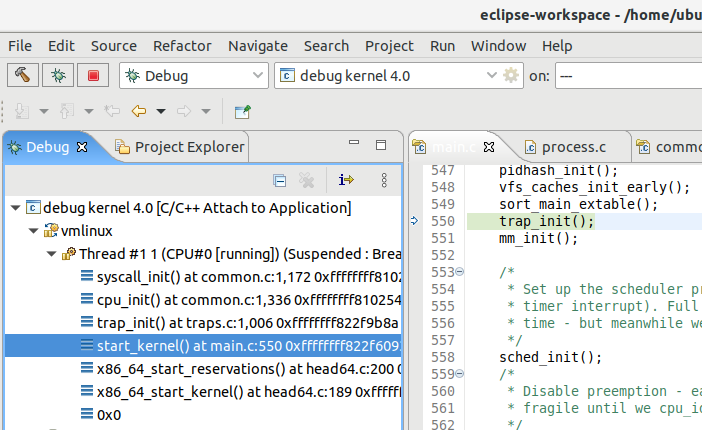
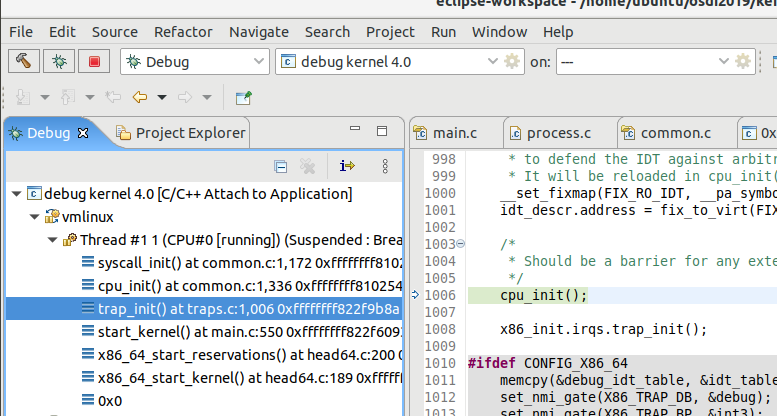
hw04.system call entry point 406410114 郭晏誠

1. 當應用程式發出system call時，Linux會從哪裡開始執行

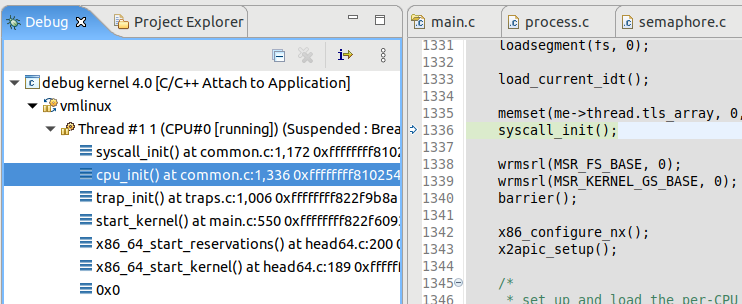
(i)start\_kernel()呼叫trap\_init()



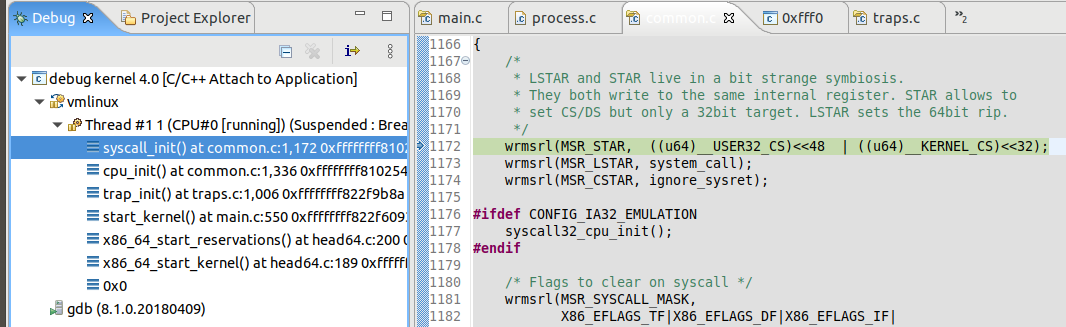
(ii) trap\_init()呼叫cpu\_inti()



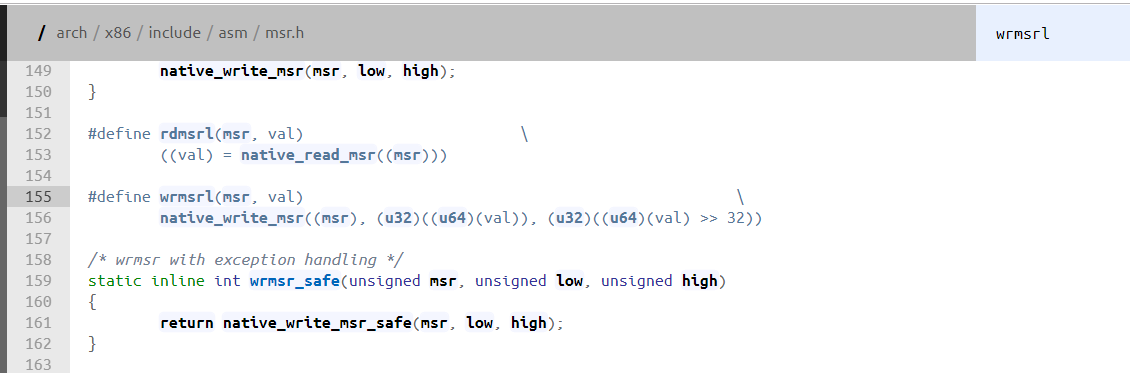
(iii)在cpu\_init()呼叫syscall\_init()



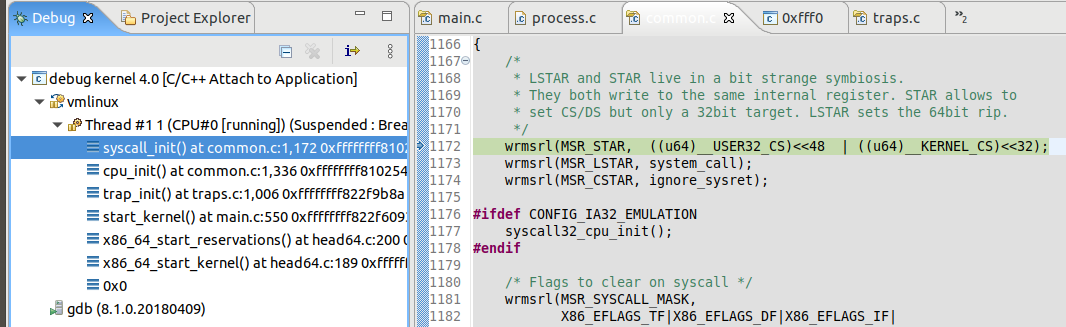
(iv) syscall\_init()用wrmsrl(MSR\_STAR, ((u64)\_\_USER32\_CS)<<48 | ((u64)\_\_KERNEL\_CS)<<32);



Google結果是msr 是machine specific register ##不是每個CPU都有

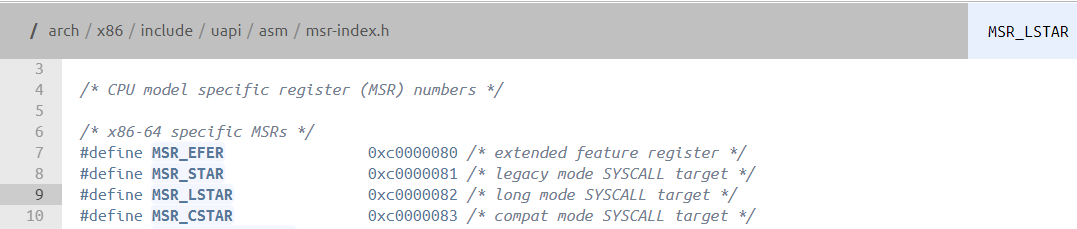


(v)wrmsrl(MSR\_LSTAR, system\_call);// 將system\_call 放到MSR\_LSTAR

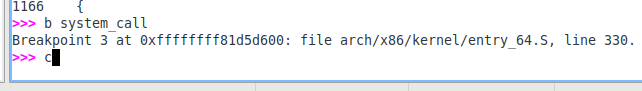


Google發現進入點在#define MSR\_LSTAR 0xc0000082 /\* long mode SYSCALL target \*/

Long mode 64位元 ；compat mode為32位元

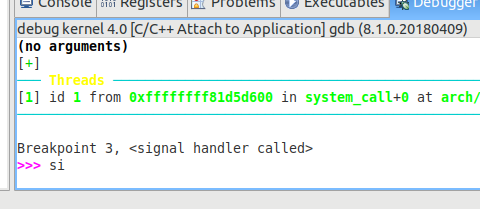


(vi)b system\_call

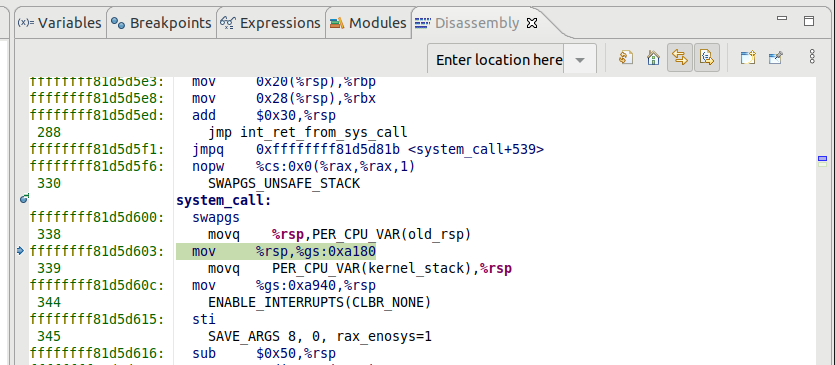


1. Linux kernel如何使用rax暫存器呼叫對應的system call處理函數？

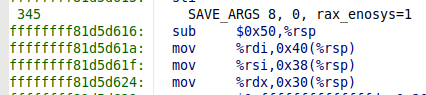
(i)輸入si 看組合語言



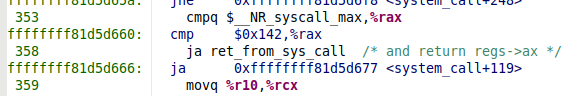
(ii)看到右邊有330 system\_call



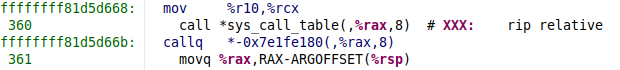
(iii)把rdi rsi rdx放到stack



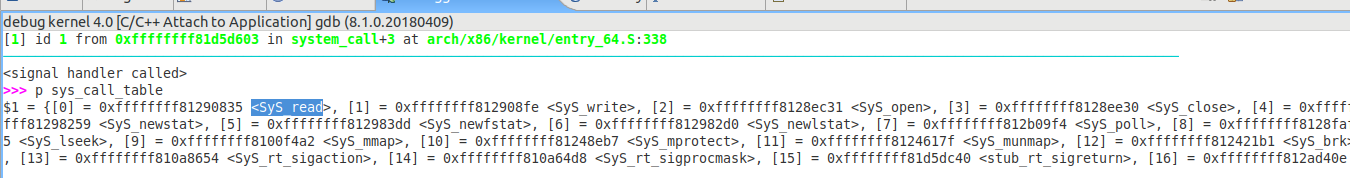
(iiii)檢查rax是否超過355



(iv)看sys\_call\_table



(v)p sys\_call\_table # 看sys\_call\_table



(vi) info registers eax #印出rax 63號sys\_newuname 切換使用者模式





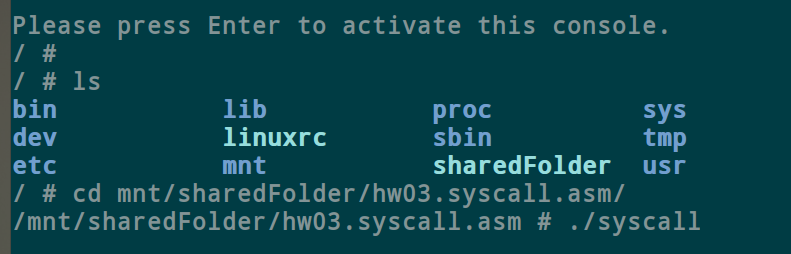
3.去看HW3 syscall一開始是呼叫幾號syscall並且印出來

(i)先在eclipse輸入file /home/ubuntu/vmlinux 按下c

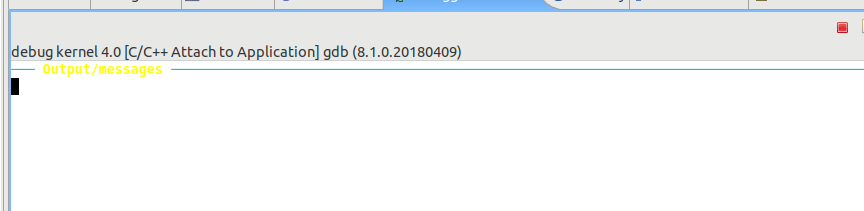
一張含有 螢幕擷取畫面 的圖片

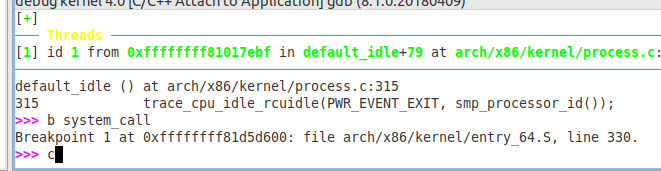
自動產生的描述

(ii)dgb\_qemu.sh會開始跑到可以打字的位置，先準備好要跑的程式 但是先不要按下enter



(iii)回到eclipse按下ctrl+c 並輸入要中斷的位置 ( b system\_call )再輸入c





(iv)回到cmd按enter會再跳回eclipse

(v)輸入c按下enter 再利用info registers eax 檢查rax看是哪一號syscall

(vi) 連續操作三次可得下圖

第一次rax為0是sys\_read (讀取輸入的指令)

第二次rax為1是sys\_write(將指令寫到其他位置，例如history)

第三次rax為2是sys\_open (利用sys\_open開啟檔案)

故執行後的第一個syscall是sys\_open

