作業八：從使用者模式追蹤到核心模式

學習目標：

1. 了解作業系統對上層軟體提供的進入點（錯誤、system call、breakpoint）
2. 與作業三、四合併在一起，進一步的了解 system call 從 user space 到 kernel space 如何運作。

題目：

- 以下步驟請對應作業三、作業四一起閱讀、了解
- 請撰寫或者使用 dropbox 內的程式碼，程式碼的名稱必須為 test_syscall。程式碼中必須使用組合語言呼叫 system call，例如：write
- 將上述程式碼放到 sharedFolder 中，並使用 gcc 編譯，編譯出來的程式必須能在『Linux in QEMU』中正確執行
- 在『dbg-Linux5.0-in-QEMU.sh』中執行你的程式碼，於 Eclipse 中進行追蹤
- （問題一）設定中斷點在 test_syscall 發出 system call 之前，請在這個地方截圖
- （問題二）使用單步追蹤（si），直到 Linux kernel，請在進入 Linux kernel 時截圖

- （問題三）請說明 Linux kernel 如何用 RAX 暫存器判斷要呼叫哪個 Linux 內部的函數
- （問題四）請大致說明作業系統如何處理 write。

作業繳交：

1. 學號、姓名（請隱藏個人資訊，例如：學號 687410007，姓名：羅 X 五）
2. 文件

   甲、將問題一 ~ 四直接寫成 pdf 文件

繳交：

1. 繳交期限：請參考網頁
2. 如果真的不會寫，記得去請教朋友。在你的報告上寫你請教了誰即可。