

<<Pentium汇编语言程序设计>>

图书基本信息

书名：<<Pentium汇编语言程序设计>>

13位ISBN编号：9787040178135

10位ISBN编号：7040178133

出版时间：2007-12

出版时间：高等教育出版社

作者：张长海,郭东伟,左万历,赵国相

页数：313

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Pentium汇编语言程序设计>>

前言

汇编语言是一种介于机器语言和高级语言之间的计算机编程语言，它既不像机器语言那样直接使用计算机所认识 and 理解的二进制代码，也不像高级语言那样独立于硬件之外直接面向用户。汇编语言是计算机提供给用户的最快而又最有效的语言，也是能够利用计算机所有硬件特性并能直接控制硬件的语言，利用汇编语言可以编写出在空间和时间上都高效的程序。

在很多需要直接控制硬件的场合，更是非用汇编语言不可。

当前绝大部分微型机使用Intel体系结构的微处理器，这种体系结构的微处理器目前已经发展到Pentium，因此Pentium汇编语言程序设计已经成为计算机科学与技术专业的一门重要课程。

目前，全国高等教育正处于教学改革的时期，新的教学思路、新的课程体系和教学内容正在形成。结合21世纪计算机科学与技术专业课程改革的基本思路，在多年教学实践的基础上，我们编写了《Pentium汇编语言程序设计》，它是教育部《高等教育面向21世纪教学内容和课程体系改革计划》的研究成果之一。

本书以Pentium微处理器为基础，系统地介绍Pentium微处理器的编程结构、指令系统、汇编语言程序设计的基础知识、程序设计方法和应用技术。在浮点运算和系统程序编制方面的扩充是本书的特色。

<<Pentium汇编语言程序设计>>

内容概要

《汇编语言程序设计》以Pentium微处理器为主线，系统地介绍了汇编语言程序设计的基础知识、程序设计方法和应用技术。

《汇编语言程序设计》由基础理论、编程工具、编程方法和实际应用4部分共11章组成。包括Pentium微处理器的基本体系结构；Pentium微处理器的指令格式、寻址方式、指令系统、伪指令和汇编语言程序格式；汇编语言进行程序设计的基本方法，包括顺序、分支、循环、子程序等基本程序结构、编程方法和技巧；输入输出、异常、中断程序设计、文件系统；在Pentium芯片支持下算术程序设计方法，Pentium系统程序设计方法等。

在浮点运算和系统程序编制方面的扩充是《汇编语言程序设计》的特色。

《汇编语言程序设计》内容丰富、充实，由浅入深，循序渐进的讲述方式适合于具有一定程序设计基础的初学者学习汇编语言程序设计方法。

《汇编语言程序设计》可作为高等学校计算机科学与技术、通信工程、电气工程、自动化等专业“汇编语言程序设计”课程的教材，也可供自学者及从事计算机应用的工程技术人员使用、学习和参考。

<<Pentium汇编语言程序设计>>

书籍目录

第一章 Pentium微处理器1.1 计算机系统组成1.1.1 硬件1.1.2 软件1.2 Intel系列CPU简介1.2.1 Intel系列CPU发展简史1.2.2 Pentium家族的其他成员1.3 Pentium微处理器的基本结构1.4 Pentium寄存器1.4.1 基本结构寄存器1.4.2 系统级寄存器1.4.3 调试寄存器1.4.4 模型专用寄存器1.4.5 浮点处理部件本章小结习题一第二章 Pentium微处理器指令系统2.1 Pentium汇编语言汇编指令格式2.2 Pcntium微处理器内部指令格式2.2.1 前缀2.2.2 操作码2.2.3 寻址方式2.2.4 位(偏)移量及立即数2.3 寻址方式2.3.1 数据的寻址方式2.3.2 转移地址的寻址方式2.3.3 堆栈地址寻址2.4 指令系统2.4.1 数据传送指令2.4.2 算术运算指令2.4.3 BCD码调整指令2.4.4 逻辑运算指令2.4.5 位处理指令2.4.6 控制转移指令2.4.7 条件设置指令2.4.8 串操作指令2.4.9 处理器控制指令本章小结习题二第三章 Pentium宏汇编3.1 Pentium汇编程序结构3.1.1 程序3.1.2 模块3.1.3 段3.1.4 语句3.2 数据定义——数据段3.2.1 标识符3.2.2 数3.2.3 字符串3.2.4 符号常量3.2.5 变量及变量说明3.2.6 表达式3.2.7 数组3.3 堆栈定义——栈段3.4 指令代码——代码段3.4.1 段寄存器初值3.4.2 程序定位3.4.3 标号3.4.4 程序启动地址和结束地址3.4.5 处理器选择3.5 段分组3.6 简化段3.6.1 存储模式3.6.2 段定义3.6.3 预定义符号3.6.4 两点说明3.6.5 实例3.7 汇编语言程序上机过程3.7.1 用EDIT建立.ASM文件3.7.2 使用MASM产生.OBJ等文件3.7.3 用LINK程序产生.EXE文件3.8 可执行程序3.8.1.COM文件3.8.2.EXE文件本章小结习题三第四章 汇编语言程序设计方法4.1 顺序程序设计4.2 分支程序设计4.2.1 IF语句4.2.2 CASE语句4.3 循环程序设计4.3.1 WHILE循环4.3.2 REPEATUNTIL循环4.3.3 FOR循环4.3.4 循环嵌套4.3.5 循环结构的限制本章小结习题四第五章 子程序和宏5.1 带子程序的程序5.2 变量分配5.2.1 在子程序代码段内分配5.2.2 在数据段的公共数据区分配5.2.3 在栈中分配5.3 参数传递5.3.1 值参数与变量参数5.3.2 传送方式5.3.3 子程序作参数5.4 递归5.5 栈的使用技巧与子程序的限制5.6 可重入性5.7 宏5.7.1 宏定义和扩展5.7.2 参数5.7.3 算子5.7.4 宏嵌套5.7.5 涉及宏的伪指令5.8 库5.8.1 子程序库5.8.2 宏库5.8.3 宏与子程序的区别本章小结习题五第六章 输入,输出程序设计6.1 I/O寻址概述6.1.1 I/O设备及其数据传送方式6.1.2 I/O接口和I/O端口地址6.2 I/O指令6.2.1 寄存器I/O指令6.2.2 块I/O指令6.3 简单I/O程序设计6.4 查询方式I/O程序6.4.1 异步串行通信介绍6.4.2 异步串行通信接口6.4.3 I/O宏指令和端口定义6.4.4 查询方式的串行I/O程序本章小结习题六第七章 异常和中断7.1 概述7.1.1 异常和中断的概念7.1.2 异常和中断的类型7.1.3 向量号和中断向量表7.1.4 优先级7.2 异常和中断处理程序设计7.2.1 异常和中断处理程序的一般结构7.2.2 编写异常和中断处理程序7.2.3 用户自行安排中断程序处理程序7.3 BIOS中断程序设计7.3.1 概述7.3.2 利用BIOS中断控制键盘7.3.3 利用BIOS中断控制显示7.3.4 利用BIOS中断进行磁盘读写7.4 DOS功能调用7.4.1 键盘功能7.4.2 显示功能本章小结习题七第八章 文件系统8.1 DOS文件系统概述8.2 文件句柄方式操作8.3 基本文件服务功能8.3.1 建立文件8.3.2 打开文件8.3.3 关闭文件8.3.4 读文件8.3.5 写文件8.3.6 文件指针操作8.4 文件管理功能8.4.1 删除文件8.4.2 文件更名8.4.3 取出/设置文件属性8.4.4 文件查找本章小结习题八第九章 浮点运算9.1 数字协处理器和IEEE浮点数9.1.1 IEEE浮点标准9.1.2 其他数据格式9.2 数字协处理器编程初步9.2.1 寄存器结构9.2.2 简单编程举例9.2.3 整型数操作9.3 浮点运算状态9.4 浮点运算指令一览本章小结习题九第十章 保护模式10.1 Pentium指令系统的发展10.2 保护模式简介10.3 存储管理机制10.4 保护机制10.4.1 任务保护10.4.2 输入/输出指令保护10.5 中断与异常10.6 操作系统指令10.7 工作模式与模式切换本章小结习题十第十一章 汇编语言高级技术11.1 结构11.1.1 结构的定义11.1.2 结构变量的定义11.1.3 结构变量及其字段的访问11.2 记录11.2.1 记录的定义11.2.2 记录变量的定义及初始化11.2.3 涉及记录的运算11.2.4 访问记录或字段11.3 条件汇编与重复汇编11.3.1 条件汇编11.3.2 重复汇编11.4 多模块程序设计11.4.1 模块化程序设计11.4.2 模块的连接11.4.3 模块之间的参数传送11.5 汇编语言与高级语言的连接11.5.1 概述11.5.2 C语言程序与汇编语言程序连接时的系统规则11.5.3 连接方法本章小结附录附录一 ACSII字符集附录二 键盘扫描码表附录三 BIOS中断附录四 DOS中断INT21H功能调用参考文献

章节摘录

10.2保护模式简介 保护模式不仅可以充分发挥CPU的计算能力,扩展寻址范围,更重要的是保护模式提供了一系列硬件支持,相应的CPU也提供了硬件实现机制,来保证现代操作系统和应用程序的需要。

由于从80386开始在后续的80486及Pentium等CPU中,保护模式基本相同,因此为行文方便,在很多资料中都称为386保护模式,例如,在Windows NT的手册和文件目录中称i386保护模式,虽然Windows NT并不能在80386下运行,只能在80486以上机型运行。

最原始的以DOS为代表的操作系统,是单用户、单进程的。

也就是说,在一个时刻只能有一个应用程序在运行。

虽然可以使用各种技术在后台装载一些程序,但是这些程序和操作系统共享一套内存。

也就是说,各个程序都可以访问某一个相同内存地址。

因此某个程序的崩溃很容易导致整个操作系统的崩溃。

在这种方式下,系统的所有软硬件资源对应用程序都是可见的,也都是可操作的,由此给系统带来了很大的不安全性。

现代操作系统,如UNIX和Windows都是多用户、多进程的,可以同时运行多个应用程序。

站在每个应用程序角度看,它们都使用一致的地址空间,但是实际上操作系统把不同程序的内存空间映射到不同的物理内存上,以保证互不冲突。

原则上应用程序之间不能互相访问,也不能访问操作系统的内存,一个进程的崩溃不能导致整个系统的崩溃。

另外,对于很多特定硬件,应用程序不应该直接访问,而应该通过相应的操作系统调用来完成。

如果应用程序有意或无意地试图违反上述约定,则产生异常中断,转入操作系统进行处理。

另外,虽然32位处理器可以提供高达4 GB的内存空间,但计算机硬件不一定配备这么多数量的内存,同时多个应用程序加上操作系统使用的内存容量总和也完全可能超过实际内存数量,甚至超过4 GB,因此引入了虚拟内存机制。

虽然虚拟内存机制的实现主要由操作系统来完成,但是处理器必须提供相应的支持。

综上所述,在保护模式下,要提供的硬件机制包括存储管理机制、任务保护机制、中断和异常机制、I/O等特权指令管理以及提供进程切换、运行模式切换的支持。

下面就从这几个方面来介绍Pentium保护模式。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>