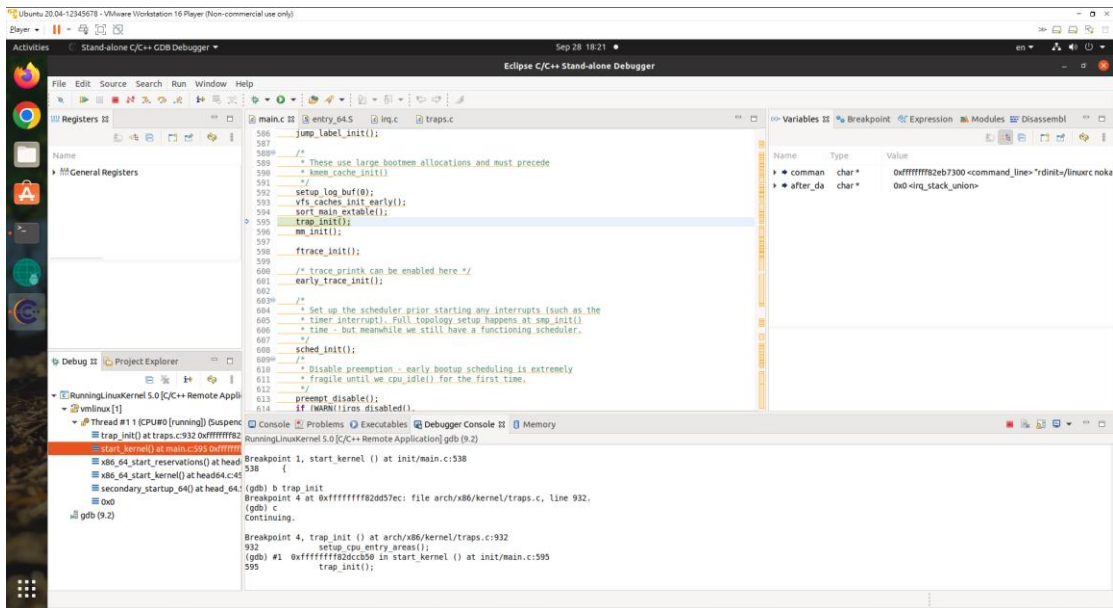


HW2 觀察中斷

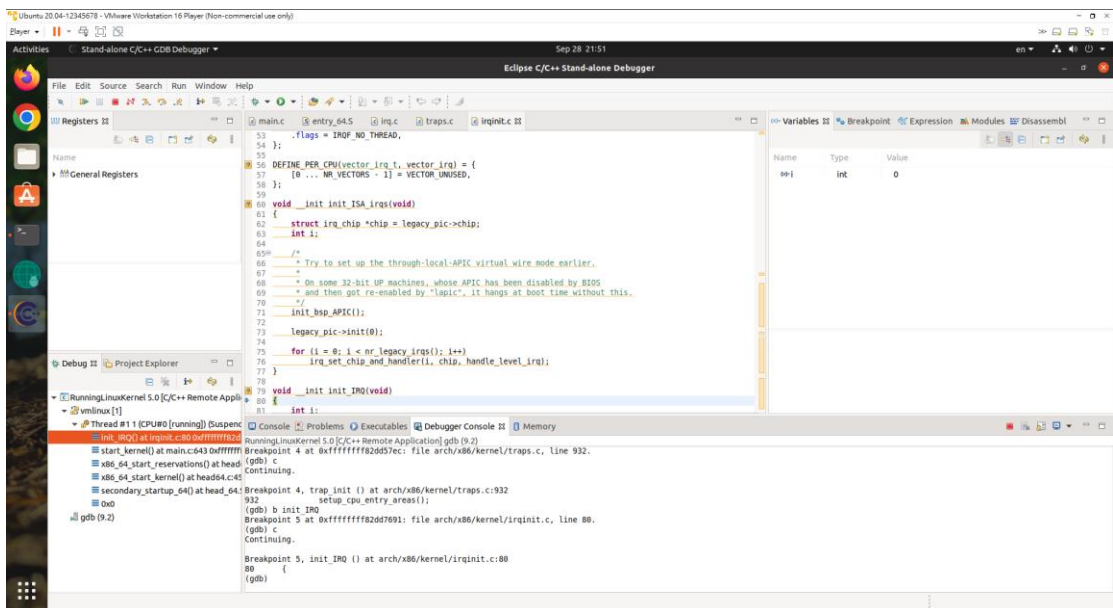
系級：資工二 學號：409410114 姓名：周述君

- 一份簡單的報告，請將題目所說的八個中斷點予以截圖

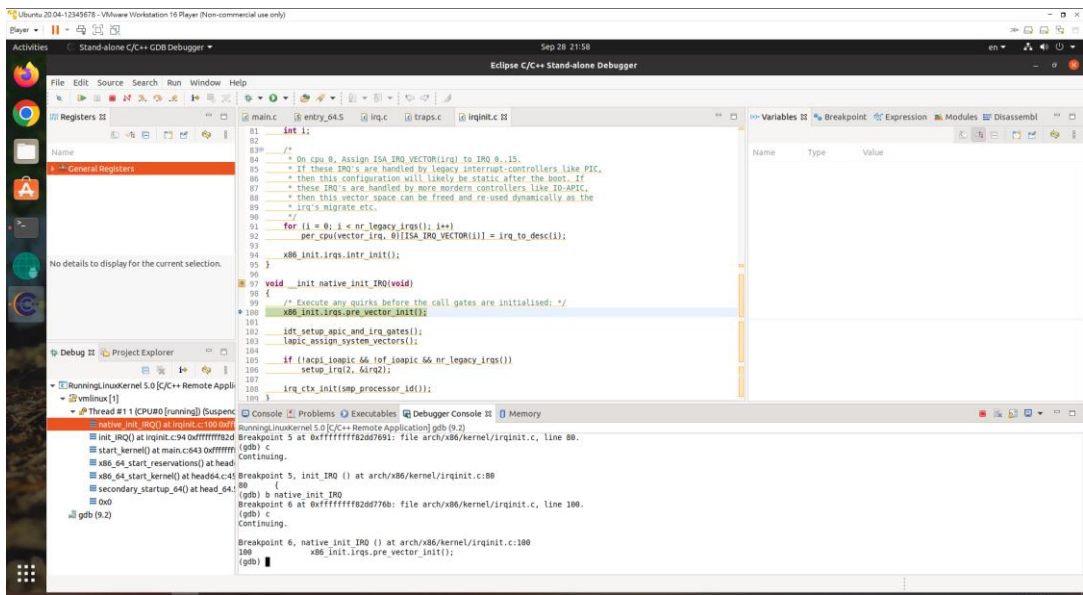
● 截圖 1. b trap_init



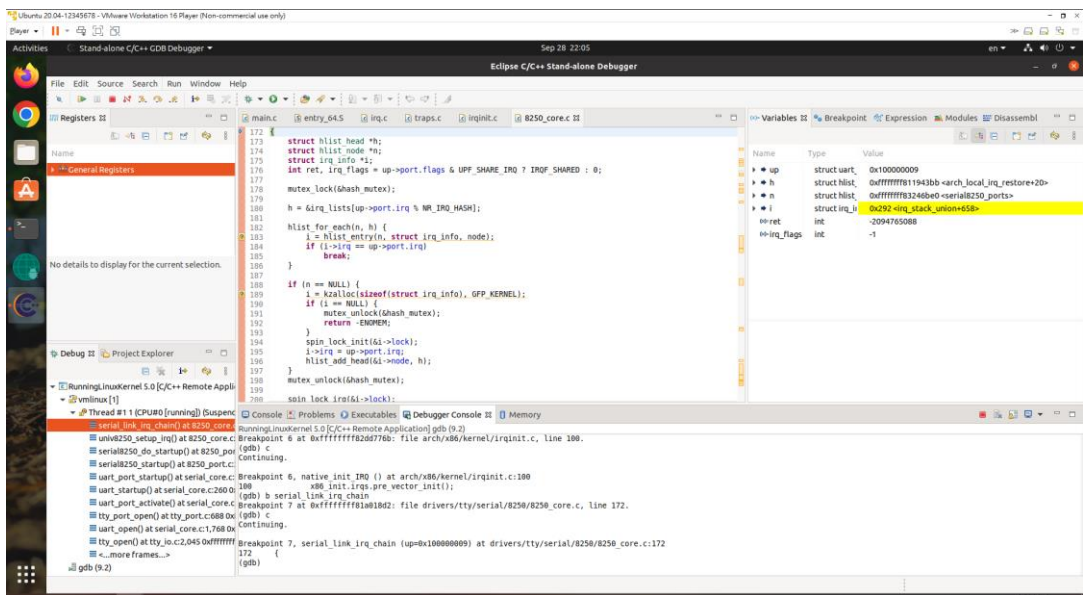
● 截圖 2. B init_IRQ



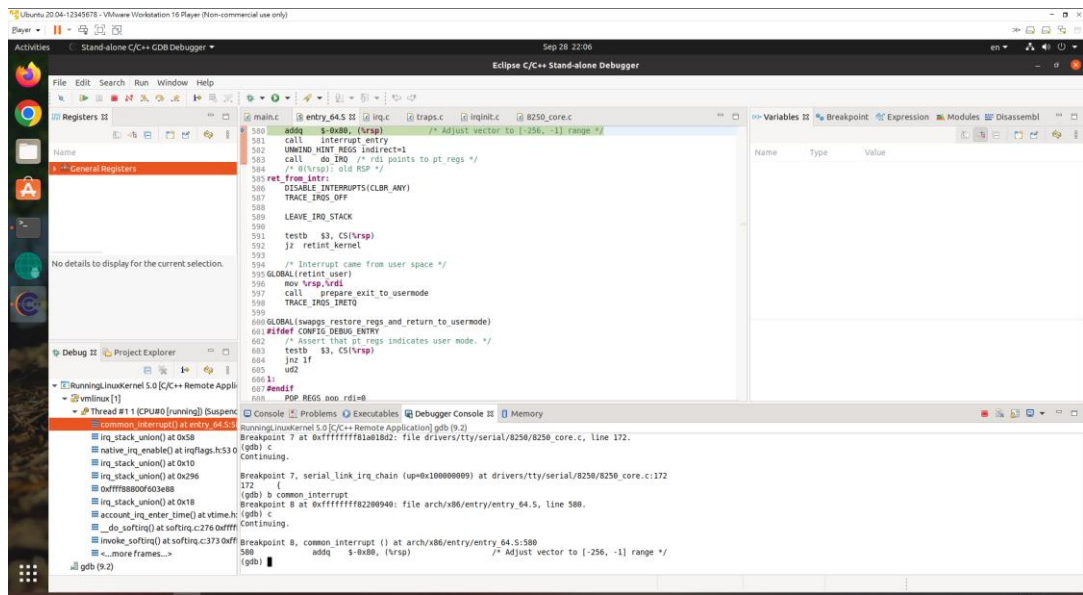
● 截圖 3. B native_init_IRQ



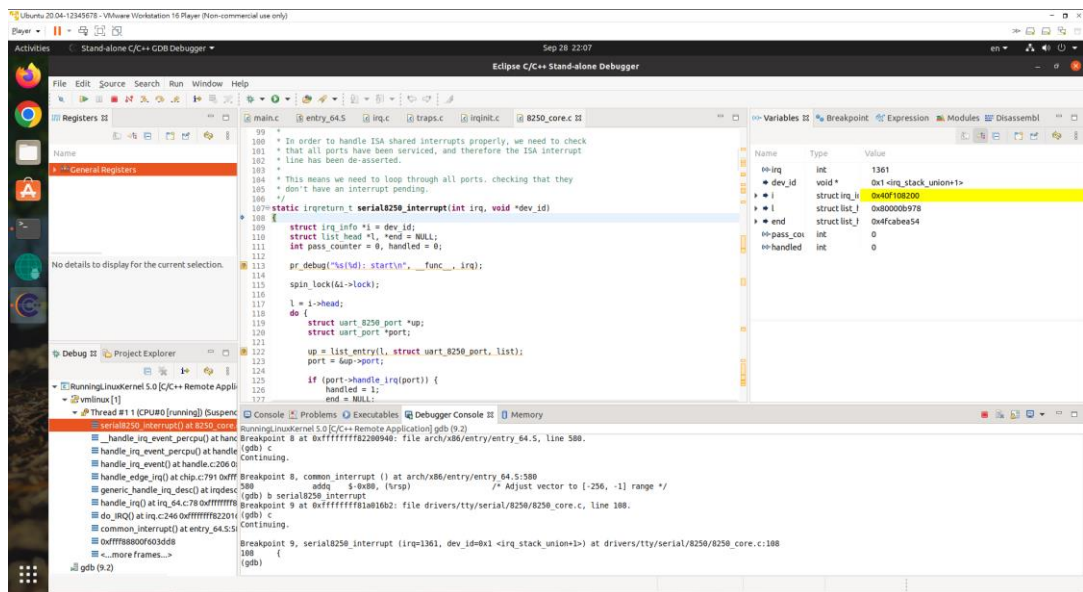
● 截圖 4.b serial_link_irq_chain



● 截圖 5.b common_interrupt



● 截圖 6.b serial8250_interrupt



● 截圖 7. disass irq_entries_start

Running.LinuxKernel 5.0 [C/C++ Remote Application] gdb (9.2)

```

(gdb) disass irq_entries_start
Dump of assembler code for function irq_entries_start:
0xfffffffff8220010 <+0>: pushq %ebx
0xfffffffff8220012 <+2>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220017 <+7>: nop
0xfffffffff8220018 <+8>: pushq %ebx
0xfffffffff822001a <+10>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822001b <+11>: nop
0xfffffffff822001c <+12>: pushq %ebx
0xfffffffff822001e <+14>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822001f <+15>: nop
0xfffffffff8220020 <+16>: pushq %ebx
0xfffffffff8220022 <+18>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220023 <+19>: nop
0xfffffffff8220024 <+20>: pushq %ebx
0xfffffffff8220026 <+22>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220027 <+23>: nop
0xfffffffff8220028 <+24>: pushq %ebx
0xfffffffff822002a <+26>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822002b <+27>: nop
0xfffffffff822002c <+28>: pushq %ebx
0xfffffffff822002e <+30>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822002f <+31>: nop
0xfffffffff8220030 <+32>: pushq %ebx
0xfffffffff8220032 <+34>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220033 <+35>: nop
0xfffffffff8220034 <+36>: pushq %ebx
0xfffffffff8220036 <+38>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220037 <+39>: nop
0xfffffffff8220038 <+40>: pushq %ebx
0xfffffffff822003a <+42>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822003b <+43>: nop
0xfffffffff822003c <+44>: pushq %ebx
0xfffffffff822003e <+46>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822003f <+47>: nop
0xfffffffff8220040 <+48>: pushq %ebx
0xfffffffff8220042 <+50>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220043 <+51>: nop
0xfffffffff8220044 <+52>: pushq %ebx
0xfffffffff8220046 <+54>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220047 <+55>: nop
0xfffffffff8220048 <+56>: pushq %ebx
0xfffffffff822004a <+58>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822004b <+59>: nop
0xfffffffff822004c <+60>: pushq %ebx
0xfffffffff822004e <+62>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822004f <+63>: nop
  
```

● 截圖 8. b*(irq_entries_start+56)

Running.LinuxKernel 5.0 [C/C++ Remote Application] gdb (9.2)

```

(gdb) disass b*(irq_entries_start+56)
Dump of assembler code for function b*(irq_entries_start+56):
0xfffffffff8220042 <+1586>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220047 <+1591>: nop
0xfffffffff8220048 <+1592>: pushq %ebx
0xfffffffff822004a <+1594>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822004b <+1595>: nop
0xfffffffff822004c <+1596>: pushq %ebx
0xfffffffff822004e <+1598>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822004f <+1599>: nop
0xfffffffff8220050 <+1600>: pushq %ebx
0xfffffffff8220052 <+1602>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220053 <+1603>: nop
0xfffffffff8220054 <+1604>: pushq %ebx
0xfffffffff8220056 <+1606>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220057 <+1607>: nop
0xfffffffff8220058 <+1608>: pushq %ebx
0xfffffffff822005a <+1610>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822005b <+1611>: nop
0xfffffffff822005c <+1612>: pushq %ebx
0xfffffffff822005e <+1614>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822005f <+1615>: nop
0xfffffffff8220060 <+1616>: pushq %ebx
0xfffffffff8220062 <+1618>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220063 <+1619>: nop
0xfffffffff8220064 <+1620>: pushq %ebx
0xfffffffff8220066 <+1622>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff8220067 <+1623>: nop
0xfffffffff8220068 <+1624>: pushq %ebx
0xfffffffff822006a <+1626>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822006b <+1627>: nop
0xfffffffff822006c <+1628>: pushq %ebx
0xfffffffff822006e <+1630>: jmpq 0xfffffffff82200940 <common_interrupt>
0xfffffffff822006f <+1631>: nop
  
```

2. 在（問題 1.）的報告中，說明 Linux 如何設定中斷向量

Linux 中斷向量表分為內部和外部中斷

CPU 內建的中斷事件又稱 **software interrupt** 或 **trap**，linux 在 **start_kernel** 中會先呼叫 **trap_init** 來初始化處理器的 **trap**，再將 CPU 內部中斷的中段處理函數寫入中斷向量表。

外部中斷的部分是由 **start_kernel** 呼叫 **init_IRQ** 來初始化 16 個一般外部中斷陣列，再呼叫 **x86_init.irqs.intr_init()** 建立外部中斷向量表，直到呼叫 **set_intr_gate** 為止。

Dump of assembler code for function **irq_entries_start**:

```
0xffffffff82200210 <+0>:    pushq  $0x5f
0xffffffff82200212 <+2>:    jmpq   0xffffffff82200940 <common_interrupt>
0xffffffff82200217 <+7>:    nop
0xffffffff82200218 <+8>:    pushq  $0x5e
0xffffffff8220021a <+10>:   jmpq   0xffffffff82200940 <common_interrupt>
0xffffffff8220021f <+15>:   nop
```


3. 在（問題 1.）的報告中，說明 Linux 如何從中斷向量的組合語言部分（interrupt service routine，這裡只討論外部中斷）跳躍到特定的中斷函數以 serial port 為例，他是第四號外部中斷(從/proc/irq/4 可知)

```
0xffffffff82200248 <+56>:  pushq  $0x58
0xffffffff8220024a <+58>:  jmpq   0xffffffff82200940 <common_interrupt>
0xffffffff8220024f <+63>:  nop
```

特別要注意的是，第四號中斷在程式碼中到底是 irq_entries_start 中的第 X 號組合語言並非一對一的對應，例如在這個例子中是「第八號組合語言」。Linux 對這個中斷的「軟體編號」是「0x58」。

由 common_interrupt 的程式碼發現，所有中斷服務的函數都會跳到下面這段組合語言，呼叫 interrupt_entry 將所有暫存器放入堆疊，這部分的重點是「製造堆疊」，堆疊內的資料型態為「pt_regs」，並且將專斷的軟體編號(0x58)放到 pr_regs 的 orig_ax。

```
common_interrupt:
addq  $-0x80, (%rsp)          /* Adjust vector to [-256, -1] range */
call  interrupt_entry
UNWIND_HINT_REGS indirect=1
call  do_IRQ                 /* rdi points to pt_regs */
```

問題一的截圖六，如果裝置發出中斷，do_IRQ 會把中斷向量的編號記錄下來，再根據編號往下找函數指標，接著從 do_IRQ 一層一層呼叫到 serial8250_interrupt，由截圖七可看到中斷向量所指向的程式碼。其中 do_IRQ 為 C 語言，do_IRQ 的程式如下，這部分的重點是：由於 orig_ax 放的是「中斷的軟體編號」，因此將這個編號作為「中斷向量物件」的索引，即 __this_cpu_read(vector_irq[vector])，並在 handle_irq 中呼叫該函數。在這個例子中的函數即 serial8250_interrupt。約略等同於下列程式碼 desc = __this_cpu_read(vector_irq[vector]);desc->action->handler(...);