作業七：追蹤System call在Linux kernel的進入點

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：

在作業三中，我們使用組合語言發出system call，在這個作業則需要你追蹤system call到底是怎樣呼叫核心內部的函數，並且了解breakpoint、divide by zero的實現方式與system call非常的相像。

1. （35 pt）請將中斷點設定在start\_kernel裡面的trap\_init()，請解釋下面三行程式的大概意義，例如：第一行程式碼中的X86\_TRAP\_DE是divide by zero這個exception的編號，而divide\_error是發生這個exception時，CPU會執行divide\_error這個位置的程式碼。

請解釋剩餘二行程式碼。

* 1. set\_intr\_gate(X86\_TRAP\_DE, divide\_error);
  2. set\_system\_trap\_gate(SYSCALL\_VECTOR, & system\_call);
  3. set\_nmi\_gate(X86\_TRAP\_BP, &int3);

1. （35 pt）請撰寫三隻程式分別觸發上述三行程式所設定的程式碼。（提示：可以在系統開機完成以後，要執行你的程式碼之前，於gdb中按下control+c設定中斷點後再開始執行程式）
2. （30 pt）請約略的敘述一下system\_call這段程式碼中，如何使用RAX暫存器，獲得system call內部的實現函數的位置。

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
   1. 依照問題2的順序，執行檔名稱分別為：div、syscall、int3。必須附上source code及makefile。
   2. 將問題1~3直接寫成pdf文件

繳交：

1. 繳交期限：2020/11/25（星期三）晚上11:59:59
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。

資源：

可使用下列主機，帳號為：s學號（如：s687410007），密碼為：ecourse上的email（如：shiwulo@gmail.com）

osdi[1~5].cs.ccu.edu.tw，例如：ssh osdi1.cs.ccu.edu.tw