80386程序员参考手册

章立生　王英斌　郭京虎　孙　义

北京科海集团公司培训中心
北京信通集团技术培训部

编辑：科海培训中心教材部
发行：科海培训中心资料组
地址：北京2725信箱　科海培训中心
资料组
（北京海淀路332路黄庄站旁）
印刷：河北省蔚县印刷厂

目　录

第二部分 系统程序设计

第三部分 兼容性

第四部分 指令系统

第一章　80386引论

80386是一个先进的32位微处理器，它是为多任务操作系统优化而设计的，是为需要非常高的性能的应用而设计的。32位寄存器和数据通路支持32位的地址和数据类型。处理器能编址多达4G字节的物理存贮器和4T（2^{64}）字节的虚拟存贮器。片上的存贮器管理机制包括有地址转换寄存器，先进的多任务硬件，保护机构和分页的虚拟存贮器，即使在基于ROM的软件中，专用的调试寄存器也能提供数据和代码的断点。

1.1　本手册的组织

这本书可分为以下五个部分，它们表达了80386的体系结构：

第一部分　——应用程序设计
第二部分　——系统程序设计
第三部分　——兼容性
第四部分　——指令系统
第五部分　——附录

这几个部分是部分地按体系结构本身和按使用这本书的不同方法来确定的。正如在下面的表中所指示的那样，后两部分打算作为那些确实想加入为80386开发软件工作的程序员的参考资料。前三部分是说明性的，表示体系结构特性的目的，发展术语和思想及描述指令（当指令与特定的目的或特定的体系结构特性相关时）。

解释说明	第一部分　应用程序设计 第二部分　系统程序设计 第三部分　兼容性
参　　考	第四部分　指令系统 附录

前三部分遵守80386 CPU的执行方式和保护特性。应用特性与系统特性之间的区别由80386的保护机构决定。保护的目的之一是防止应用程序影响操作系统，因此，处理器使得某些寄存器和指令对应用程序是不可访问的。在第一部分讨论的特性是那些应用程序可访问的特性，第二部分讨论的特性是那些仅对已经给定特定特权或在非保护系统中的系统软件是可访问的。

80386的处理方式也决定了这种可访问的特性。它有以下三种处理方式：

1．保护方式

2．实地址方式

3．虚拟8086方式

保护方式是80386处理器的自然32位环境，在这种方式中，所有的指令和特性都是可用的。

实地址方式（通常也称为“实方式”）是处理器在复位（RESET）后立即进入的处理方式。在实方式中，对程序员来说80386表现为具有一些新的指令的快速8086。80386的大多数应用程序都仅使用实方式来初始化。

虚拟8086方式（也称为V86方式）是一个动态方式，处理器能够重复和迅速地在V86方式和保护方式之间切换。CPU从保护方式进入V86方式以执行一个8086程序，然后退出V86方式并进入保护方式以继续执行一个宿主80386程序。

那些对于在保护模式中的应用程序和对于所有在V86方式中的程序可用的特性都是相同的。这些特性组成了第一部分的内容。对于在保护方式中的系统软件可用的其它特性组成第二部分。第三部分解释实地址方式和V86方式，以及怎样执行32位和16位程序的混合。

在所有方式中可用	第一部分——应用程序设计
仅在保护方式中可用	第二部分——系统程序设计
兼容方式	第三部分——兼容性

1.1.1 第一部分——应用程序设计

这一部分表述由应用程序员使用的体系结构的几个方面：

第二章——基本程序设计模型：介绍存贮器组织模型；定义数据类型；描述由应用程序使用的寄存器组；介绍堆栈；解释串操作；定义一条指令的各个部分；解释寻址计算；介绍中断和异常，因为它们会用于应用程序设计。

第三章——应用程序指令系统。简述应用程序设计通常使用的指令；按功能相关组理解指令，例如，串操作理解为一种动作，而控制转移指令考虑为另一种动作；解释指令中的概念；每条指令的细节在第四部分（指令系统参考）中介绍。

1.1.2 第二部分——系统程序设计

这一部分描述通常由系统程序员使用的体系结构的那些方面。系统程序员编写操作系统，设备驱动程序，调试程序以及其它支持在80386保护方式中的应用程序的软件。

第四章——系统体系结构：概述由系统程序员使用的80386的特性；介绍在第一部分没有讨论的80386的其它寄存器和数据结构；介绍所支持的寄存器和数据结构环境中的面向系统的指令；指出更加详细地讨论每个寄存器、数据结构和指令的章节。

第五章——存贮器管理：描述数据结构、寄存器、支持虚拟存贮器的指令的细节和分段与分页的概念；解释系统设计者怎样能够在完全线性的（“平面的”）组织和分段和分

页的组织范围内选择存贮器组织的模型。

第六章——保护：扩充80386的存贮器组织的特性以包括保护特性，因为它应用于分段和分页；解释特权规则、堆栈切换，指针有效性、用户和管理员方式的实现。关于多任务系统的保护方面在后面的章节中讨论。

第七章——多任务：解释80386硬件怎样支持多任务系统，这些多任务系统具有环境切换操作和栈间保护的特性。

第八章——输入/输出：揭示80386的I/O特性，包括I/O指令、保护（当它与I/O有关时）以及I/O允许位图。

第九章——异常和中断：解释80386的基本中断机制；表示中断和异常怎样与保护有关；讨论所有可能的异常事件、列出原因，并包括处理和恢复异常事件所必须的信息。

第十章——初始化：定义在RESET或加电后处理器的状态；解释怎样为实地址方式或保护方式设立寄存器、标志和数据结构；给出一个初始化程序实例。

第十一章——协处理和多处理：解释支持数值协处理器和共享存贮器的多CPU的指令和标志。

第十二章——调试：描述怎样使用80386的调试寄存器。

1.1.3 第三部分——兼容性

本书的其它部分主要是以32位机器来看待80386处理器，而为了简单性忽略了16位操作的机制。实际上，80386是一个32位机器，但它的设计也完全支持16位操作数和寻址。通过解释80386支持16位程序和在32位程序中的16位操作的体系结构特性，这一部分提供80386的完全的描述。为了执行16位程序使用了所有三个处理器模型：保护方式能直接运行16位的80286保护方式程序，实方式执行8086程序和实方式80286程序，虚拟8086方式在多任务环境中与其它80386保护方式程序一起执行8086程序。另外，在保护方式中，32位和16位模块以及单个32位和16位操作能混合使用。

第十三章——执行80286保护方式代码：在80386的保护方式中，80386能执行完全的80286保护方式系统，因为80286功能是80386功能的一个子集。

第十四章——80386实地址方式：解释80386 CPU的实方式。在这个方式中，80386好象是具有增强的指令系统的快速实方式80286或快速8086。

第十五章——虚拟8086方式：80386能够在它的保护方式和V86方式之间快速切换，使它具有与32位程序一起执行多个8086程序的能力。

第十六章——混合16位和32位代码：即使在一个程序或一个任务中，80386也能够混合32位模块和16位模块。因此，任何给定的模块都能够使用16位和32位的操作数和地址。

1.1.4 第四部分——指令系统

第一、二、三部分在指令与特定的体系结构方面相关时描述了指令的一般概述，但是，这一部分按字母顺序描述指令，提供汇编语言程序员和调试程序、编译程序和操作系统等的编程者所需要的细节。指令描述包括操作的算法描述，标志设置的影响，对标志设置的影响，操作数尺寸和地址尺寸属性的影响，处理器方式的影响以及可能的异常。

1.1.5 附录

附录是按为了由汇编语言程序员和系统程序员作快速参考的格式描述了译码表和其它细节。

1.2 相关的文献

下列书目含有关于80386微处理器的其它材料：

- 80386引论
- 80386硬件参考手册
- 80386系统软件设计者指南
- 具有集成存贮器管理的80386高性能32位微处理器（数据表）。

1.3 表示约定

本手册为数据结构格式、指令的符号表达和16进制数使用了专门的表示法。对这些表示法的回顾会使得阅读这本手册更容易些。

1.3.1 数据结构格式

在表示存贮器中的数据结构时，图中的右低部分是低地址部分，地址是朝着左上方向增长的。位的位数是从右向左标号。图1-1表示这个约定。

1.3.2 未定义位和软件兼容性

在许多寄存器和存贮器的总体描述中，有些位标记为未定义。当这些位标记为未定义时（如图1-1所示），这对于与未来的处理器的兼容性是重要的，软件将这些位作为未定义处理。软件在处理这些未定义位时应遵循下列要点：

- 当测试那些含有未定义位的寄存器的值时，不要依赖任何未定义位的状态。
- 当在存贮器或在另一寄存器中存放未定义位时，不要依赖于任何未定义位的状态。
- 不要依赖能保持写入未定义位的信息的能力。
- 当装入一个寄存器时，给未定义位总是装入零，或者给未定义位重装入先前从同一寄存器存放的值。

注　　意

依赖于未定义寄存器位的值，使得软件依赖于80386处理这些位的未指定方式。依赖于未定义值会有使软件与未来定义这些位的使用的处理器的不兼容的危险。应避免任何软件对未定义的80386寄存器位的状态的依赖性。

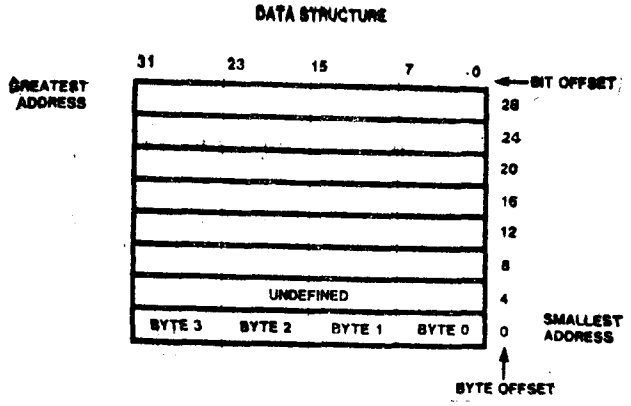

图 1-1　数据结构举例

1.3.3　指令操作数

当指令按符号表示时，使用80386汇编语言的一个子集。在这个子集中，一条指令有下面的格式：

标号：前缀　助记符　参数1，参数2，参数3

这里：

- 标号是一个跟有一个冒号的标识符。
- 前缀是指令前缀的一个可选保留字。
- 助记符是一类具有相同功能的指令操作码的保留字。
- 操作数“参数1”，“参数2”和“参数3”是可选的。可以有0至3个操作数，这要取决于操作码。它们是以数据项的文字或标识符出现的。操作数标识符是保留字，或是寄存器，或者假定为赋予在程序的另一部分定义的数据项（这点不在示例中表示）。当在修改数据的指令中出现两个操作数时，右边的操作数是源操作数，左边的操作数是目的操作数。

例如：

LOADREG: MOV EAX, SUBTOTAL

在这个例子中，LOADREG是一个标号，MOV是一个操作码的助记标识符，EAX是目的操作数，SUBTOTAL是源操作数。

1.3.4　十六进制数

用一个16进制数字串后跟上字母H来表示以16为基的数。一个16进制数字是集合(0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F）中的一个字符。在有些情况下，特别是在程序语法中，如果一个数是从A-F中的一个数字开始，则在该数前要加上一个“0”，例如，OFH与十进制数15相等。

第一部分　应用程序设计

第二章　基本程序设计模型

本章描述80386在保护方式下运行时，由汇编语言程序员所见到的80386应用程序设计环境。本章给程序员介绍那些直接影响应用程序的设计和实现的80386的体系结构的特性。其它章节讨论与系统程序设计相关或同其它处理器（8086处理器系列）相容相关的80386特性。

基本程序设计模型包括以下几个方面：

- 存贮器组织和分段
- 数据类型
- 寄存器
- 指令格式
- 操作数选择
- 中断和异常

注意输入/输出不包括在基本程序设计模型部分。系统设计者可以选择使I/O指令对应用程序是可用的，也可以选择将这些功能为操作系统保留。由于这个原因，80386的I/O特性在第二部分讨论。

本章含有为每个对应用程序可见的体系结构方面的章节。

2.1　存贮器组织和分段

80386系统的物理存贮器组织为一个8位字节序列。每个字节赋予一个从0到最大值为$2^{32}-1$（4G字节）的唯一地址。

但是，80386程序独立于物理地址空间。也就是说，不需了解有多少物理存贮器可用、也不需精确了解指令和数据存放于物理存贮器的位置，就可以编写程序。

应用程序员所见的存贮器组织模型是由系统软件设计者所确定的。80386的体系结构给设计者为每个任务选择一个模型的自由。存贮器组织的模型介于下面两种极端情况之间：

- 一个“平面”地址空间，含有最大为4G字节的单维阵列。
- 一个分段地址空间，含有一组最多为16,383个线性地址空间，每个线性地址空间最多为4G字节。

两种模型都能提供存贮器保护。不同的任务可以使用不同的存贮器组织模型。设计者用来确定存贮器组织模型的准则和系统程序员用来实现该模型的方法在第二部分（系统程序设计）中介绍。

2.1.1　“平面”模型

在一个存贮器组织的“平面”模型中，应用程序员所见的是一个最大为2^{32}字节（4G