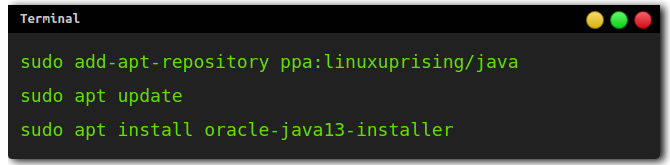
安裝Linux\_Kernel\_除錯(Ubuntu 18.04)

1. 安裝Oracle java
2. 安裝Eclipse
3. 切換gcc
4. 安裝kernel需要的工具
5. 下載Kernel
6. 確認檔案
7. 編譯Linux kernel前設定
8. 編譯Linux kernel
9. 複製檔案與執行qemu
10. 安裝Oracle java

sudo add-apt-repository ppa:linuxuprising/java

sudo apt update

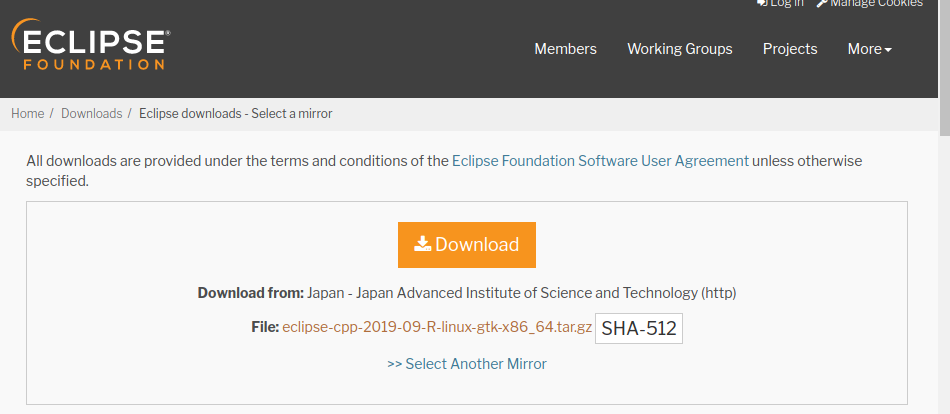
sudo apt install oracle-java13-installer #一直按下讀完說明 按右選OK #安裝過程中間有選項都按Y



1. 安裝Eclipse

(1)選擇Eclipse( Ubuntu 64bit C++平台)

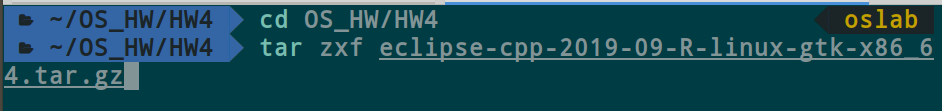
<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2019-09/R/eclipse-cpp-2019-09-R-linux-gtk-x86_64.tar.gz>



(2)解壓縮

$ cd OS\_HW/HW4 #cd到指定目錄

$ tar zxf eclipse-cpp-2019-09-R-linux-gtk-x86\_64.tar.gz #把檔案解壓縮

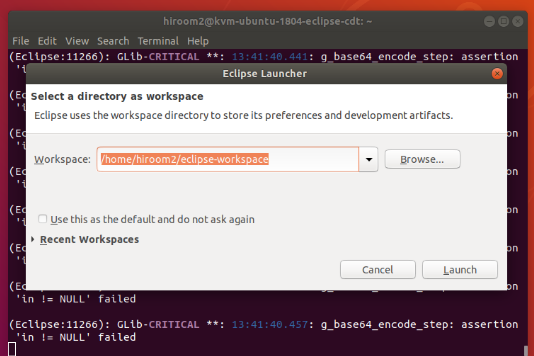


(3)啟動

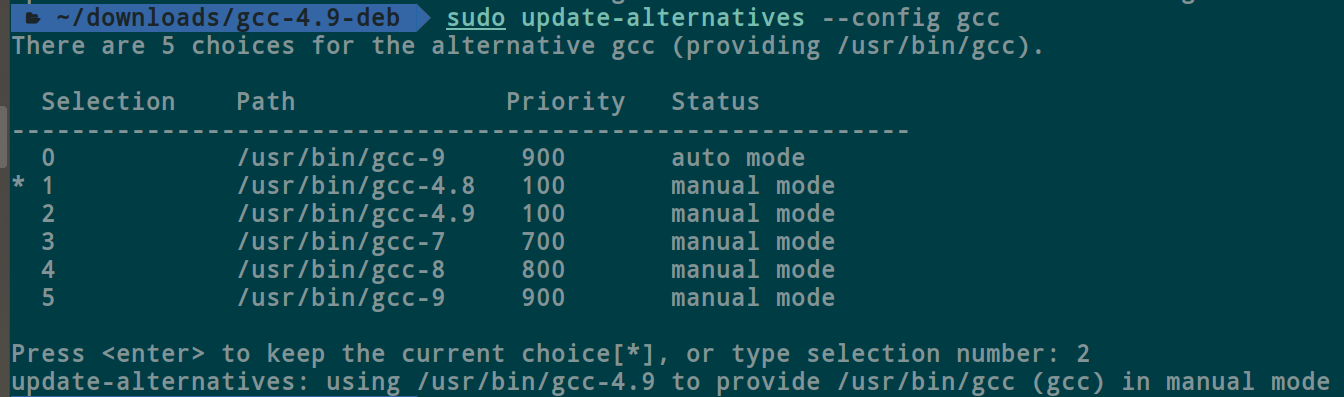
$ ./eclipse/eclipse



設定wrokspace



1. 切換gcc 4.8 (編譯kernel 4.0 需要對應到gcc4.8版本)

(1)sudo update-alternatives --config gcc ##輸入要的版本 這裡是1

1. 安裝kernel需要的工具

$ sudo apt-get install -y fakeroot kernel-package linux-source libc6-dev binutils-dev make bin86 module-init-tools build-essential

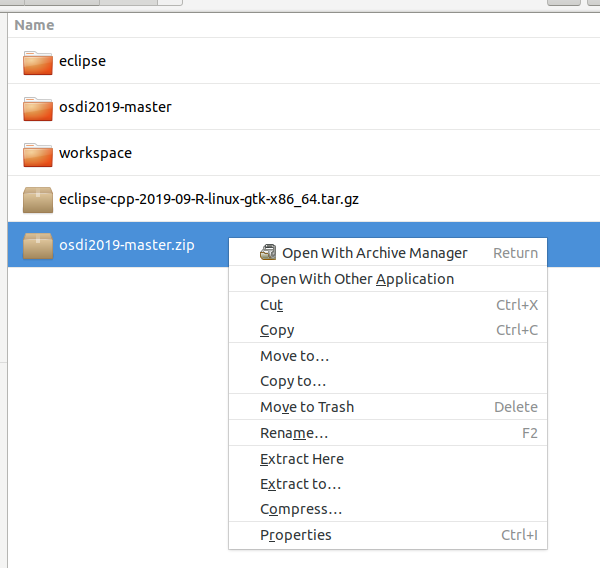
#uboot-mkimage不支持18.04先不用裝 gcc 不用裝因為有了

1. 下載Kernel

(1)網址：<https://github.com/shiwulo/osdi2019>



(2)解壓縮



1. 確認檔案

$ cd ~/osdi2019/kernel-4.0/linux-4.0/\_install\_x86

$ mkdir dev #創建dev資料夾

$ cd dev

**Sudo -s**

**mknod -m 622 ./console c 5 1**

**mknod -m 666 ./null c 1 3**

**mknod -m 666 ./zero c 1 5**

**mknod -m 666 ./ptmx c 5 2**

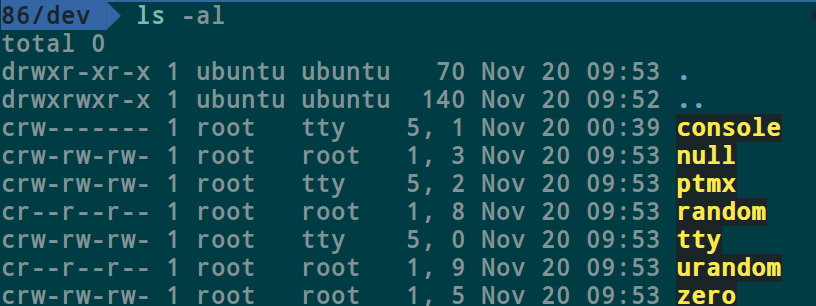
**mknod -m 666 ./tty c 5 0**

**mknod -m 444 ./random c 1 8**

**mknod -m 444 ./urandom c 1 9**

**chown root:tty ./{console,ptmx,tty}**

$ ls -al #確認



1. 編譯Linux kernel前設定

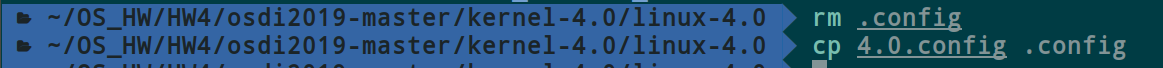
(1) menuconfig

$ cd ~/osdi2019/kernel-4.0/linux-4.0

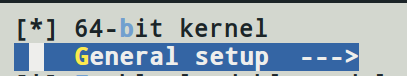
$ rm .config#刪除舊的config

$ cp 4.0.config .config #放入4.0 config

$ make menuconfig #

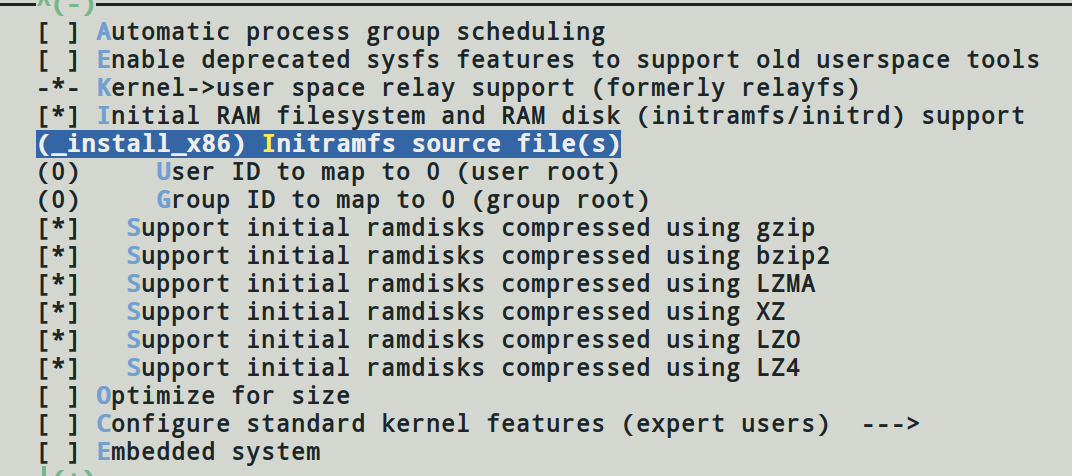


General setup



initramfs source file #設定一開始要去哪裡找起始檔案 沒有東西開不了機

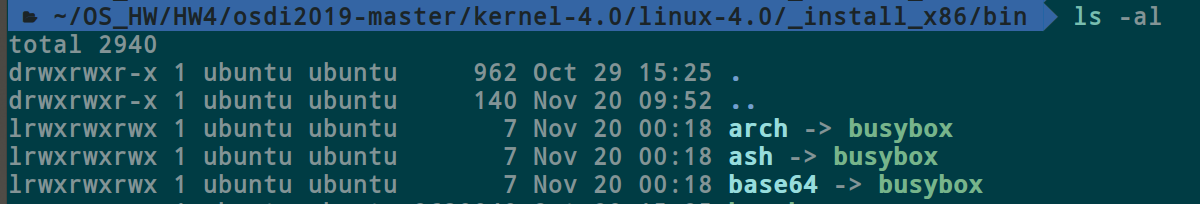
確認是否有\_install\_x86 有就按EXIT即可



(2)確認busybox

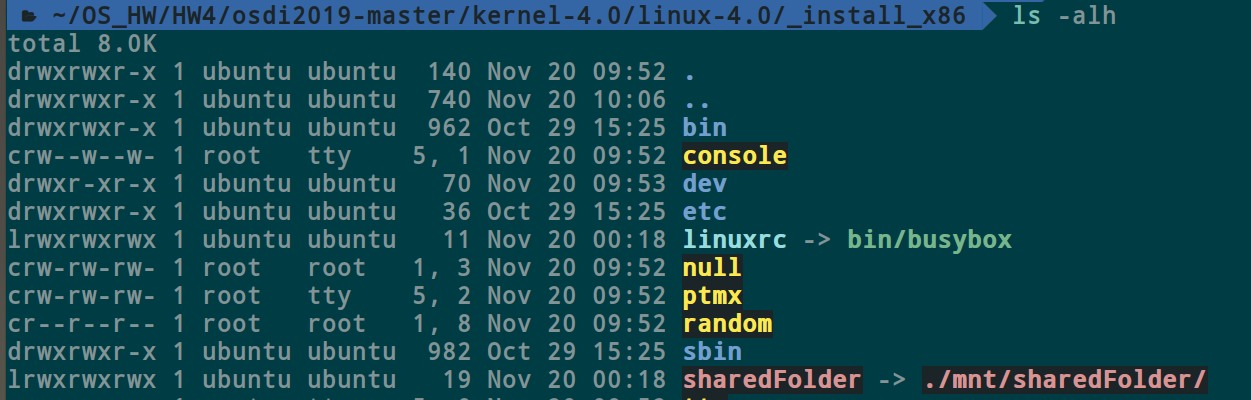
$ cd ~/osdi2019/kernel-4.0/linux-4.0/\_install\_x86/bin

$ ls -al



(3)確認sharefolder

(i) ./mnt/sharedFolder/ 目前還不存在，但可以透過etc去看



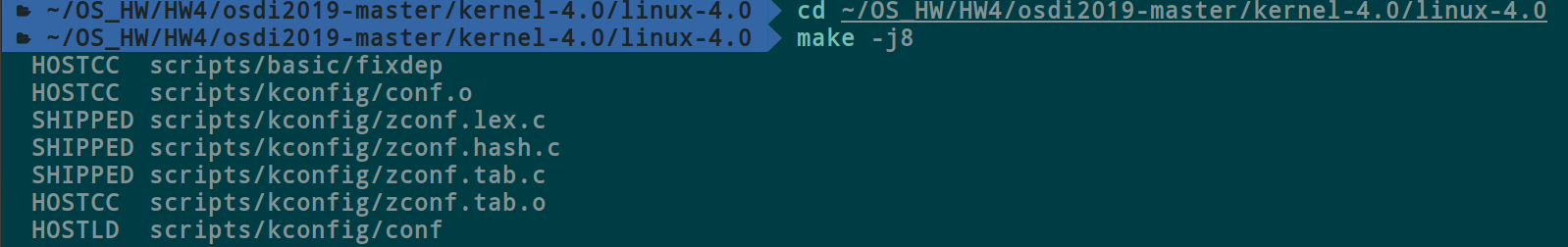
(ii) $ cd etc

$ vim fstab #vmware跟host machine共享資料夾



$ make -j #-j可改成 -j8 核心數量Ｘ2 比較快但可能出錯 成功的話重開始active kernel

#若需重新make -j8 需要先make clean一次



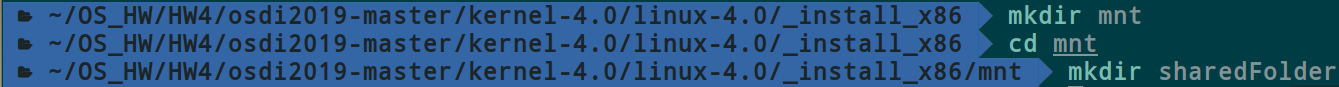
(4) 切換gcc 4.8

按照3.操作即可

1. 編譯linux kernel
   1. $ cd ~/osdi2019/kernel-4.0/linux-4.0/\_install\_x86
   2. $ mkdir mnt

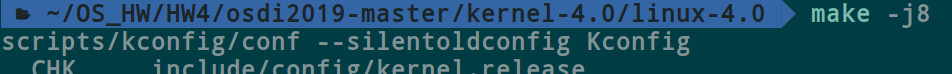
$ cd mnt

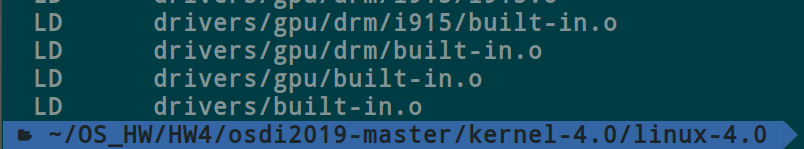
$ mkdir sharedFolder



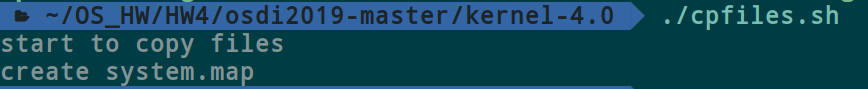
* 1. $ cd.. 到~/osdi2019/kernel-4.0/linux-4.0

$ make -j8 #開始編譯 8為核心數4\*2 可以按情況更改 這裡會很久XD 一般要1-2小時 但是老師有拿掉不需要的driver 大約4-5分鐘





1. 複製檔案與執行qemu
   1. $ cd ~/osdi2019/kernel-4.0
   2. $ ./cpfiles.sh

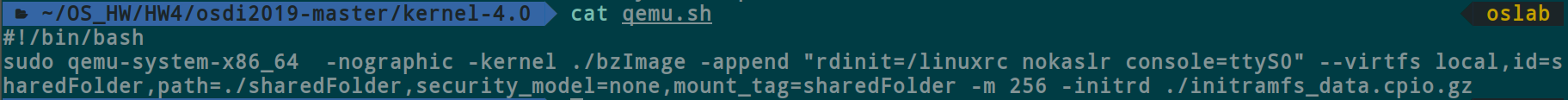


* 1. $ cat qemu.sh # -m 256 256MB 記憶體 ./initramfs\_data.cpio.gz busy box

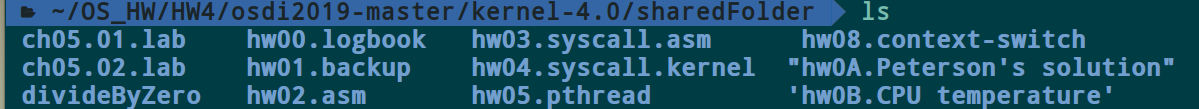
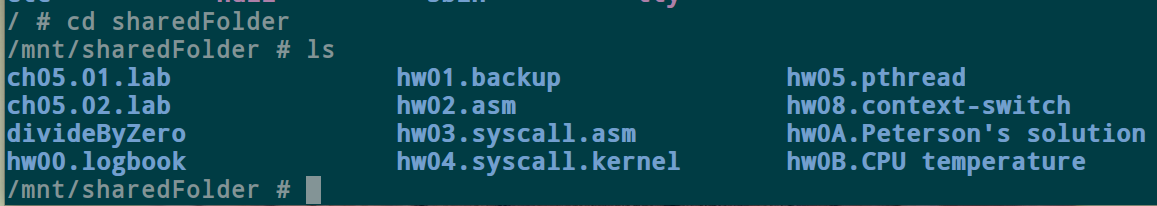
# "rdinit=/linuxrc nokaslr console=ttyS0" 一開始要執行的執行檔 和 console

# --virtfs local,id=sharedFolder #連上主機位置 -nographic 純文字輸出

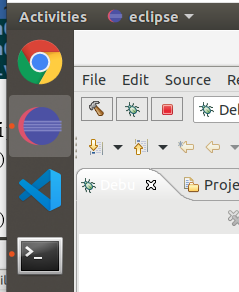
# -kernel ./bzImage 用哪個kernel -append開機的時候傳遞參數給bzimage



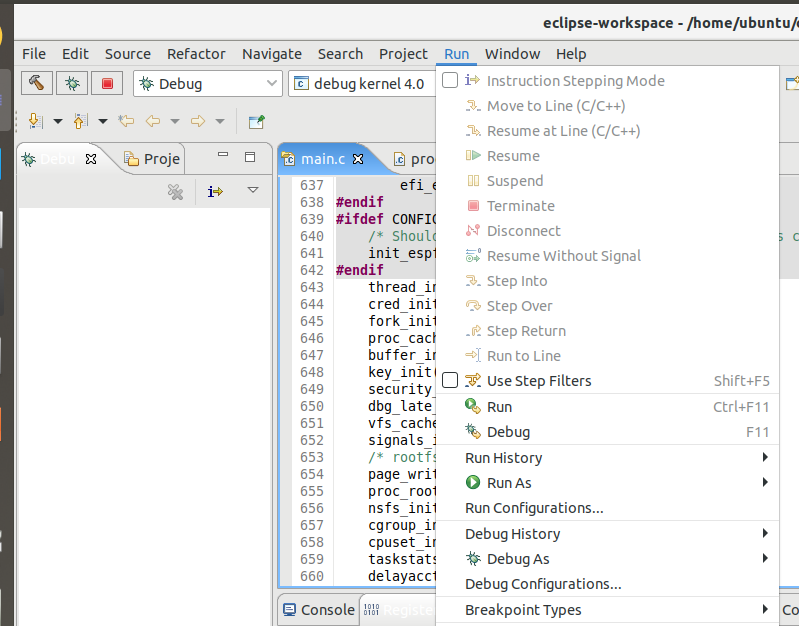
* 1. $./qemu.sh #如果有錯要創造檔案重新make -j8 發現host跟client共享資料夾



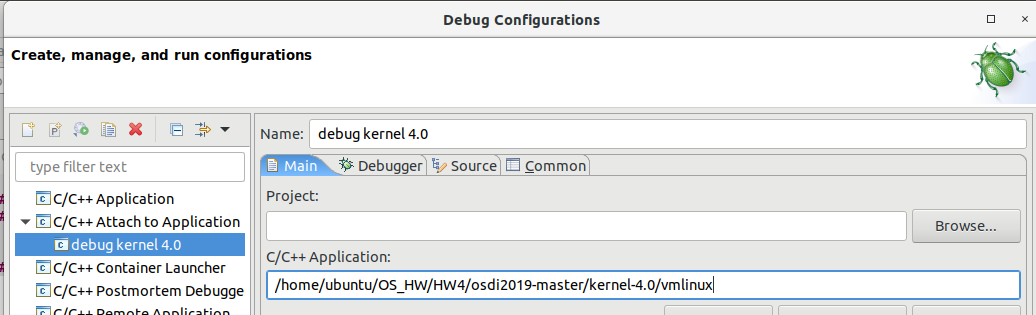
1. Eclipse教學
   1. 打開安裝好的Eclipse



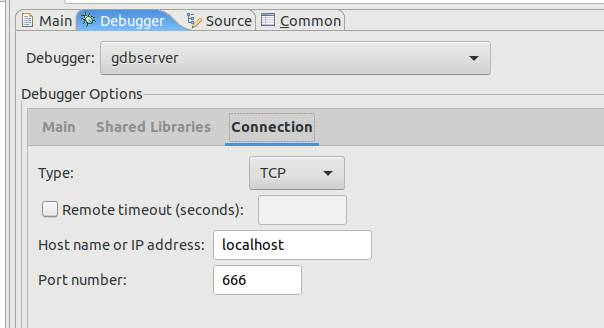
* 1. 設定Debug config



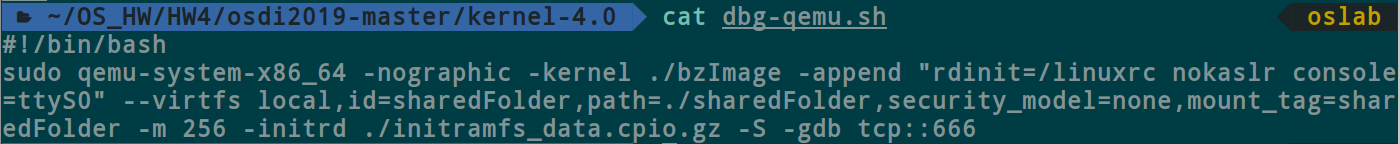
(i)mian 修改路徑到vmlinux



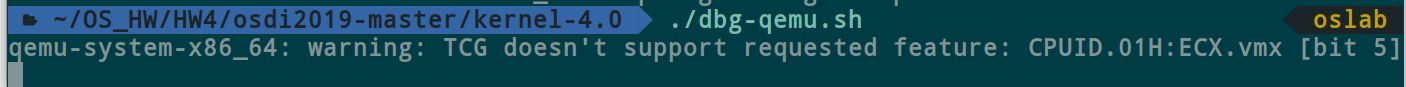
(ii)Debugger -> localhost按照此設定



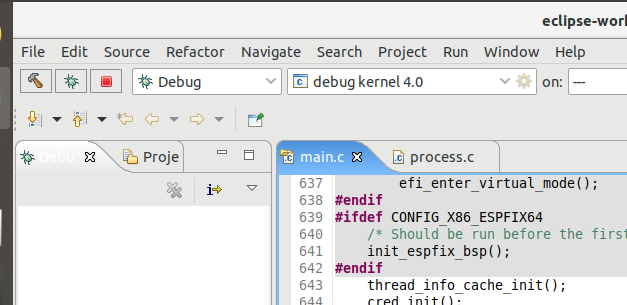
(3) cat dbg-qemu.sh # -S -gdb tcp::666 代表下指令給QEMU 用TCP做廚錯



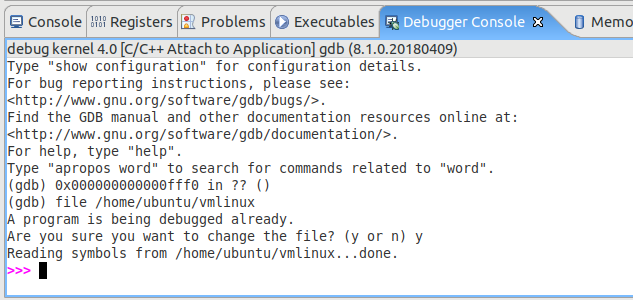
(4) $ ./dbg-qemu.sh #開始除錯會停住



(5)$ 回到Eclipse 按小蟲子



(6) 選Degugger Console 輸入file /home/ubuntu/vmlinux 按y

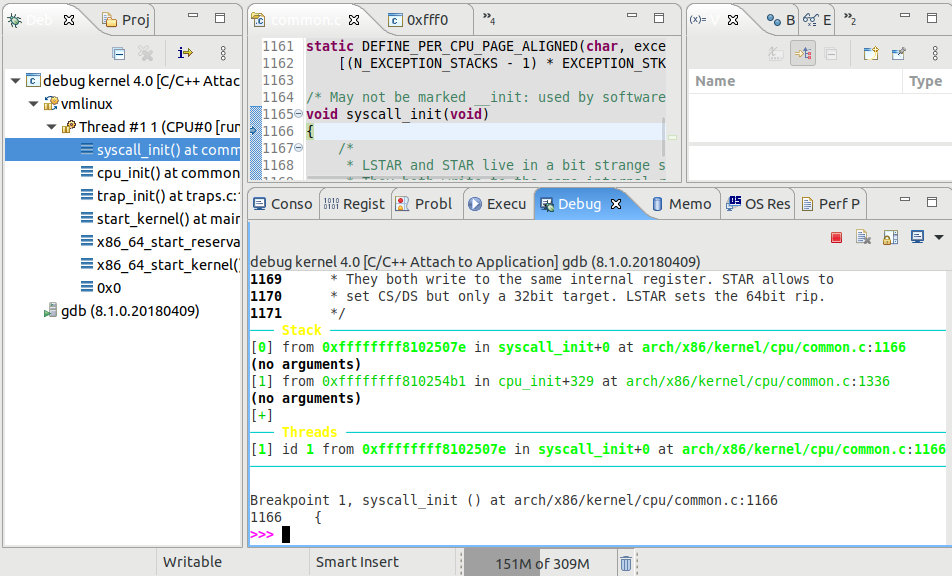


(7)輸入b syscall\_init

然後輸入c enter開始debug

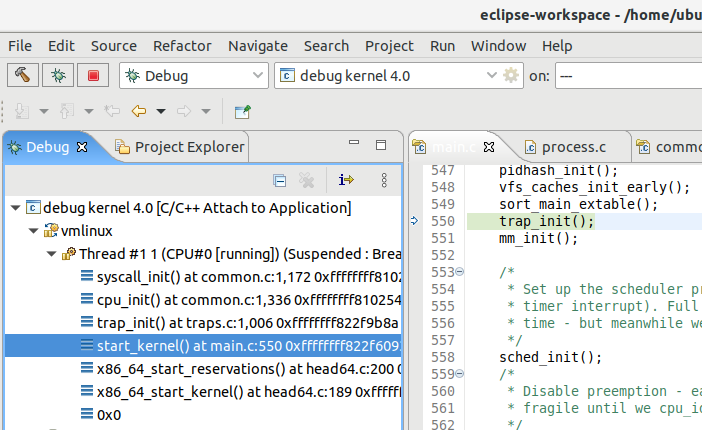


(8)追蹤到所需要的程式

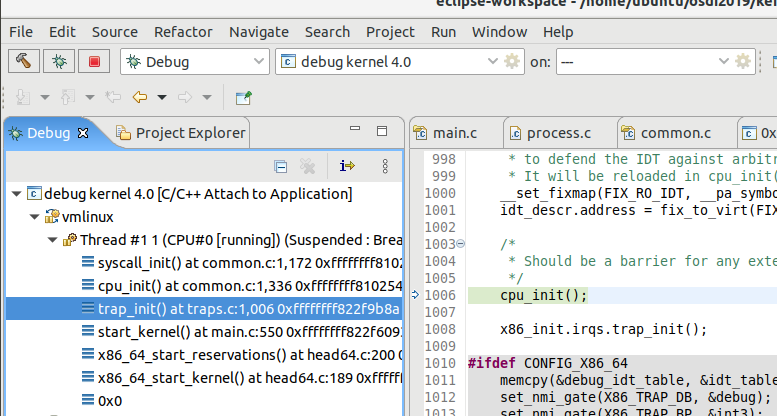


(9)追蹤過程

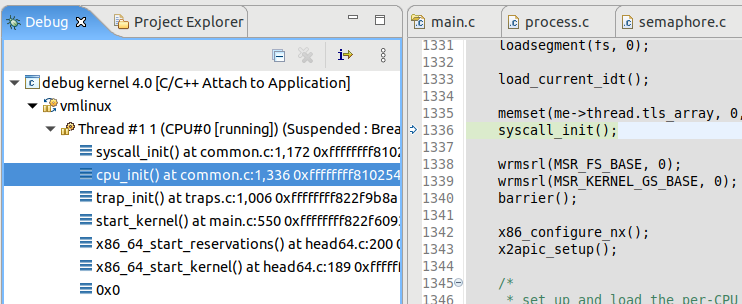
(i)start\_kernel()呼叫trap\_init()



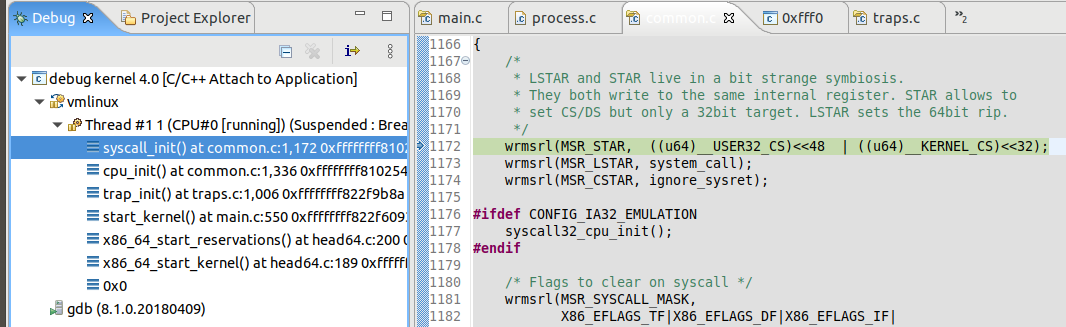
(ii) trap\_init()呼叫cpu\_inti()



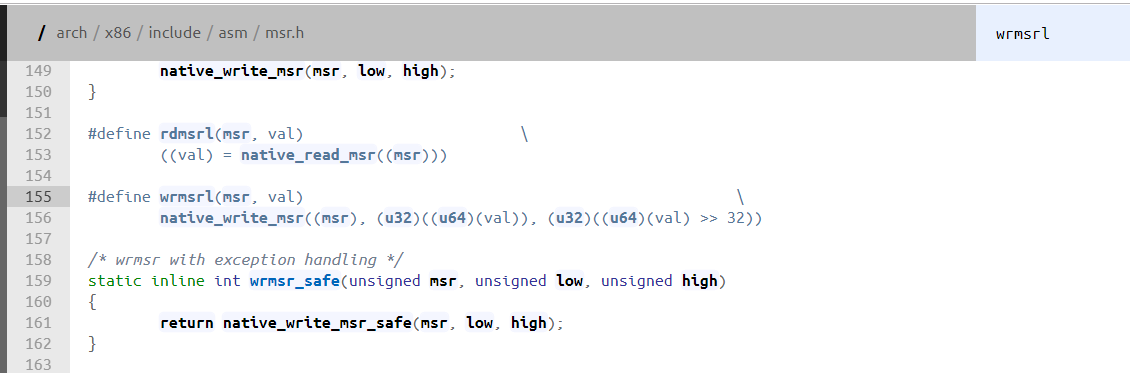
(iii)在cpu\_init()呼叫syscall\_init()



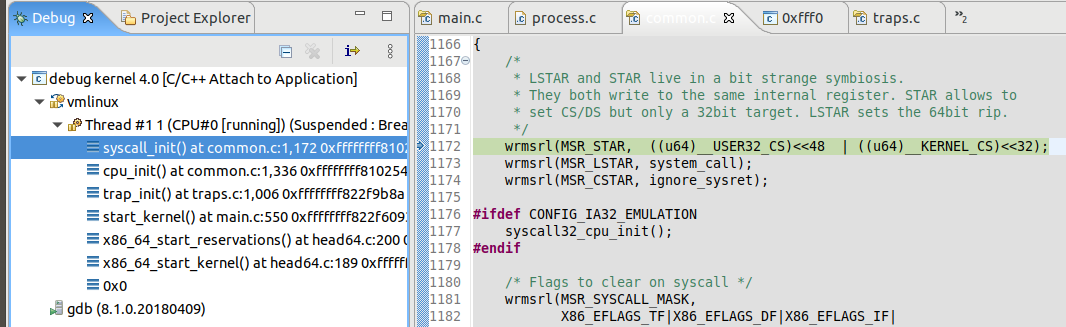
(iv) syscall\_init()用wrmsrl(MSR\_STAR, ((u64)\_\_USER32\_CS)<<48 | ((u64)\_\_KERNEL\_CS)<<32);



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

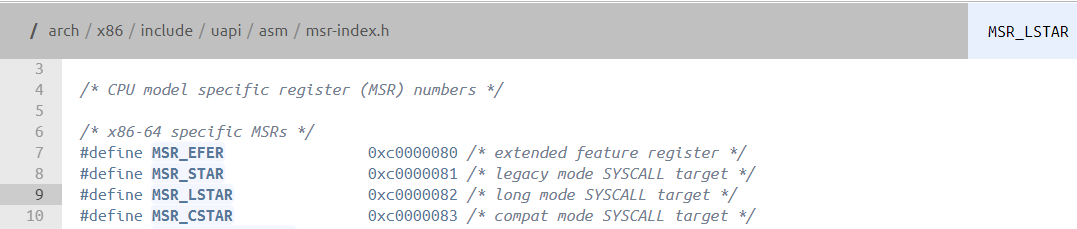


(v)wrmsrl(MSR\_LSTAR, system\_call);// 將system\_call 放到MSR\_LSTAR

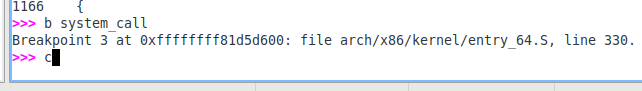


Google發現進入點在#define MSR\_LSTAR 0xc0000082 /\* long mode SYSCALL target \*/

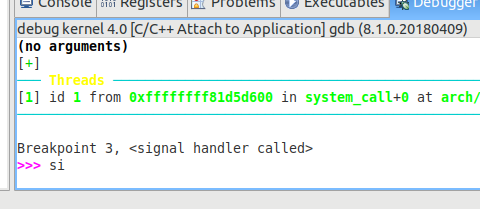
Long mode 64位元 ；compat mode為32位元



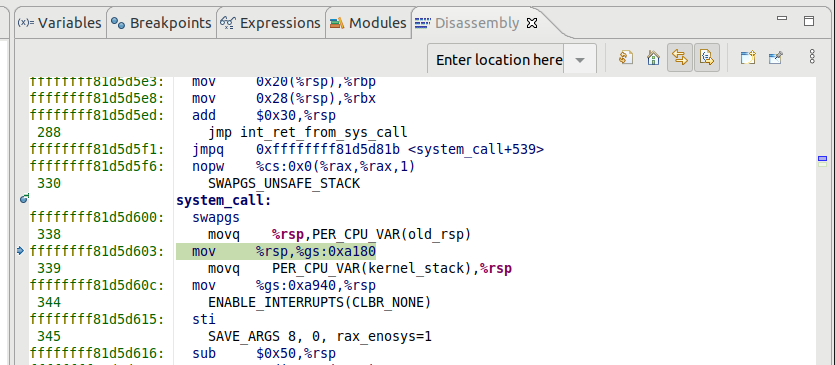
(vi)b system\_call



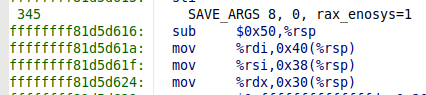
(vii)輸入si 看組合語言



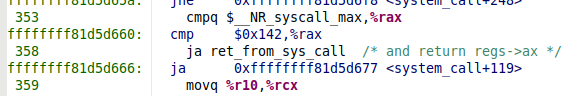
(viii)看到右邊有330 system\_call



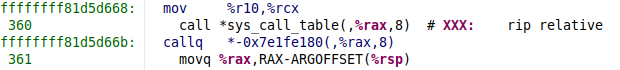
(ivv)把rdi rsi rdx放到stack



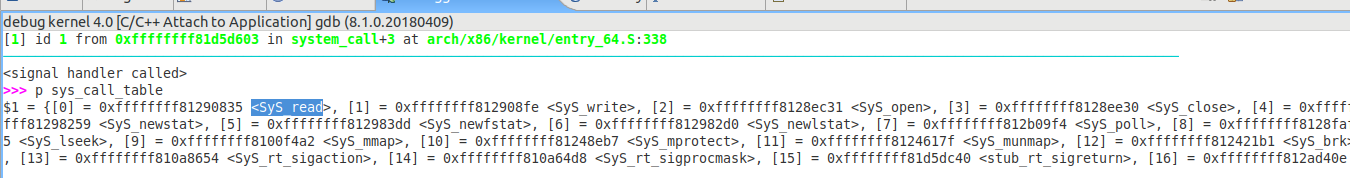
(vv)檢查rax是否超過355



(vvi)看sys\_call\_table



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



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





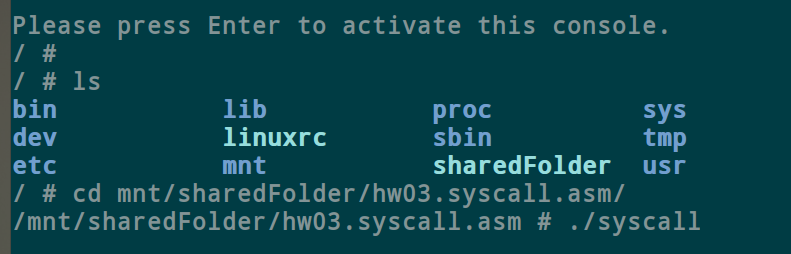
1. 追蹤其他任意程式碼

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

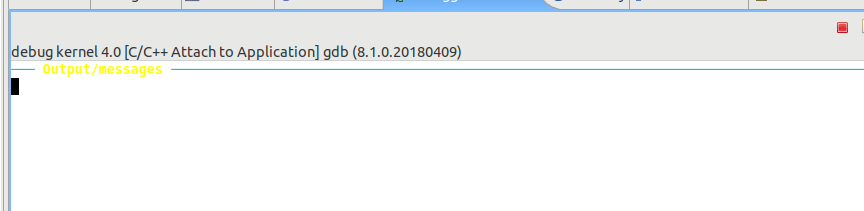
一張含有 螢幕擷取畫面 的圖片

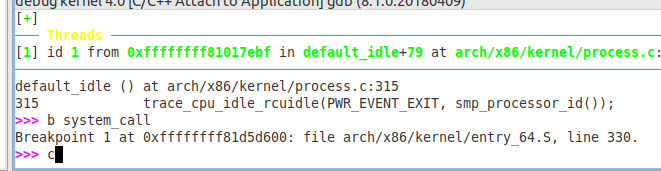
自動產生的描述

(2)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



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





(4)回到cmd按enter會再跳回eclipse

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

