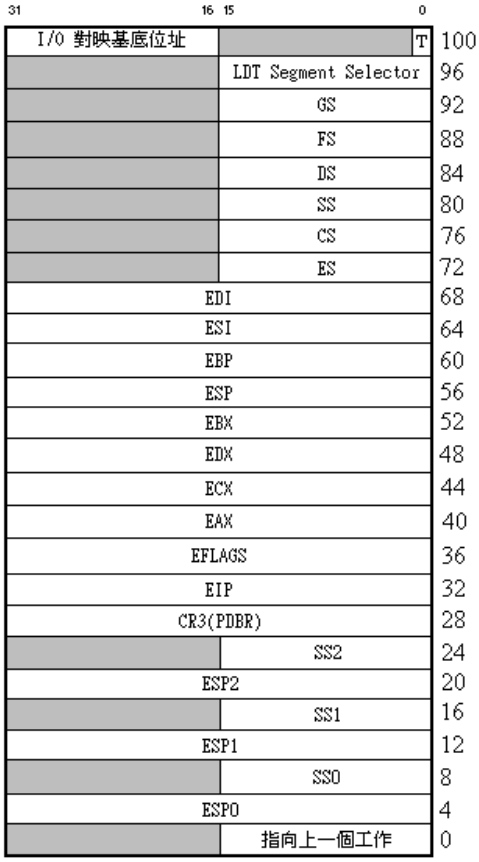
工作狀態表

工作狀態分段（TSS）

TSS 是儲存一個工作的狀態的系統 segment。它和一般的 segment 一樣，也有 segment descriptor。不過，它的 segment descriptor 只能存放在 GDT 中。如果一個 segment selector 的 TI 是 1（代表對 LDT 存取），而想利用它來存取 TSS 的話，會導致 general-protection（#GP）例外。同時，如果想把指向 TSS 的 segment selector 載入到分段暫存器中的話，也會導致 general-protection（#GP）例外。
TSS 的 segment descriptor 格式如下：

TSS Descriptor

這個 segment descriptor 和一般的 segment descriptor 很類似（參考「記憶體管理」中的「分段架構」），只有在型態的地方，標示出 TSS descriptor。在型態位元中，因為 1001B 是表示 inactive 的工作，而 1011B 是表示 active 的工作，因此，上圖的 B 位元可視為工作的「忙碌」位元。如果 B 為 0，則表示工作是 inactive，反之則是 active。
因為工作是不可遞迴的，所以處理器利用 B 位元來判斷工作是否被中斷。為了避免可能的問題，系統必須確定所有的 TSS 都只有一個 TSS descriptor 指向它們。
此外，一個 32bit 的 TSS 的大小至少是 68H 個位元組，所以在 TSS descriptor 中的 Limit 至少必須是 67H（68H - 1）。如果 Limit 小於 67H，則在載入 TSS 時，會發生 invalid-TSS（#TS）例外。如果在 TSS 中有 I/O 對映表，則 TSS 還要再更大。因為處理器並不限制 TSS 的大小，所以作業系統可以在 TSS 中存放任何相關的資料。
在 TSS descriptor 中的 DPL 欄位表示 TSS 的特權等級。只有在 CPL 小於或等於 TSS 的 DPL 的程序，才能執行這個工作。通常在系統中，會把所有的 TSS 的權限設為 2 或更小，以確保只有系統可以執行這些工作。不過，在某些特別的情形中，也可能會想讓一般的應用程式能執行某些工作，這時就可以把這些 TSS 的 DPL 設為 3。
TSS 的內容分為兩大類：動態欄位和靜態欄位。在工作切換時，處理器會更新被暫停執行的工作的 TSS 中的動態欄位。而處理器一般不會更動靜態欄位的內容。TSS 的格式如下：



Task-State Segment

在上圖中，灰色部分是保留位元，應設為 0。其中，屬於動態欄位的有：

- 通用暫存器，包括 EAX、ECX、EDX、EBX、ESP、EBP、ESI、和 EDI。
- 分段暫存器，包括 ES、CS、SS、DS、FS、GS。
- EFLAGS 暫存器。
- EIP 暫存器。
- 指向上一個工作的連結。當這個工作是被另一個工作以 CALL、中斷、或例外的形式呼叫時，這個欄位會指向呼叫者的工作。這樣，在返回時才能返回到正確的工作中。

下面列出的欄位是靜態欄位，這些欄位在工作建立時就設定了，一般情形下不會改變：

- LDT Segment Selector。
- CR3 暫存器。
- 特權堆疊指標（SS0:ESP0、SS1:ESP1、和 SS2:ESP2，分別是 ring 0、ring 1、ring 2 所用的堆疊）。
- T 旗標，是除錯時用的。如果 T 為 1，則在切換到這個工作時，會產生 debug 例外。
- I/O 對映基底位址。這個位址是一個 16 bit 的偏移量，指向 TSS 中「I/O 對映表（I/O permission bit map）」和「中斷導向表（interrupt redirection bitmap）」的基底位址。中斷導向表是用在 Virtual-86 模式中，在本文中不討論。

如果開啟了分頁功能，則要注意必須讓 TSS 的前 104 bytes 在同一個分頁中。否則，在工作切換時可能會發生錯誤。此外，目前的 TSS、上一個 TSS、和兩者在 descriptor table 中的 entry 都必須是可任意讀寫的（不可以是唯讀）。

工作暫存器

工作暫存器（TR）存放 TSS descriptor 的 segment selector，和整個 TSS descriptor。其中，segment selector 的部分是可見部分（visible part）。在把 segment selector 載入到 TR 中時，處理器會把 GDT 中整個 TSS descriptor 都載入到 TR 中。這樣，在需要 TSS descriptor 的資料時，就不用在到記憶體中存取了。
利用 LTR 指令，可以從 GDT 中載入一個 TSS descriptor 到 TR 中。這是一個特權指令，所以只有 CPL 為 0 的程式可以使用。而 STR 指令可以把 TR 中的資料寫到記憶體或通用暫存器中。這個指令可以用來判斷目前執行的工作，而且可以在任何特權等級下使用。不

過，一般情形下，還是只有系統程式會使用到這個指令。

Task gate

Task gate 指向一個 TSS，可以放在 GDT、LDT、或 IDT 中。Task gate 的格式可以參考「中斷 / 例外處理」的「中斷描述表」）。在 task-gate 中的 segment selector 欄位，是指向 TSS 的。它的 RPL 欄位會被忽略。在 task-gate 中的 DPL 則是控制 task-gate 的存取權限。在經由 task-gate 執行工作時，呼叫者的 CPL 和 segment selector 中的 RPL 必須等於或小於 task-gate 的 DPL，才能呼叫成功。