计算机类

重点院校推荐教材

# 微型计算机原理与接口技术

WEIXING JISUANJI
YUANLI YU JIEKOU JISHU

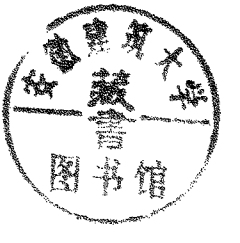
赵国相 于秀峰 编著

科学出版社
www.sciencep.com

重点院校推荐教材

# 微型计算机原理与接口技术

赵国相　于秀峰　编著




科学出版社

北京

## 内 容 简 介

本书将“微型计算机原理”、“微型计算机接口技术”和“汇编语言程序设计”三门课程的内容有机地融为一体。《微型计算机原理与汇编语言程序设计》和《微型计算机原理与接口技术》两本书为同一门课程连续使用的教材。本书以 Pentium 的实模式与保护模式为主线，用 Pentium 实模式的实现技术来替代 Intel 8086 的内容（目前流行以 Intel 8086 为基础）；通过分析 Pentium 的保护模式，把当今微机领域内具有代表性的新设计、新技术、新思想和新潮流展示给读者；列举了一定数量的 I/O 接口硬件及程序设计实例，有助于建立微机系统的整机概念，加深对微机工作过程的理解，使学生初步具有微机系统软、硬件开发的能力。

本书共 10 章。内容包括：Pentium 的存储管理；输入/输出；中断；总线；可编程接口芯片及应用；串行通信和可编程串行通信接口 8251；数模转换及模数转换；键盘接口技术；CRT 显示器接口技术；硬磁盘存储器。

本书可作为高等学校计算机科学与技术、通信工程、电气工程及自动化等专业的教材，也可供从事计算机应用工作的工程技术人员及其他自学者学习和参考。

**图书在版编目(CIP)数据**

微型计算机原理与接口技术/赵国相，于秀峰编著 .—北京：科学出版社，2004

重点院校推荐教材

ISBN 7-03-013025-1

Ⅰ.微… Ⅱ.①赵…②于… Ⅲ.①微型计算机-理论-高等学校-教材 ②微型计算机-接口设备-高等学校-教材 Ⅳ.TP36

中国版本图书馆 CIP 数据核字(2004)第 015940 号

责任编辑：鞠丽娜 马长芳/责任校对：包志虹

责任印制：吕春珉/封面设计：王 浩

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2004 年 4 月第 一 版 开本：B5(720×1000)

2004 年 4 月第一次印刷 印张：21 3/4

印数：1—5 000 字数：434 000

**定价：30.00 元**

（如有印装质量问题，我社负责调换〈环伟〉）

# 前　言

微机原理、汇编语言程序设计及接口技术三部分内容是计算机科学与技术、通信工程、电气工程及自动化等专业的核心课程。在以前的教学体系中，大部分院校都将其分成三门课，即“微机原理及应用”、“微型计算机接口技术”和“汇编语言程序设计”。随着集成电路技术的飞速发展，许多大型计算机甚至巨型计算机的成熟技术已逐步下移至微型计算机，促使微型计算机发展非常快，随之带来两个问题：一是微型计算机的结构日趋复杂，这就使微机原理、汇编语言程序设计及接口技术三部分内容彼此相关的程度更加密切、互相交融；二是新课程及新内容不断增加，每门课程的学时越来越少，使得旧的内容删不掉，新的内容又加不进来，于是出现了教学内容与实际严重脱节的现象，家用微机早已使用奔腾(Pentium)微处理器，而课堂上仍在讲 Intel 8088/8086 微处理器。若仍将微机原理、汇编语言程序设计及接口技术三部分内容分为三门课，势必造成在内容上时有冲突，有些内容学生不得不学两遍，甚至还要多，有时还会造成对某些问题或概念理解得不透彻。所以，改革目前微机课程教学体系，把微机原理、汇编语言及接口技术合为一体来讲授，势在必行。

本书将“微型计算机原理”、“微型计算机接口技术”和“汇编语言程序设计”三门课程的内容有机地融为一体。《微型计算机原理与汇编语言程序设计》和《微型计算机原理与接口技术》两本书为同一门课程连续使用的教材。它是在将三门课程合为一门(即“微型计算机原理、汇编语言、接口技术”)的三次教学实践基础上进行修改整理而成的，实际上也是我们二十几年来从事这三门课程的教学总结。本书以 Pentium 的实模式与保护模式为主线，用 Pentium 实模式的实现技术来替代 Intel 8086 的内容(目前流行以 Intel 8086 为基础)；通过分析 Pentium 的保护模式，把当今微机领域内具有代表性的新设计、新技术、新思想和新潮流展示给读者；列举了一定数量的 I/O 接口硬件及程序设计实例，有助于建立微机系统的整机概念，加深对微机工作过程的理解，使学生初步具有微机系统软、硬件开发的能力。

《微型计算机原理与汇编语言程序设计》一书共分 8 章。第 1 章，主要讲述微处理器发展简况，分别介绍第 1～4 代微处理器 Intel 8008、Zilog 的 Z80、Intel 8086、Intel 80386 的基本结构和功能特征。第 2 章主要介绍 Pentium～Pentium Ⅳ微处理器的编程结构及功能，讲述总线接口、预取缓冲部件、整数流水线、浮点流水线、Cache 部件、指令译码部件、控制部件、分段部件、分页部件。超流水线和超标量技术流水线结构、指令译码操作、寄存器重命名技术、乱序执行技术、退出流水线操作、饱和运算、积和运算能力、动态执行技术，Pentium 微处理器的引脚功能、

Pentium微处理器的基本时序(非流水线式读/写周期、突发式读写总线周期、流水线式读写总线周期)。第3章讲述在16位模式及32位模式的指令格式、寻址方式和指令系统。第4章讲述汇编语言程序格式、伪指令和汇编语言上机过程。第5章讲述双分支程序设计、多分支程序设计、循环程序设计的结构、循环程序设计方法和多重循环程序设计。第6章讲述子程序的结构、子程序的参数传递方法、子程序的嵌套与递归和子程序设计举例。第7章讲述宏汇编、重复汇编、条件汇编、模块化程序设计。第8章讲述半导体存储器的分类及性能指标、ROM及RAM存储芯片、Pentium的存储器接口、Pentium的高速缓冲存储器(Cache)及二级Cache与一级Cache的关系等。

《微型计算机原理与接口技术》一书共10章。第1章主要讲述虚拟存储器、Pentium分段存储管理和分页存储管理。第2章主要介绍为什么要用接口电路、I/O接口的一般编程结构、CPU与外设之间数据传送的控制方式(程序查询传送方式、程序中断方式、DMA传送方式、I/O处理机方式)、DMA控制器8237A及其应用。第3章主要讲述中断的基本概念、中断接口电路、中断处理过程、Pentium中断机制、实模式中断处理过程、保护模式中断操作和可编程中断控制器8259A。第4章主要讲述总线的概念及分类、ISA总线、PCI总线。第5章讲述可编程并行输入/输出接口芯片8255A、8255A各种工作方式的应用举例、可编程计数器/定时器8253及其在计数和实时测频系统中的应用举例。第6章讲述数字串行通信系统模型、RS-232-C串行通信接口总线、通用串行总线USB简介、可编程串行通信接口芯片8251A、串行通信系统实例。第7章主要讲述实时微机控制系统的硬件结构、传感器、数/模转换器及应用、模/数转换器及应用、功率开关器件及接口。第8章讲述键盘的结构、键的识别(行扫描法、行反转法)、微机与键盘的接口、BIOS键盘中断及DOS键盘功能调用。第9章讲述CRT显示器的工作原理、黑白字符显示器的基本原理、CRT控制器、IBM PC系列机的显示系统(MDA适配器、CGA适配器/EGA、VGA适配器)、对显示器的编程。第10章讲述数据磁记录的基本原理、硬磁盘存储器类型、硬磁盘上信息的分布、硬磁盘驱动器、硬磁盘控制器、硬磁盘接口、磁盘文件存取技术(DOS文件代号式磁盘存取、BIOS磁盘文件存取)。

在本书的写作过程中得到了张长海、胡成全教授的大力支持与帮助、在此表示感谢!

由于作者水平有限,书中难免有错误和不当之处,恳请读者和同行专家批评指正。

作　者

2003年11月于吉林大学

# 目　录

# 第 1 章　Pentium 微型计算机的存储器管理

## 1.1　概　　述

虚拟存储器（virtual memory）又称为虚拟存储系统。虚拟存储器是为满足用户对存储空间不断扩大的要求而提出的，随着用户程序复杂性的增加，占用存储空间越来越大。其解决办法是，可扩大主存，但是造价高，空间利用率很低，并非是好的途径。采用虚拟存储器，可圆满地解决这个问题。虚拟存储器这个概念是 1961 年由英国曼彻斯特大学的 Kilburn 等人提出的，并于 20 世纪 70 年代广泛应用于大中型计算机之中，现在的微型计算机也都采用了这种技术。虚拟存储器由主存储器和辅助存储器共同组成。它把辅助存储器作为主存储器的扩充，对应用程序员来说，好像计算机系统有一个容量很大的主存。虚拟存储系统的目标是为了增加存储器的存储容量，它的速度接近于主存，单位造价接近于辅存，因此性能价格比很高。

### 1.1.1　地址空间及地址

我们已知道，CPU 只能执行已装入主存的那一部分程序块，与此同时，为了提高主存的空间利用率，还应及时释放已不使用的信息在主存中所占用的空间，以便于装入其他有用的信息。这样，随着程序的运行，各种信息就会在主存与辅存之间不断地调进、调出。

在虚拟存储器中有 3 种地址空间及对应的 3 种地址。虚拟地址空间又称为虚存地址空间，是应用程序员用来编写程序的地址空间，与此相对应的地址称为虚地址或逻辑地址；主存地址空间又称为实存地址空间，是存储、运行程序的空间，其相应的地址称为主存物理地址或实地址；辅存地址空间也就是磁盘存储器的地址空间，是用来存放程序的空间，相应的地址称为辅存地址或磁盘地址。

### 1.1.2　工作原理

虚拟存储器由硬件和软件（操作系统）自动实现对存储信息的调度和管理，其工作过程如图 1.1.1 所示。

当应用程序访问虚拟存储器时，必须给出逻辑地址，首先进行内部地址变换。如果要访问的信息在主存中（也就是内部地址变换成功），则根据变换所得到的物理地址访问主存储器；如果内部地址变换失败，则要根据逻辑地址进行外部地址变换，得到辅存地址。与此同时，还需检查主存中是否有空闲区，如果没

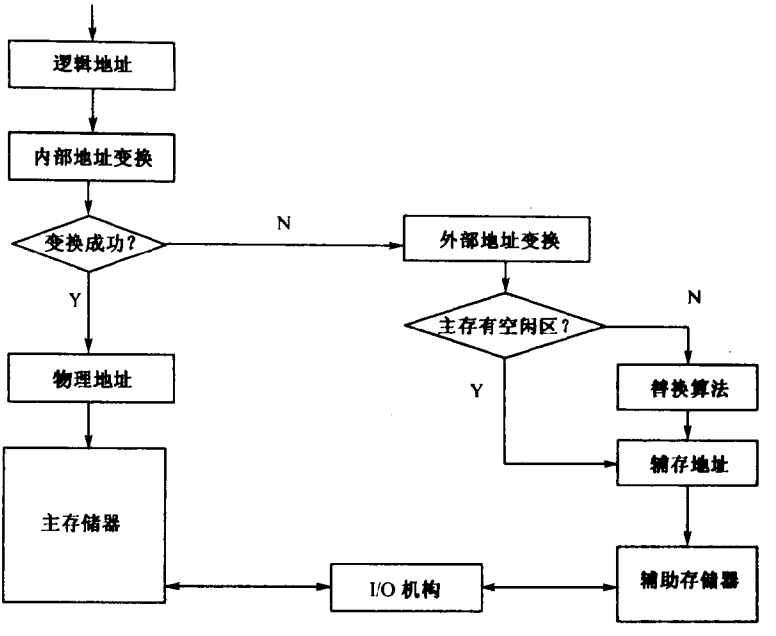



图 1.1.1　虚拟存储器工作过程示意图

有，就要根据替换算法，把主存中暂时不用的某块信息通过I/O机构调出，送往辅存，再把得到的辅存地址中的信息块送往主存；如果主存中有空闲区域，则直接把辅存中有关的信息块送往主存。

由于采用的存储映象算法不同，就形成多种不同存储管理方式的虚拟存储器，其中主要有段式、页式和段页式 3 种。尽管使用的存储管理方式有不同，作为虚拟存储器的基本原理、工作过程及有关技术问题还是有许多相似之处的。

Pentium 支持分段存储管理、分页存储管理和段页式存储管理。Pentium 微处理器片内存储管理部件负责对物理存储器实施安全可靠且行之有效的存储管理操作。当存储管理部件正常运转时，程序是不能直接对物理存储器进行寻址操作的，只能对一个被称之为虚拟存储器的存储器模型进行寻址操作。Pentium 微处理机的存储管理部件是由分段部件和分页部件组成的。分段部件是一种可以提供多个各自独立的地址空间机构。而分页部件使用少量的随机存储器（RAM）和磁盘存储器去支持一个很大的地址空间模型的存储管理机构。Pentium 微处理机的分段部件和分页部件既可以单独使用其中的一种，也可以两种同时使用。由程序提供的地址叫做逻辑地址。分段部件的功能之一就是将一个逻辑地址转换成一种连续的不分段的地址空间，这种地址叫做线性地址。而分页部件的主要功能就是将线性地址转换成物理地址。

## 1.2 Pentium 的分段存储管理

### 1.2.1 分段存储管理的基本思想

通常，一个程序由多个模块组成，特别是在结构化程序设计思想提出之后，程序的模块性就更强了。一个复杂的大程序总可以分解成多个在逻辑上相对独立的模块，模块间的界面和调用关系是可以清楚定义的。这些模块可以是主程序、各种能赋予名称的子程序或过程，也可以是表格、数组、树、向量等某类数据元素的集合。模块的大小可以各不相同，有的甚至事先无法确定。但每一个模块都是一个具有特定功能的独立的程序段，都是以该段的起点为 0 相对编址。当某一程序段（模块）从辅存调入主存时，只要由系统赋予该段一个基址，就可以把基址和每个单元在段内的相对位移量组合起来，形成这些单元在主存中各自的实际地址。把主存按段分配的存储管理方式就称为段式管理。程序进入内存时，各程序段要求占据相对独立的内存区间。因此，现代微机系统把物理空间分成相对独立的许多内存段，每个内存段放置一个程序段，至此内存段与程序段统一，统称为段。一个程序拥有多个段，不同程序占据不完全相同的几个段。而且管理系统所需要的信息放置在属于系统所有的段中。分段管理后，系统必须知道每个段的必要信息（段信息）才能完善地管理各段，这些信息包括：段在物理空间的开始地址、段的界限，段是数据型还是程序型，内存标志等多方面的内容。32 位机把每个段的段信息放入一个数据结构中，称作段描述符（或简称描述符）。又把所有的段描述符分类，组成顺序排列的表，称作段描述符表（或简称段表）。每个段描述符表都放在内存中备用。显然每个段描述符表占据的物理空间也形成一个段。每个段描述符表也有自己的描述符，称作描述表描述符。

1. 分段存储管理工作过程

图 1.2.1 给出了分段存储管理的示意图。一个程序 A 具有 4 个模块，程序 A 对应的段描述符表中有 4 个段描述符，每个模块对应一个段描述符。段表的一行为一个段描述符，段描述符内容包括基址、界限和访问控制等。基址是装入模块的首地址，界限指出该段的长度。

由图 1.2.1 不难看出：当需要访问某程序中的一个信息时，第一步要从内存中找到段描述符表；第二步从段描述符表中找到相应的段描述符，由段描述符中的内存标志指出该段目前是否在内存，若内存标志不成立时，系统就知道该段目前不在内存，系统去外存寻找此程序段；若找到就把它调入内存，然后修改段描述符以便与内存段统一；第三步从段描述符找到段在内存的位置；第四步从段中找到信息所在内存的物理地址；第五步访问该物理地址。由此可知，只要系统建立了一个程序段的描述符，系统就开始管理此程序段，而无论该段内容是否真正

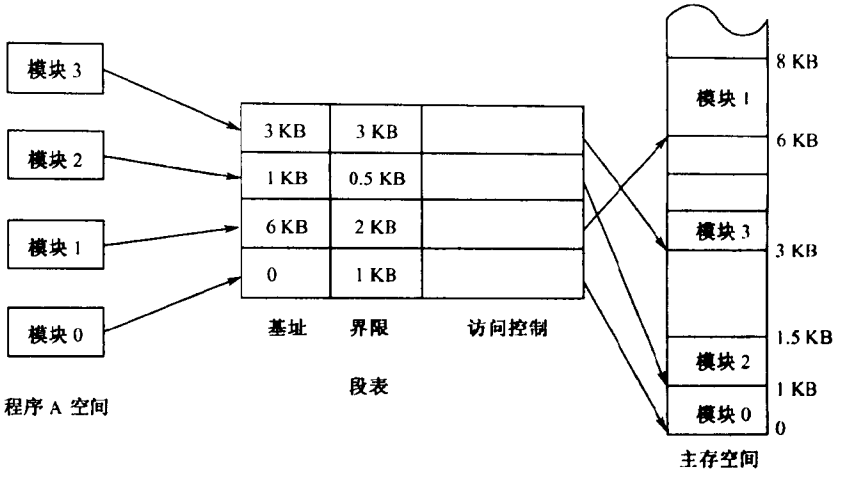



图 1.2.1　分段存储管理的示意图

在内存。也就是说，此标志位使系统可以把外存的一部分作为内存的延伸，与内存统一管理。这一复合存储空间称为虚拟内存。程序段只要进入虚拟内存就可以被系统管理，自动调入内存执行。因此程序员编程时使用虚拟空间即可，无需考虑物理空间大小，因此常称虚拟空间为编程空间。

2．虚拟地址和虚拟地址空间

Pentium 微处理机在保护模式下的存储器管理单元使用 48 位的存储器指针（图 1.2.2），它分为段选择符（或简称为选择符）和偏移量两部分。该 48 位存储器指针称为虚拟地址，它在程序中用以规定指令或数据的存储器位置。段选择符 16 位长，偏移量 32 位长。段选择符可放在 Pentium 微处理机段寄存器中。若要访问存储器中的代码，则段选择符应放在 CS 中；若要访问存储器中的数据，则段选择符应放在 DS、GS、ES、FS、SS 中的任意一个。指针的这一部分选择的是 Pentium 微处理机虚拟地址空间中一个特定的段。

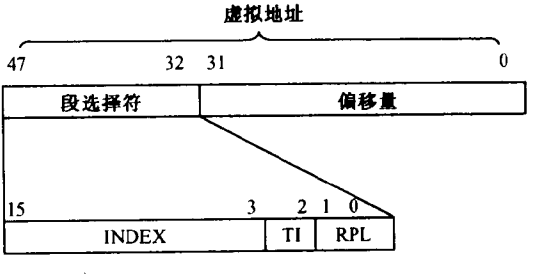



图 1.2.2　保护模式下的存储器指针及段选择符格式

偏移量放在 Pentium 微处理机的用户可访问的寄存器中。若要访问存储器中的代码，则偏移量放在 EIP 寄存器中。若要访问存储器中的数据，则偏移量放在 EAX、EBX、ECX、EDX、ESI、EDI 等寄存器中。由于偏移量是 32 位长，段大小可达 4GB（一个字节已被定义为 8 位二进制代码，用 B 来表示一个字节）。我们说段大小可达 4GB 是因为段大小实际上是可变的，它可从 1B 到 4GB。

图 1.2.2 又说明了 16 位的段选择符内分为 3 个字段，13 位索引字段，1 位表选择字段 TI 和 2 位的请求特权级字段 RPL。2 位的 RPL 并不用于存储器段选择，因此，16 位中只有 14 位用于寻址，这样虚拟地址空间可容纳 $2^{14}$（16K）个存储器段，每个段最大可达 4GB。这些段就是 Pentium 微处理机存储器管理所管理的虚拟地址空间的基本元素。

另外一种计算虚拟地址空间大小的办法是将 14 位段选择符和 32 位偏移量结合起来，得到 46 位虚拟地址，从而 Pentium 微处理机的虚拟地址空间可包含 $2^{46}$（64T）个字节。

3. 虚实地址转换

Pentium 微处理机的分段存储管理机制允许将 46 位虚拟地址映射到硬件所需的 32 位物理地址。该地址转换大致过程如图 1.2.3 所示。首先由虚拟（逻辑）地址段选择符部分的 13 位索引字段确定段描述符在段表（也称段描述符表）的位置；然后取出段描述符中的 32 位基地址并与偏移量相加即得到 32 位的线性地址。如果不启用分页功能，则线性地址就直接作为物理地址。

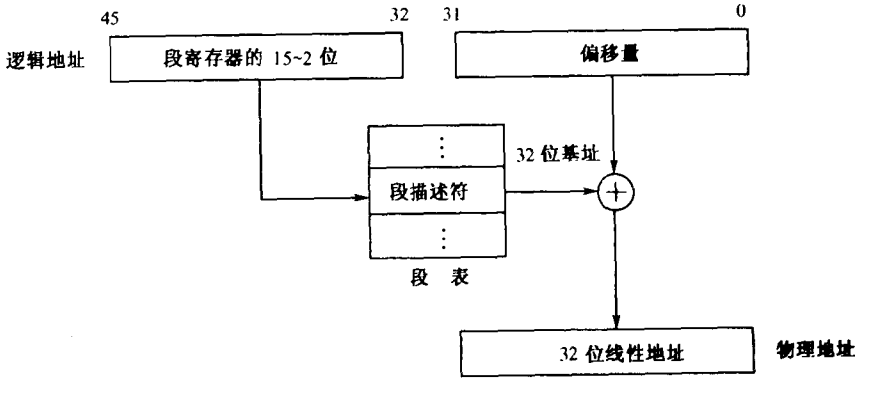



图 1.2.3 虚实地址转换示意图

## 1.2.2 段描述符

段描述符是 Pentium 微处理机存储管理硬件用以管理 64T 字节虚拟存储地址空间分段的基本元素。一个段描述符对应于虚拟地址空间中的一个存储器段。段

描述符是位于主存中的一种数据结构，由系统程序创建，它为处理机提供段的基本信息。所有段描述符均由 8 个字节组成。段描述符内保存着供处理机使用的有关段的属性、段的大小规模、段在存储器中的位置以及控制和状态信息。一般说来，各段描述符都是由各种编译程序、各种连接程序、各种装入程序或者是操作系统产生的，而不是由各种应用程序生成的。段描述符按段的性质可分为程序段描述符、系统段描述符和门描述符，如图 1.2.4 所示。其中程序段描述符又分为代码段描述符、堆栈段描述符和数据段描述符。系统段描述符又称为特殊段描述符，包括局部描述符表（LDT）描述符和任务状态段描述符（TSS）。门描述符包括任务门描述符、调用门描述符、中断门描述符和陷阱门描述符。对于不同的描述符，其格式存在差异。

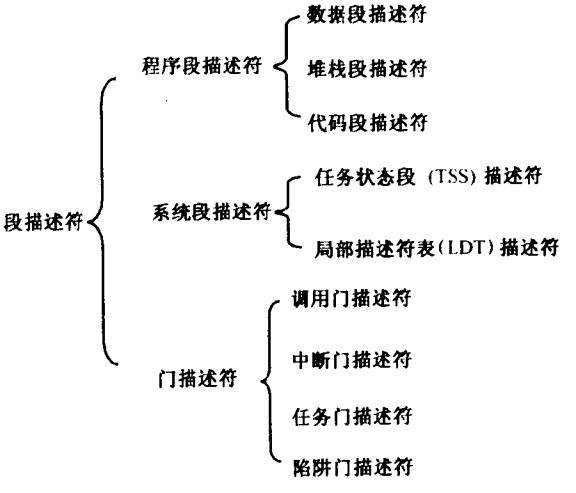



图 1.2.4　段描述符的分类

1. 程序段描述符

程序段描述符的格式如表 1.2.1 所示。

（1）基地址字段

Pentium 微处理机用这个字段来规定某一个段在 4GB 物理地址空间中的位置。段描述符的第 2～4 和第 7 字节组成了 32 位的基址字段，这个基址可以访问 4G（$2^{32}$）字节的主存空间。

（2）段界限字段

段描述符中的段界限字段是用来定义段的大小规模。段描述符的第 0、1 字节和第 6 字节的低 4 位是 20 位的段界限字段，该字段的值决定了段的长度，而该字段的值的单位由“G”位决定；“G”位称作粒度位，用来确定段界限所使用的长度单位。

表 1.2.1 程序段描述符的格式

| D7 | | | | | D0 | 字节 |
|---|---|---|---|---|---|---|
| 段界限 7～0 | | | | | | 0 |
| 段界限 15～8 | | | | | | 1 |
| 基址 7～0 | | | | | | 2 |
| 基址 15～8 | | | | | | 3 |
| 基址 23～16 | | | | | | 4 |
| P | DPL | | S | TYPE | | 5 |
| G | D/B | 0 | AVL | 段界限 19～16 | | 6 |
| 基址 31～24 | | | | | | 7 |

(3) 粒度位 G 字段

段描述符中的这个字段是用来确定段界限所使用的长度单位。段描述符的第 6 字节的 D7 位是粒度位 G 字段。当 G=0 时，段的长度以一个字节为单位；当 G=1 时，段的长度以 4K ($2^{12}$) 字节为单位。当 G=0 时，段界限字段值的范围可从 1B 到 1MB (因为段界限为 20 位) 范围。在这种情况下，段界限字段的值可在 1B 的基础上，每次增值 1B。当 G=1 时，段界限字段值的范围可从 4KB 到 4GB。在这种情况下，段界限字段的值可在 4KB 的基础上，每次增值 4KB。

(4) 分类 S 字段

段描述符中的第 5 字节的 D4 位 "S" 字段是用来区分是系统段描述符还是非系统段描述符。当 S=0 时，是系统段描述符；当 S=1 时，是非系统段描述符。

(5) 段存在位 P 字段

P 字段表示该段是否在内存中。段描述符的第 5 字节的 D7 位是 P 字段。当 P=1 时，表示该段在内存中；当 P=0 时，表示该段不在内存中。

(6) 系统可用位 AVL 字段

AVL 字段表示系统软件是否可用本段。段描述符的第 6 字节的 D4 位是 AVL 字段。当 AVL=1 时，表示系统软件可用本段；当 AVL=0 时，表示系统软件不能用本段。

(7) 特权级 DPL 字段

这个字段用来定义段的特权级。段描述符的第 5 字节的 D6、D5 位是 DPL 字段。DPL 字段占有 2 位，故有 4 个特权级，00、01、10、11，分别称作 0 级、1 级、2 级、3 级。0 级的特权最高，1 级次之，3 级的特权最低。借助于保护机构用这个字段定义的特权级去控制对这个段的访问，其访问原则将保证高特权级的代码或数据不被低特权级的程序所破坏。