https://play.google.com/store/apps/details?id=jp.co.namura.magnifier&hl=zh_HK
https://www.pngfind.com/mpng/iRhTJTh_tux-linux-logo-start-menu-linux-icons-hd/

作業一：核心進入點

中正大學 作業系統實驗室
指導教授：羅習五

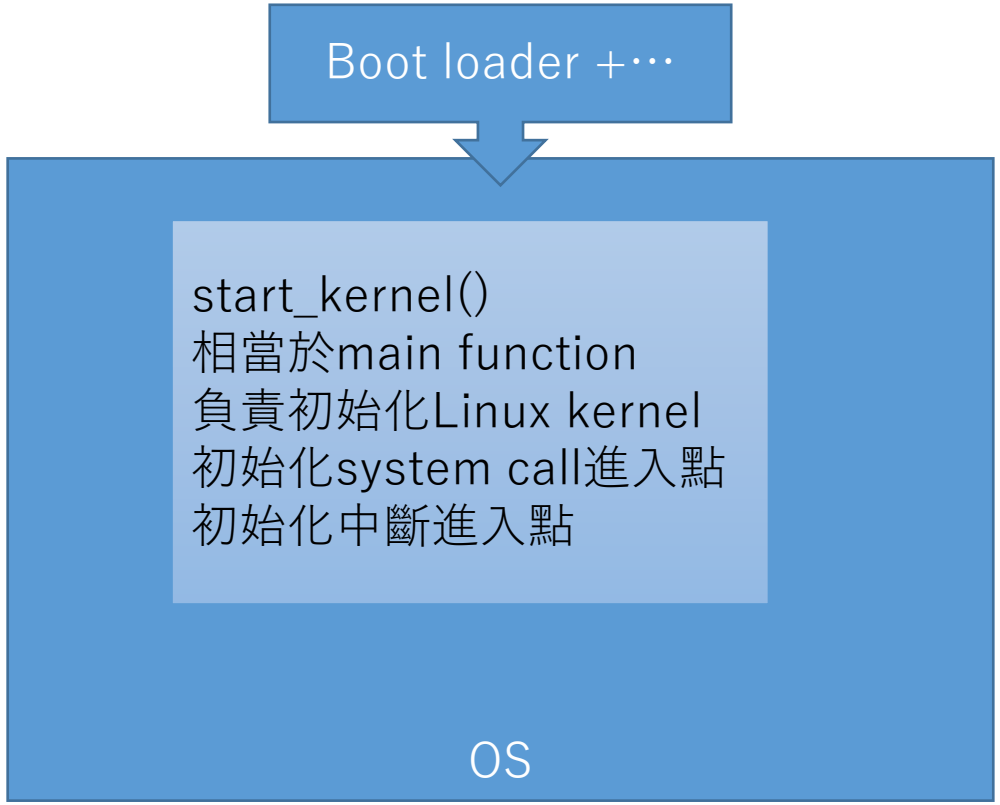
圖片來源

- 新垣結衣
  - https://makey.asia/column.php?id=532
  - http://pic.haibao.com/image/14284778.html?kw=%E6%96%B0%E5%9E%A3%E7%BB%93%E8%A1%A3
  - https://huaban.com/pins/835412722/

作業目標及負責助教

- 作業目標：
  - 了解如何用QEMU及gdb+Eclipse對Linux kernel除錯
  - 藉由核心進入點，了解探索Linux kernel的技巧
- 負責助教：
  - 請看網頁

核心進入點



核心進入點

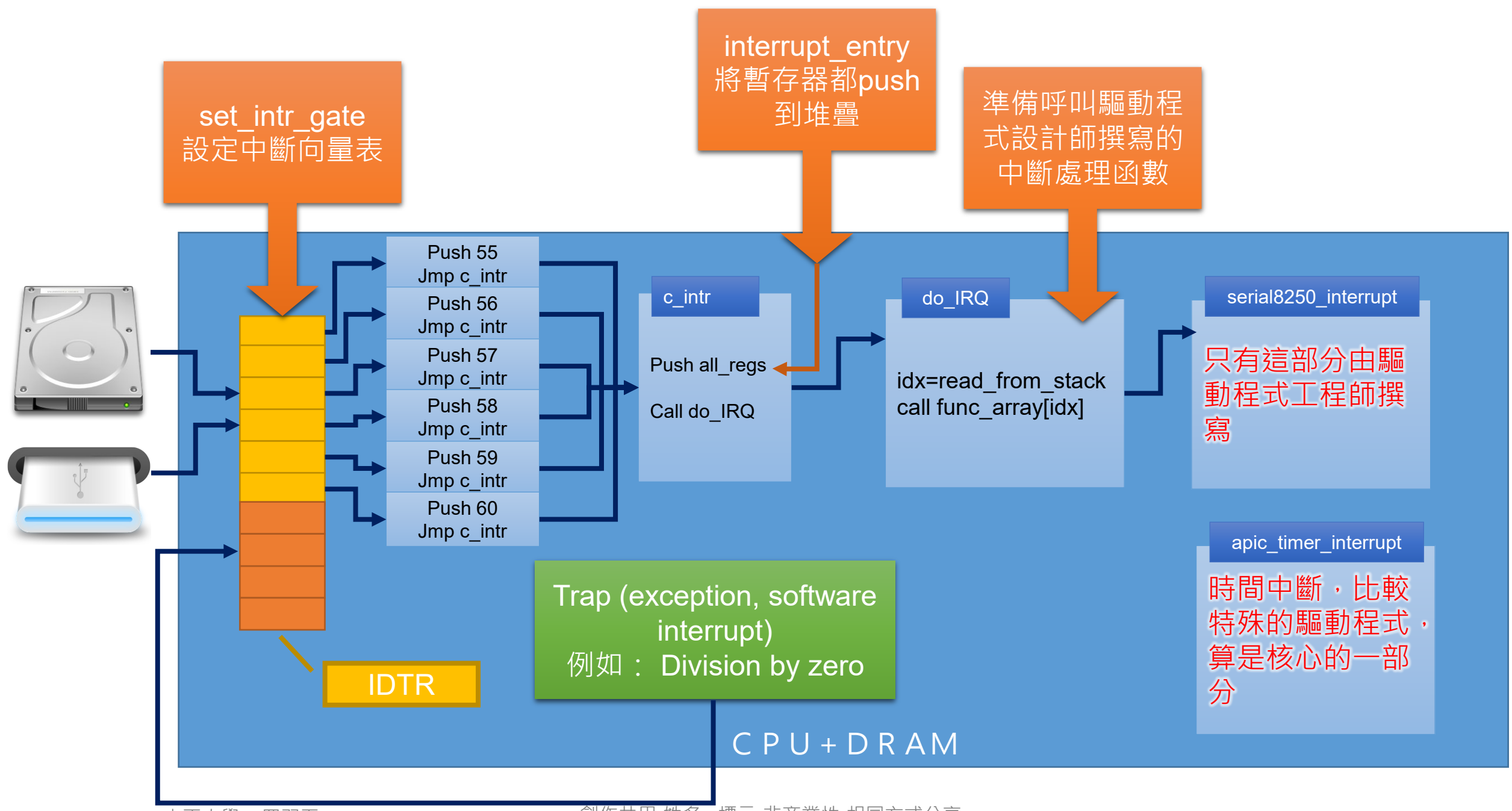
WRMSR — Write to Model Specific Register
通常使用頻率較少或者只適用某些處理器的暫存器，都必須用WRMSR
https://www.felixcloutier.com/x86/wrmsr

Boot loader +…

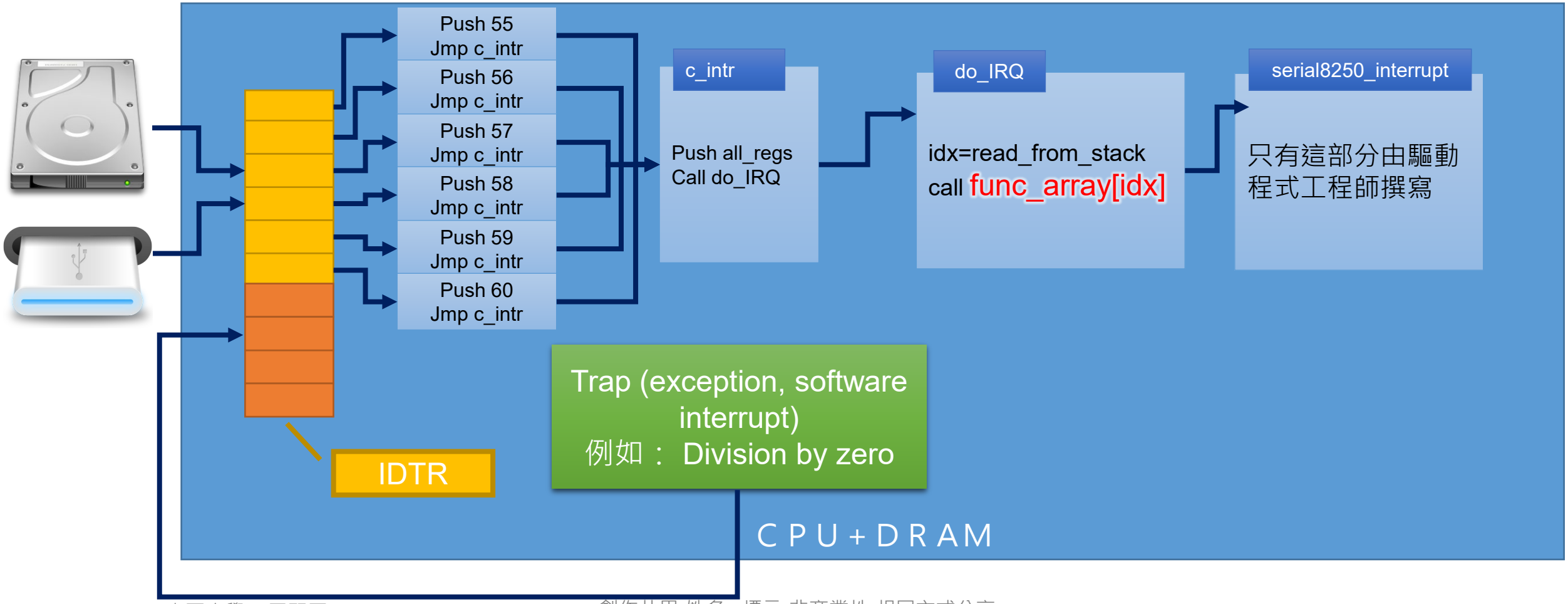
start_kernel()
相當於main function
負責初始化Linux kernel
初始化system call進入點
初始化中斷進入點

syscall_init
使用WRMSR將 entry_SYSCALL_64
寫入CPU內部的暫存器
往後只要執行asm(“syscall”)就會跳到
entry_SYSCALL_64

OS



從OS開發者看「註冊中斷」



關於堆疊

- 不管是system call、trap、interrupt，在呼叫第一個C函數時，堆疊一定是長這樣子

```
struct pt_regs {
/*
 * C ABI says these regs are callee-preserved. They aren't saved on kernel entry
 * unless syscall needs a complete, fully filled "struct pt_regs".
 */
        unsigned long r15;
        unsigned long r14;
        unsigned long r13;
        unsigned long r12;
        unsigned long rbp;
        unsigned long rbx;
/* These regs are callee-clobbered. Always saved on kernel entry. */
        unsigned long r11;
        unsigned long r10;
        unsigned long r9;
        unsigned long r8;
        unsigned long rax;
        unsigned long rcx;
        unsigned long rdx;
        unsigned long rsi;
        unsigned long rdi;
/*
 * On syscall entry, this is syscall#. On CPU exception, this is error code.
 * On hw interrupt, it's IRQ number:
 */
        unsigned long orig_rax;
/* Return frame for iretq */
        unsigned long rip;
        unsigned long cs;
        unsigned long eflags;
        unsigned long rsp;
        unsigned long ss;
/* top of stack page */
};
```

作業系統概論基於GNU/Linux

中正大學，資工系，作業系統實驗室，副教授 羅習五，shiwulo@gmail.com

Intel的系統暫存器

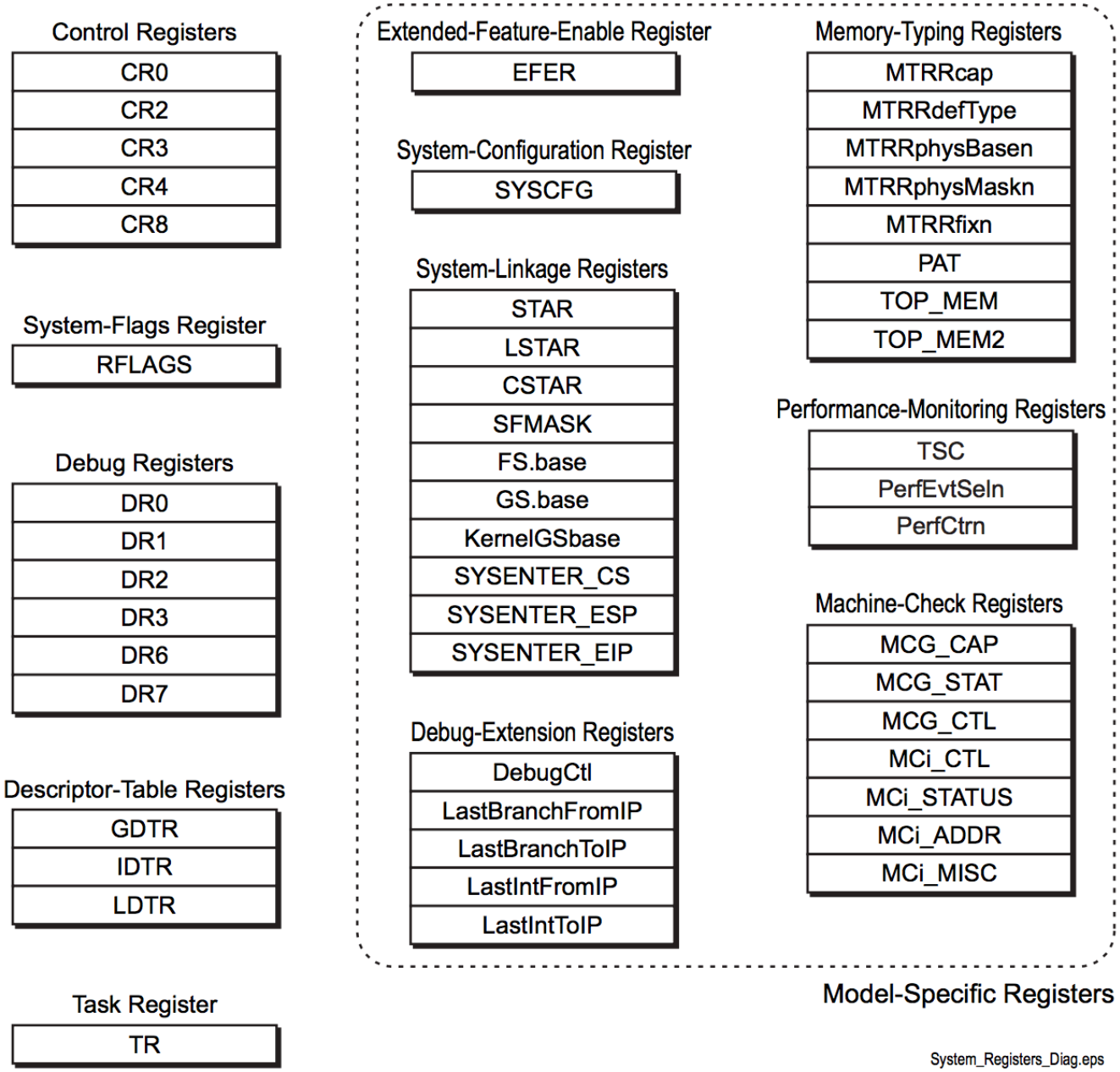


Figure 1-7. System Registers

- 大概介紹86的語法
  - https://software.intel.com/content/www/us/en/develop/articles/introduction-to-x64-assembly.html?wapkw=