOS 功能调用。
84H	21H	33	任何种 DOS 功能调用
88H	22H	34	指向 DOS 的结束地址
8CH	23H	35	指向 DOS 的 Ctrl-Break 处理程序
90H	24H	36	指向 DOS 的严重错误处理程序
94H	25H	37	DOS 绝对磁盘读调用
98H	26H	38	DOS 绝对磁盘写调用
9CH	27H	39	程序终止，但仍驻留内存的 DOS 功能调用
A0-FFH	38-3FH	40-63	为 DOS 保留的单元
100H	40H	64	保留单元
104H	41H	65	指向硬盘 0 参数表，BIOS 使这个向量指向 ROM 驻留表。
108-10FH	42-43H	66-67	保留单元
110H	44H	68	PCjr 机使用，用于指向低分辨率图形字符参数表。
114H	45	69	保留单元

绝对地址	中断向量		说 明
	16 进制	10 进制	
118H	46H	70	指向硬盘 1 的参数表, BIOS 使这个向量指向 ROM 驻留表。
11CH	47H	71	保留单元
120H	48H	72	PCjr 机使用, 用于把 PCjr 的键盘代码变换为标准的键盘代码。
124H	49H	73	指向键盘增强服务变换表
128-17FH	4A-5FH	74-95	保留单元
180-19FH	60-67H	96-103	为用户程序保留的单元
1A0-1BFH	68-6FH	104-111	未使用
1G0H	70H	112	硬件中断(IRQ-interrupt request) 8——实时钟中断
1C4H	71H	113	硬件中断 9
1C8H	72H	114	硬件中断 10
1CCH	73H	115	硬件中断 11
1D0H	74H	116	硬件中断 12
1D4H	75H	117	硬件中断 13——BIOS 把这个中断向量重定向为非屏蔽中断(NMI)
1D8H	76H	118	硬件中断 14
1DCH	77H	119	硬件中断 15
1E0-1FFH	78-7FH	120-127	未使用
200-217H	80-85H	128-133	为 BASIC 保留
218-3C3H	86-F0H	134-240	BASIC 程序运行时, 提供给 BASIC 解释程序使用
3C4-3FFH	F1-FFH	241-255	未使用

表 2.1 中断向量表