Chapter 8 Input/Output 1

8.1 I/O Addressing 1

8.1.1 I/O Address Space 1

8.1.2 Memory-Mapped I/O 2

8.2 I/O Instructions 3

8.2.1 Register I/O Instructions 3

8.2.2 Block I/O Instructions 4

8.3 Protection and I/O 5

8.3.1 I/O Privilege Level 5

8.3.2 I/O Permission Bit Map 6

# Chapter 8 Input/Output

本章从以下角度介绍了80386的I/O功能：

* I/O端口的寻址方法
* 导致I/O操作的指令
* 保护适用于I/O指令和I/O端口地址的使用。

# 8.1 I/O Addressing

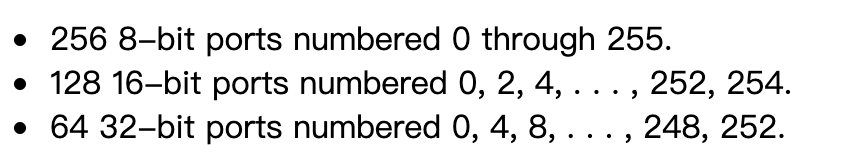
80386允许以两种方式之一执行输入/输出：

* 通过单独的I / O地址空间（使用特定的I / O指令）
* 通过内存映射的I / O（MMIO）（使用通用操作数操作指令）。

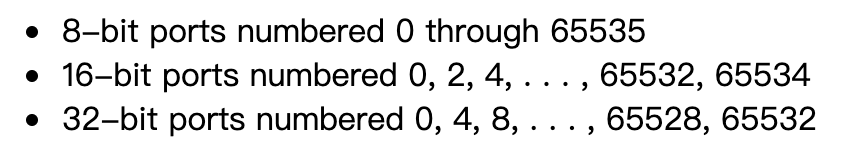
## 8.1.1 I/O Address Space

80386提供了一个与物理内存不同的独立I / O地址空间，可用于寻址用于外部16个设备的输入/输出端口。 I / O地址空间由2 ^（16）（64K）个可单独寻址的8位端口组成； 任何两个连续的8位端口都可以视为16位端口； 四个连续的8位端口可以视为32位端口。 因此，I / O地址空间最多可容纳64K 8位端口，32K 16位端口或16K 32位端口。

该程序可以通过两种方式指定端口的地址。 使用立即数字节常量，程序可以指定：



使用DX中的值，程序可以指定：



80386可以一次将32位，16位或8位传输到位于I / O空间中的设备。 像内存中的双字一样，应将32位端口对齐到可被4整除的地址，以便可以在单次总线访问中传输32位。 像内存中的字一样，16位端口应对齐在偶数地址处，以便可以在单个总线访问中传输这16位。 8位端口可以位于偶数或奇数地址。

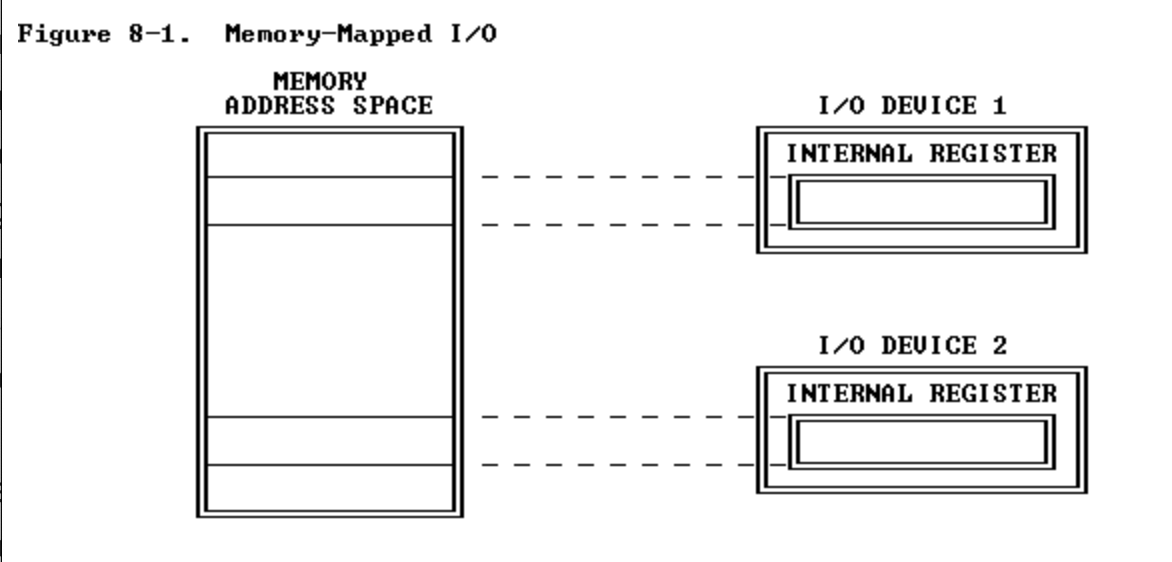
指令IN和OUT在I / O地址空间中的寄存器和端口之间移动数据。 INS和OUTS指令在存储器地址空间和I / O地址空间中的端口之间移动数据串。

## 8.1.2 Memory-Mapped I/O

I / O设备也可以放置在80386内存地址空间中。 只要设备像内存组件一样做出响应，它们就不会与处理器区分开。

内存映射的I / O提供了额外的编程灵活性。 任何引用内存的指令均可用于访问位于内存空间中的I / O端口。 例如，MOV指令可以在任何寄存器和端口之间传输数据。 以及AND，OR和TEST指令可用于操作设备内部寄存器中的位（见图8-1）。 通过完整指令集执行的内存映射I / O保留了用于选择所需I / O设备的寻址模式的完整补充（例如，直接地址，间接地址，基址寄存器，索引寄存器，缩放比例）。

像任何其他内存引用一样，内存映射的I / O在保护模式下执行时也要进行访问保护和控制。 有关内存保护的讨论，请参见第6章。



# 8.2 I/O Instructions

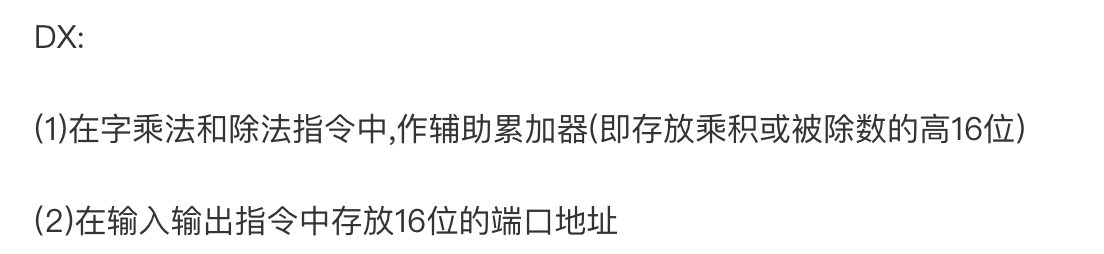
80386的I / O指令提供对处理器I / O端口的访问，以用于与外围设备之间的数据传输。 这些指令将I / O地址空间中端口的地址作为一个操作数。 I / O指令分为两类：

* 那些传输位于寄存器中的单个项目（字节，字或双字）的文件。
* 那些传输位于内存中的字符串（字节，字或双字字符串）。 这些被称为“字符串I / O指令”或“块I / O指令”。

## 8.2.1 Register I/O Instructions

提供了I / O指令IN和OUT，以在I / O端口和EAX（32位I / O），AX（16位I / O）或AL（8位I / O）通用寄存器之间移动数据。 IN和OUT指令可直接对I / O端口进行寻址，并在指令中编码多达256个端口地址之一，或者通过DX寄存器间接对多达64K端口地址之一进行寻址。

Dx寄存器的使用：

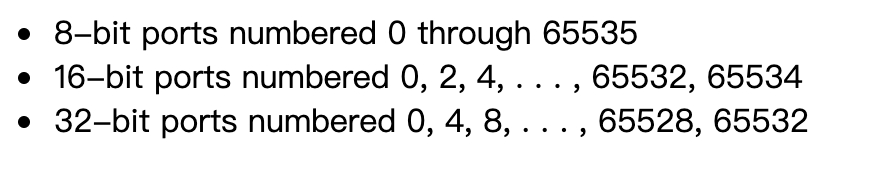


IN（端口输入）将字节，字或双字从输入端口传输到AL，AX或EAX。 如果程序使用IN指令指定AL，则处理器会将8位从所选端口传输到AL。 如果程序使用IN指令指定AX，则处理器会将16位从端口传输到AX。 如果程序使用IN指令指定EAX，则处理器会将32位从端口传输到EAX。

OUT（输出到端口）将字节，字或双字从AL，AX或EAX传输到输出端口。 程序可以使用与IN指令相同的方法指定端口号。

## 8.2.2 Block I/O Instructions

块（或字符串）I / O指令INS和OUTS在I / O端口和内存空间之间移动数据块。 块I / O指令使用DX寄存器在I / O地址空间中指定端口的地址。 INS和OUTS使用DX指定：



块I / O指令使用SI或DI来指定源或目标存储器地址。 对于每次传输，SI或DI都会根据标志寄存器中的方向位指定自动递增或递减。

INS和OUTS与重复前缀一起使用时，会导致块输入或输出操作。 REP（重复前缀）可修改INS和OUTS，以提供在I / O端口和内存之间传输数据块的方法。 这些块I / O指令是字符串原语（有关字符串原语的更多信息，请参阅第3章）。 它们无需使用单独的LOOP指令或中间寄存器来保存数据，从而简化了编程并提高了数据传输速度。

字符串I / O原语可以对字节字符串，单词字符串或双字字符串进行操作。 每次传输后，对于字节操作数，ESI或EDI中的内存地址更新为1，对于字操作数，更新为2，对于双字操作数，更新为4。 方向标志（DF）中的值确定处理器是自动递增ESI还是EDI（DF = 0），还是是否自动递减这些寄存器（DF = 1）。

INS（来自端口的输入字符串）将字节或单词字符串元素从输入端口传输到内存。助记符INSB，INSW和INSD是显式指定操作数大小的变体。如果程序指定INSB，则处理器会将8位从所选端口传输到ES：EDI指示的存储位置。如果程序指定INSW，则处理器会将16位从端口传输到ES：EDI指示的存储位置。如果程序指定INSD，则处理器会将32位从端口传输到ES：EDI指示的存储位置。不能为INS指令更改目标段寄存器选择（ES）。结合REP前缀，INS将信息块从输入端口移至一系列连续的存储位置。

OUTS（输出字符串到端口）将字节，字或双字字符串元素从内存传输到输出端口。助记符OUTSB，OUTSW和OUTSD是显式指定操作数大小的变体。如果程序指定OUTSB，则处理器会将8位从ES：EDI指示的存储位置传输到所选端口。如果程序指定OUTSW，则处理器会将16位数据从ES：EDI指示的存储位置传输到所选端口。如果程序指定OUTSD，则处理器会将32位从ES：EDI指示的存储位置传输到所选端口。结合REP前缀，OUTS将信息块从DS：ESI指示的一系列连续存储位置移至输出端口。

# 8.3 Protection and I/O

两种机制为I / O功能提供保护：

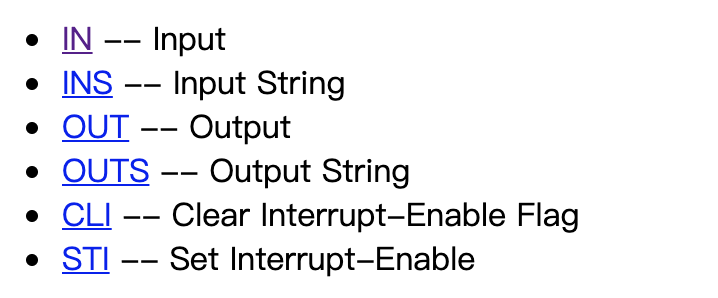
* EFLAGS寄存器中的IOPL字段定义使用与I / O相关的指令的权利。
* 80386 TSS段的I / O许可权位图定义了在I / O地址空间中使用端口的权利。

这些机制仅在保护模式下运行，包括虚拟8086模式。 它们不能在实模式下运行。 在实模式下，没有保护I / O空间。 任何过程都可以执行I / O指令，并且任何I / O端口都可以由I / O指令寻址。

## 8.3.1 I/O Privilege Level

处理I / O的指令需要受到限制，但也需要由特权级别（非零）执行的过程执行。 因此，处理器使用标志寄存器的两位来存储I / O特权级别（IOPL）。 IOPL定义了执行与I / O相关的指令所需的特权级别。

仅当CPL <= IOPL时，才能执行以下指令：



这些指令称为“敏感”指令，因为它们对IOPL敏感。

要使用敏感指令，过程必须以至少与IOPL指定的特权相同的特权级别执行（CPL <= IOPL）。特权较低的过程尝试使用敏感指令会导致一般保护异常。

因为每个任务都有自己的标志寄存器唯一副本，所以每个任务可以具有不同的IOPL。 IOPL为三，其主要功能是执行I / O（设备驱动程序）的任务可以从中受益，从而使该任务的所有过程都可以执行I / O。其他任务通常将IOPL设置为零或一，保留对特权最大的过程执行I / O指令的权利。

一个任务只能使用POPF指令来更改IOPL。但是，此类更改是特权。除非该过程在特权级别0上执行，否则任何过程都不能更改IOPL（标志寄存器中的I / O特权级别）。特权较低的过程尝试更改IOPL不会导致异常；否则，它不会导致异常。 IOPL保持不变。

除CLI和STI外，还可以使用POPF指令来更改中断使能标志（IF）。但是，POPF对IF的更改对IOPL敏感。仅当在至少与IOPL一样特权的级别上执行时，过程才能使用POPF指令更改IF。特权较低的过程尝试以这种方式更改IF不会导致异常。 IF保持不变。

## 8.3.2 I/O Permission Bit Map

直接引用处理器I / O空间中地址的I / O指令为IN，INS，OUT，OUTS。 80386具有选择性地捕获对特定I / O地址的引用的能力。 TSS段中的I / O权限位图是启用选择性陷阱的结构（请参见图8-2）。 I / O权限图是位向量。地图的大小及其在TSS段中的位置是可变的。处理器通过TSS固定部分中的I / O映射基字段来定位I / O许可映射。 I / O映射基字段为16位宽，包含I / O许可映射开头的偏移量。 I / O许可映射的上限与TSS段的上限相同。

在保护模式下，当处理器遇到I / O指令（IN，INS，OUT或OUTS）时，处理器首先检查CPL是否等于IOPL。如果满足此条件，则可能会进行I / O操作。如果不是，则处理器检查I / O权限映射。 （在虚拟8086模式下，处理器无需考虑IOPL即可查阅地图。请参阅第15章。）

映射中的每一位都对应一个I / O端口字节地址。例如，端口41的位位于I / O映射基数+ 5（位偏移量1）。处理器测试与I / O操作所跨越的I / O地址相对应的所有位；例如，双字操作测试对应于四个相邻字节地址的四个位。如果设置了任何测试位，则处理器会发出一般保护异常的信号。如果所有测试位均为零，则I / O操作可以继续进行。

I / O权限映射不必代表所有I / O地址。映射未覆盖的I / O地址被视为映射中只有一位。例如，如果TSS限制等于I / O映射基数+ 31，则将映射前256个I / O端口；否则，将映射前256个I / O端口。任何大于255的端口上的I / O操作都会导致异常。

如果I / O映射基数大于或等于TSS限制，则TSS段没有I / O许可映射，并且当CPL> IOPL时，80386程序中的所有I / O指令都会导致异常。

因为I / O权限映射在TSS段中，所以不同的任务可以具有不同的映射。因此，操作系统可以通过更改任务的TSS中的I / O权限映射为任务分配端口。

