

关于 80X86 保护模式下任务切换的实现

刘丽莉¹, 万 春²

(1. 集美大学计算机与通信工程系, 福建 厦门 361021; 2. 集美大学网络中心, 福建 厦门 361021)

摘要: 首先简要介绍了 80X86 保护模式的特点, 然后介绍了该模式下相关的数据结构和任务切换的途径, 最后介绍了任务切换的实现过程。

关键词: 保护模式; 任务切换; 汇编语言

中图分类号: TP311.11 **文献标识码:** A

The Implement of Task Switch in Protected Mode Programming on Intel 80X86

LIU Li-li¹, WAN Chun²

(1. Department of Computer Science and Communication Engineering, Jimei University, Xiamen 361021, China;

2. Network Center, Jimei University, Xiamen 361021, China)

Abstract: Introduces characters of protected mode programming on Intel 80X86, then describes data structure and methods of task switch. Finally, gives an implement procedure of task switch in protected mode programming.

Keywords: protected mode; task switch; assemble language

0 引 言

汇编语言程序以其运行速度快、占用空间少、能直接和有效地操纵硬件而著称, 但汇编语言程序编写繁琐、调试困难。在实际应用中, 有时为了编制效率更高的程序, 或需要直接和有效地控制硬件, 需要编写汇编语言程序, 其中保护模式下任务的切换是经常遇到的问题。下面说明保护模式下任务切换的实现。

1 80X86 保护模式介绍

在保护模式下, CPU 的全部 32 根地址线有效, 可寻址 4GB 字节的物理地址空间; 扩充的存储器分段管理机制和可选的存储器分页管理机制, 不仅为存储器共享和保护提供了硬件支持, 而且为实现虚拟存储器提供了硬件支持; 支持多任务, 能够快速地进行任务切换和保护任务环境; 4 个特权级和完善的特权检查机制, 既能实现资源共享又能保证代码及数据的安全保密及任务的隔离。

2 相关数据结构

在进行保护模式编程时, 要涉及到一些数据结构, 理解和正确使用这些数据结构是编程的基础。现简要介绍如下。

2.1 段描述符

在保护模式下, 每个段由如下三个参数进行定义: 段基地址(Base Address)、段界限(Limit)和段属性(Attributes)。用于表示上述定义段的三个参数的数据称为描述符。每个描述符长 8 个字节。在保护模式下, 每个段都有一个相应的描述符来描述。按描述符所描述的对象来分, 描述符可分为如下三类: 存储段描述符、系统段描述符、门描述符。

存储段是存放可由程序直接进行访问的代码和数据的段, 存储段描述符描述存储段, 存储段描述符也被称为代码和数据段描述符。其格式如下:

第 8 字节		第 1 字节	
Base Address	Attributes	Base Address	Limit
31~24	15~0	23~0	15~0

收稿日期: 2001-12-28

作者简介: 刘丽莉(1965), 女, 河南人, 集美大学计算机与通信工程系讲师, 大学本科, 研究方向: 单片机应用与开发。
?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

段界限(Limit)共 20 位,其中 19~16 位包含在属性字节(Attributes)中。限于篇幅,两个字节长的属性字段的各位不再详细介绍,需要说明的是,其第 5、6 位(DPL)表示描述符特权级,它规定了所描述段的特权级,用于特权检查,以决定对该段能否进行访问;第 4 位(DT)说明描述符的类型。对于存储段描述符而言,DT=1,DT=0 表示是系统段描述符和门描述符。

可用定义结构的伪指令 STRUC 加以定义:

DESCRIPTOR	STRUC	
LIMITL	DW	0
BASEL	DW	0
BASEM	DB	0
ATTRIBUTES	DW	0
BASEH	DB	0
DESCRIPTOR	ENDS	

系统段是为实现存储管理机制所使用的一种特殊的段。有两种系统段:任务状态段 TSS 和局部描述符表 LDT 段。系统段描述符的格式与存储段描述符很相似,不再介绍。

例如,如下描述符 DemoTss 描述一个任务状态段,其基地址为 123456H,以字节为单位的界限为 104H,DPL=0。

```
DemoTss  DESCRIPTOR< 104H, 3456H, 12H, 89H, 0>
```

门描述符描述控制转移的入口点。门描述符的格式如下:

Offset	Attributes	Selector	Offset
31~16	15~0	15~0	15~0

其中主要包括一个 16 位的选择子和一个 32 位的偏移量。其属性字段也不再详细介绍。此外,门描述符又可分为任务门、调用门、中断门和陷阱门。调用门描述某个子程序的入口。调用门内的选择子必须指向代码段描述符。任务门指示任务。任务门内的选择子必须指向 GDT 中的任务状态段 TSS 描述符。中断门和陷阱门描述中断/异常处理程序的入口点。中断门和陷阱门内的选择子必须指向代码段描述符。

2.2 全局描述符表 GDT 与局部描述符表 LDT

一个任务会涉及多个段,每个段需要一个描述符来描述,为了便于组织管理,把描述符组织成一个线性表,称为描述符表。有三种描述符表:全局描述符表 GDT、局部描述符表 LDT 和中断描述符表 IDT。全局描述符表含有每一个任务都可能访问的段的描述符,局部描述符表含有该任务自己的一些段描述符。

如下是一个由三个描述符组成的描述符表:

DESCRIPTAB	LABEL	BYTE
DESC1	DESCRIPTOR < 1234H, 5678H, 34H, 92H, 0>	
DESC2	DESCRIPTOR < 5678H, 1234H, 56H, 89H, 0>	
DemoTss	DESCRIPTOR < 104H, 3456H, 12H, 89H, 0>	

2.3 任务状态段 TSS

任务状态段 TSS 是保存一个任务重要信息的特殊段,TSS 中保存任务现场各寄存器状态的完整映象。TSS 在任务切换中起着重要作用,通过它实现任务的挂起和恢复。任务状态段的基本信息由 104 字节组成,此外系统软件还可定义一些附加信息。其具体格式不再详细介绍。

2.4 选择子

在实模式下,逻辑地址空间中存储单元的地址由段地址和段内偏移两部分组成。在保护模式下,逻辑地址空间中存储单元的地址由段选择子和段内偏移两部分组成。

段选择子长 16 位,其格式如下:

15	3	2	1	0
描述符索引		TI	RPL	

其中高 13 位为描述符索引,是描述符在描述符表中的序号;TI=0 表示从全局描述符表中读取描述符,否则从局部描述符表;RPL 为请求特权级。

3 任务切换的途径

利用段间转移指令 JMP 或段间调用指令 CALL,通过任务门或者直接通过任务状态段,可以切换到别的任务。此外,在中断/异常或执行 IRET 指令时,也可发生任务切换。下面先就前者作一详细介绍。

3.1 直接通过任务状态段 TSS 进行任务切换

当段间转移指令 JMP 或段间调用指令 CALL 所含指针的选择子指示一个可用任务状态段 TSS 描述符时(任务状态段描述符描述某个任务状态段,可指向该任务状态段),正常情况下就发生从当前任务到该可用 TSS 对应任务(目标任务)的切换。目标任务的入口点由目标 TSS 内的 CS 和 EIP 字段所规定的指针确定,JMP 或 CALL 指令内的偏移被丢弃。这时 CPU 采用与访问数据段相同的特权级对 TSS 段描述符进行访问,TSS 段描述符的 DPL 规定了访问该描述符的最外层特权级,只有在相同或者更内层级的程序才能访问它。同时,还要求指示它的选择子的 RPL 必须满足 RPL<=DPL 的条件。当这些条件满足时,

就开始任务切换。

如在全局描述符表(首址为 DESC_{TAB})中定义了一个任务的任务状态段描述符为 DemoTss, 见上, 其描述符特权级 DPL=0。该描述符对应的选择子为 DemoTss_SEL:

DemoTss_SEL = DemoTss - DESC_{TAB}

此选择子的 TI 位和 RPL 位均为 0, 表示从 GDT 中读取描述符, 请求特权级为 0。

满足 RPL ≤ DPL 条件。用指令 JUMP16 DemoTss_SEL, 0 即可切换到该任务。其中 JUMP16 是一条宏指令, 完成 16 位偏移的段间转移。其定义如下:

```
JUMP16    MACRO    selector, offsetv
    DB    0EAH      ; 操作码
    DW    offsetv    ; 16 位偏移
    DW    selector    ; 段值/段选择子
ENDM
```

上面化繁就简地给出了直接通过 TSS 实现任务切换的汇编实现, 实际上可能要考虑的细节很多, 如堆栈的切换、参数的传递、特权级的处理等。

3.2 通过任务门进行任务切换

任务门内的选择子指示某个任务的 TSS 描述符。当段间转移指令 JMP 或段间调用指令 CALL 所含指针的选择子指示一个任务门时, 正常情况下就发生任务切换, 也即从当前任务到由任务门内的选择子所指示的 TSS 描述符对应的任务(目标任务), JMP 或 CALL 指令内的偏移被丢弃, 任务门内的偏移也无意义。这时 CPU 仍然采用与访问数据段相同的特权级对任务门进行访问, 任务门的 DPL 规定了访问该描述符的最外层特权级, 只有在相同或者更内层级的程序才能访问它。同时, 还要求指示它的选择子的 RPL 必须满足 RPL ≤ DPL 的条件。当这些条件满足时, 再检查任务门内的选择子, 要求该选择子指示 GDT 中的可用 TSS 描述符。检查完成后, 就开始任务切换。

如一个任务的任务门描述符为 ToTempT, 门内的选择子为 ToTempT_SEL, 当满足上述条件时, 可用指令 JUMP16 ToTempT_SEL, 0 切换到该任务。其中 JUMP16 是上面定义的一条宏指令, 完成 16 位偏移的段间转移。

4 任务切换的过程

下面介绍一下根据指示目标任务 TSS 描述符的选择子进行任务切换的过程, 便于理解执行相应指令实现任务切换时的内部工作过程。

第一, 测试目标任务状态段 TSS 的界限。TSS 在

任务切换过程中起着重要作用, TSS 中保存任务现场各寄存器状态的完整映象, 通过它实现任务的挂起和恢复, 不同的任务 TSS 中有个数不等的信息, 但 TSS 的界限应大于等于 104。

第二, 将寄存器现场保存到当前任务的 TSS 中。把通用寄存器、段寄存器、EIP 及 EFLAGS 的当前值保存到当前 TSS 中, 保存的 EIP 的值是返回地址, 指向引起任务切换指令的下一条指令。但不把 LDTR 和 CR3 的内容保存到 TSS 中。

第三, 把指示目标任务 TSS 的选择子装入 TR。同时把对应 TSS 描述符装入 TR 高速缓冲寄存器中。此后, 当前任务改为原任务, 目标任务改为当前任务。

第四, 基本恢复当前任务(目标任务)的寄存器现场。根据保存在 TSS 中的内容, 恢复各通用寄存器、段寄存器、EFLAGS 及 EIP。在装入段寄存器的过程中, 为了能正确地处理可能发生的异常, 只把对应选择子装入各段寄存器。还装载 CR3 寄存器。

第五, 进行链接处理。如果需要链接, 那么将指向原任务 TSS 的选择子写入当前任务 TSS 的链接字段, 把原任务和当前任务 TSS 描述符类型均改为“忙”, 并将标志寄存器 EFLAGS 中的 NT 位置 1, 表示是嵌套任务。如果需要解链, 那么将原任务 TSS 描述符类型改为“可用”。如果无链接处理, 那么将原任务 TSS 描述符置为“可用”, 把当前任务 TSS 描述符类型改为“忙”。由 JMP 指令引起的任务切换不需要实施链接/解链处理, 由 CALL 指令、中断、IRET 指令引起的任务切换要实施链接/解链处理。

第六, 把 CR0 中的 TS 标志置为 1。这表示已发生过任务切换, 在当前任务使用协处理器指令时, 产生自陷。由自陷处理程序完成有关协处理器现场的保存和恢复。这有利于快速地进行任务切换。

第七, 把 TSS 中的 CS 选择子的 RPL 作为当前任务特权级设置为 CPL。任务切换可以在一个任务的任何特权级发生, 并可切换到另一任务的任何特权级。

第八, 装载 LDTR 寄存器。一个任务可以有自己的 LDT, 也可以没有。当任务没有 LDT 时, TSS 中 LDT 选择子为 0。如果 TSS 中 LDT 选择子非空, 则从 GDT 中读出对应 LDT 描述符, 在经过测试后, 把所读 LDT 描述符装入 LDTR 高速缓冲寄存器。如果 LDT 选择子为空, 则将 LDTR 的存在位置置成 0, 表明任务不使用 LDT。

(下转第 65 页)

置对零、加卸载机构上下快速运动等子模块;

检测控制包括预压、预压检测、动态测试、电液伺服位置闭环控制、虚拟环境测试等子模块;

存储回放及显示打印包括检测、计算数据存储、动态显示、示波器功能显示、历史数据回放、报表及曲线的生成打印等子模块。

在本测试系统中, 关键技术是对加载的控制, 系统采用电液伺服位置闭环控制和推挽激振方式加载。测试过程中, 先通过板簧预压加载确定平衡位置, 然后再推挽激振加载, 即由电液伺服动作器在板簧上施加交变的载荷, 并作上下运动, 可测得脉动的位移量。可按国家规定的 QCn29035-91:《汽车钢板弹簧技术条件》中关于板簧总成预加变形量脉动幅值的计算公式计算。计算公式为:

$$y_m = 0.6(y_j + \beta y_j^{1/2}); y_{am} = 0.4(y_j + \beta y_j^{1/2})$$

式中 y_j 为车上夹紧状态弹簧满载变形量 (cm); y_{am} 为动荷脉动幅值 (cm); y_m 为预加变形量 (cm); β 为弹簧种类系数, 后总成 $\beta = 2$, 前总成 $\beta = 2.5$ 。

为了保证板簧在测试中动态位移的可控性, 采用了以位置作为反馈量的电液伺服位置闭环控制, 电液伺服位置闭环控制系统框图如图 4 所示。

由于板簧测试时位移很大, 而激振频率因受动力机构输出特性的限制而较低, 因此, 电液伺服阀、位移传感器均可视为比例环节, 伺服放大器在采用带电流负反馈的放大器时亦可忽略线圈动态影响而视为比例环节。则其闭环回路的开环传递函数可等效为^[3]:

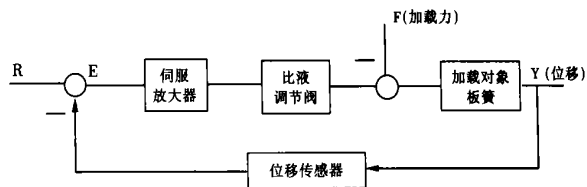


图 4 电液伺服位置闭环控制方框图

$G(S) = k_0 / \{ (S \omega_r + 1) (S^2 \omega_0^2 + 2\xi_0 S \omega_0 + 1) \}$; k_0 为开环放大倍数。

因此电液伺服位置闭环控制系统按一个“O”型的三阶系统设计。

3 结束语

基于虚拟仪器的汽车板簧载荷测试系统采用了仪器技术设计, 用户界面友好, 操作方便, 功能齐全。特别是在预压后采用推挽激振方式, 使加载、卸载刚性得到互补。而位置闭环控制又使位移波形线性对称。具有良好的测试性能。

参考文献:

- [1] National Instruments. LabWindows/CVI Programmer Reference Manual[M]. National Instruments, 1996.
- [2] National Instruments. LabWindows/CVI User Manual[M]. National Instruments, 1996.
- [3] 李洪人. 液压控制系统[M]. 北京: 国防工业出版社, 1990.

(上接第 62 页)

第九, 装载代码段寄存器 CS、堆栈段寄存器 SS 和各数据段寄存器及其它的高速缓冲寄存器。

第十, 把调试寄存器 DR7 中的局部启用位设置为 0, 以清除局部于原任务的各个断点和方式。

上面介绍了用段间转移指令 JMP、段间调用指令 CALL 实现的任务切换。下面再对中断/异常引起的任务切换作一简单介绍。

中断/异常引起的转移通过两种方式实现, 一是通过中断门或陷阱门实现转移。这时控制转移到当前任务内的一个处理程序过程, 与通过调用门的 CALL 指令一样, 从中断门或陷阱门中获取指向处理程序的 48 位全指针, 其中, 16 位选择子是对应处理程序代码段的选择子, 它指示 GDT 或 LDT 中的描述符, 32 位偏移指示处理程序入口点在代码段内的偏移。另一种方式是通过任务门的转移。这时控制转

移到一个作为独立的任务方式出现的处理程序。这才是一种任务的切换。任务门中含有指向处理程序的 48 位全指针, 其中, 16 位选择子是对应处理程序任务的 TSS 段的选择子。

段间转移指令 JMP、段间调用指令 CALL、软中断指令 INT 和中断返回指令 IRET 引起的任务切换是主动的任务切换。中断和异常(不包括软中断指令)引起的任务切换是被动的任务切换。任务切换时特权级可能发生变换, 这取决于目标任务。

参考文献:

- [1] 杨季文. 80X86 汇编语言程序设计教程[M]. 北京: 清华大学出版社, 1998.
- [2] [美] 布雷. 80X86、奔腾机汇编语言程序设计[M]. 金惠华译. 北京: 电子工业出版社, 1998.