1 中断和异常处理

1.1 异常和异常处理

处理器为了实现处理异常和中断，使用了一个数据结构，也就是中断描述符表，用于存放中断描述符。同时处理器为每个异常和中断条件都赋予了一个向量，用于作为中断描述符表的索引号。

中断可以从硬件和软件产生。外部中断通过 INTR 和 NMI 接收。NMI 接收的中断是不可屏蔽中断，其向量号为 2。INTR 接收的外部中断可屏蔽，通过设置 EFLAGS 中的 IF 位为 0 来屏蔽这些中断。这里的 IF 标志可以通过 STI 和 CLI 来设置或清零。只有当程序的 CPL(程序特权级) 小于 IOPL(I/O 特权级字段) 时，才可执行这两条指令。还有几种情况可以影响 IF 标志，比如 PUSHF 指令、IRET 指令等。当通过中断门处理一个中断时，IF 会被自动清零。

软件中断主要借助 INT 指令，在指令操作数中提供中断向量号。向量号 0 到 255 都可以作为 INT 指令的中断号，比如指令 INT 0x80 可以执行系统中断。EFLAGS 中的 IF 标志无法屏蔽软件中断。

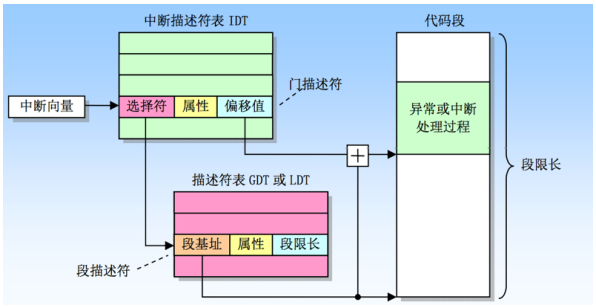
1.2 IDT 和 IDT 描述符

中断描述符表类似于全局描述符表，用于存放门描述符。处理器使用 IDTR 寄存器定位 IDT 表的位置，IDTR 有 48 位，高 32 位是 IDT 表的基地址，而低 16 位为 IDT 的长度值。使用 LIDT 指令可以将内存中的基地址和限长值加载到 IDTR 寄存器中，不过该指令只能由 CPL 为 0 的代码执行。

IDT 存放的门描述符有三类，为中断门描述符、陷阱门描述符和任务门描述符，它们都是 8 字节的。中断门描述符和陷阱门描述符中存放了段选择符和偏移值，用于对段的寻址，从而将程序执行权转移到代码段中异常或中断的处理过程中。而任务门描述符中含有段选择符，不过没有偏移值，因为在门中的偏移值没有意义。

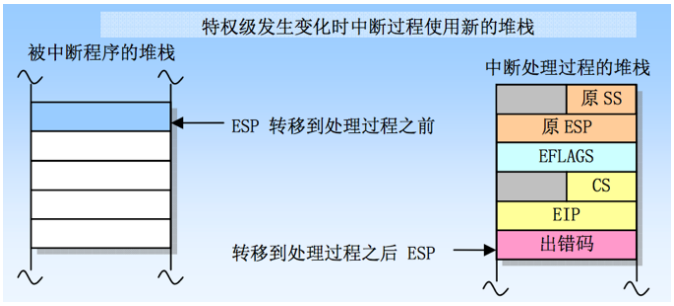
1.3 异常与中断处理

发生异常或中断时，处理器使用异常或中断的向量作为 IDT 表的索引，从而得到相应的门描述符。门描述符中的段选择符指向 GDT 或 IDT 中的段描述符，而段描述符给出相应代码段的段基址。门描述符中的偏移值作为段基址的偏移，从而指向执行异常或中断处理过程的代码段。我觉得下面这张图描述得很形象。



异常或中断处理分为两种情况。一种是在高特权级下执行，一种是在同一特权级下执行。

处理过程在高特权级上执行时，将会发生堆栈切换操作。此时处理器首先获得新栈的段选择符 SS 和栈指针 ESP，这里的段选择符和栈指针由当前任务的 TSS 段提供。然后把原栈选择符和栈指针压入新栈。随后将 EFLAGS、CS 和 EIP 当前值压入新栈。如果异常会产生一个错误号，该错误号也将被压入新栈。如下图所示。



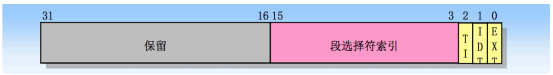
处理过程在同一特权级上执行时，过程相对简单。处理器将 EFLAGS、CS 和 EIP 当前值压入堆栈。如果异常会产生一个错误号，该错误号也会被压入堆栈。这个过程如下图所示。

中断处理过程结束时，程序使用 IRET 指令从中断处理过程中返回。此时处理器会从堆栈中弹出代码段的选择符 CS 和指令指针 EIP。同时 IRET 会把保存的寄存器内容恢复到 EFLAGS 中。在特权级保护机制下，只有当 CPL 为 0 时，IOPL 字段才会被恢复。只有 CPL 小等于 IOPL 时，IF 标志才会被改变。如果中断处理过程中发生了堆栈切换，那么 IRET 指令会切换回原来的堆栈。

需要注意的是，通过中断门访问异常或中断处理过程时，处理器会清零 IF 标志，随后再使用 IRET 指令恢复 IF 标志。而通过陷阱门访问处理过程则不会影响 IF 标志。

1.4 错误码

错误码类似于段选择符，用于寻址段。不过这里的段是与特定的异常条件有关的。段错误码的格式如下所示。



图中的低三位为 TI、IDT 和 EXT。EXT 为 0 时，表示执行程序以外的事件造成了异常。IDT 为 0 时，错误码的索引指向 GDT 或 LDT 的段描述符；当 IDT 为 1 时，错误码的索引指向 IDT 的一个门描述符。只有当 TI 为 0 时 TI 标志才有用。因为 TI 标志用于选择 GDT 表和 LDT 表。当 TI 为 1 时，错误码的索引部分指向 LDT 的段描述符；当 TI 为 0 时，错误码的索引部分指向 GDT 表中的描述符。

错误码有一个特例，也就是页故障异常的错误码。页故障错误码中没有段选择符索引，只有最低 3 位比特位有效，分别为 P、W/R 和 U/S。P=0 时，表示也不存在；P 为 1 时，表示违反页级保护权限。W/R=0 时，表示读操作引起了异常；W/R=1 时，表示异常由写操作引起。U/S=0 时，表示异常发生时 CPU 正在执行超级用户代码；U/S=1 时，表示异常发生时 CPU 正在执行一般用户代码。在第一次学习报告有提到过，CR2 控制寄存器中存放着引起页面故障异常的线性地址。

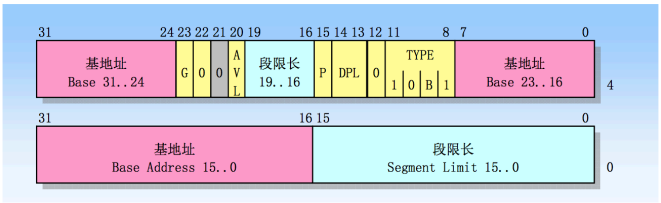
2 任务管理

2.1 用于任务管理的数据结构

为了进行任务管理，处理器定义了一些寄存器和数据结构，分别为任务状态段 TSS、TSS 描述符、任务寄存器 TR 和任务门描述符。

用于恢复一个任务执行的处理器状态信息被保存在 TSS 中。TSS 分为两类字段，一个是动态字段，还有一个是静态字段。当任务切换而被挂起时，处理器会更新动态字段的内容。而静态字段通常不会改变，它的字段内容在任务被创建时设置。

TSS 由任务状态段描述符来寻址和定义，以下是 TSS 段描述符的格式。



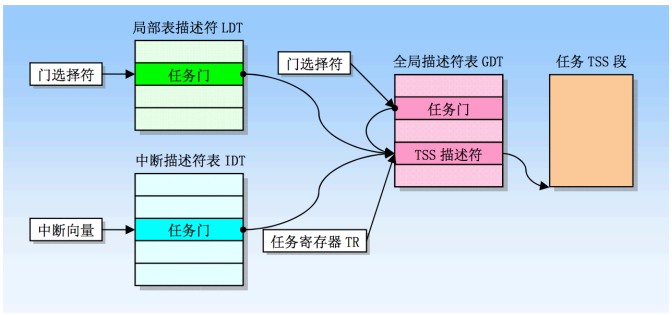
描述符中的 TYPE 字段中的 B 标志用于表示任务是否处于忙状态。B=1 时，表示任

务正忙；B=0 时，表示任务处于非活动状态。描述符中的 G 标志是颗粒度，当 G=0 时，TSS 段的长度必须大等于 104 字节。描述符中的 DPL 标志用于特权级保护机制中。当发生任务切换时，访问 TSS 的程序的 CPL 必须小等于 TSS 中的 DPL。描述符中的段基址就是任务状态段的基址。

任务寄存器存放着段选择符和当前 TSS 段描述符。LTR 指令可以在系统初始化阶段给 TR 寄存器加载初值，之后 TR 的内容会在任务切换时自动地被改变。

除了使用段选择符直接访问 GDT 中的 TSS 描述符，还可以通过任务门描述符间接地访问 TSS 描述符，因为任务门描述符含有 TSS 选择符字段。任务门描述符中的 DPL 用于支持特权级保护机制。当程序通过任务门描述符调用程序时，程序的 CPL 以及指向任务门描述符的门选择符的 RPL 都必须小等于任务门描述符中的 DPL。

下图描述了调用任务的两种方式，一种是通过任务门描述符访问，一种是通过 TSS 段描述符访问。从图中可以看出，任务门描述符可以存放在 GDT、LDT 或 IDT 表中，程序通过任务门描述符间接访问到 TSS 描述符。



2.2 任务切换

处理器可以通过 4 种方式进行任务切换操作，分别是：1. 对 TSS 描述符执行 JMP 或 CALL 指令。2. 对任务门描述符执行 JMP 或 CALL 指令。3. 通过中断或异常向量指向任务门描述符。4. 执行 IRET 指令。

以下是进行任务切换的过程：

- 首先获得新任务的 TSS 段选择符。段选择符的获取有三种途径：1. 从 JMP 或 CALL 指令操作数中获取。2. 中断向量索引到 IDT 表中的任务门描述符，获取其中的 TSS 选择符。3. 执行 IRET 指令时，从当前 TSS 的前一任务链接字段中获取。

  这里说一下前一任务链接字段，它需要和 EFLAGS 中的 NT 标志配合使用。NT 标志为 1 时，表明当前任务嵌套在另一个任务中执行，并且当前任务的 TSS 段的前一任务链接字段中存放着高一层任务的 TSS 选择符。

- 经过特权级保护机制的检查。使用 JMP 或 CALL 调用程序时，当前任务的 CPL 和新任务的 TSS 段选择符的 RPL 必须小等于新任务 TSS 段描述符的 DPL。发生中断、异常或使用 IRET 指令时，则无视特权级保护机制。

- 对 B 标志和 NT 标志的设置。如果使用 JMP 进行任务切换，则将 B 标志清零。如果使用 IRET 指令进行任务切换，则将 B 标志和 NT 标志清零。

- 将当前任务状态保存到当前任务的 TSS 中，包括所有通用寄存器的值，段寄存器中的段选择符，标志寄存器 EFLAGS 以及指令指针 EIP。
- 加载新的 TSS 段描述符。如果任务切换由 CALL、JMP、异常或者中断产生，则对新 TSS 段描述符中的 B 标志置 1。
- 将新 TSS 的段选择符和描述符加载到任务寄存器中。同时设置 CR0 寄存器的任务已切换标志 TS 为 1。
- 将新 TSS 状态加载进处理器，包括所有通用寄存器、段选择符、标志寄存器 EFLAGS、LDTR 寄存器、CR3 寄存器以及 EIP。如果任务切换由 CALL、JMP、异常或者中断产生，则将 EFLAGS 中的 NT 位置一。
- 开始执行新任务。

2.3 任务地址空间

任务的地址空间由任务能够访问的段构成，这些段有代码段、数据段、堆栈段、TSS 中引用的系统段以及任务代码能够访问的任何其他段。

在任务之间共享数据有以下三个途径:

- 通过 GDT 共享数据。这个方法是很明显而且简单的。不同的段可以映射到相同的物理地址空间中，而每个段又有对应的段描述符。这些段描述符存放在 GDT 表中。于是任务通过 GDT 共享相同的物理地址空间。这种方法的缺点是所有任务都可以共享这些段中的代码和数据。
- 让任务共享相同的 LDT。让一些特定的任务的 TSS 中 LDT 字段指向同一个 LDT，从而访问到相同的物理地址空间。
- 让不同 LDT 中的某些段描述符映射到相同的物理地址。这样的段描述符通常被称为别名段。