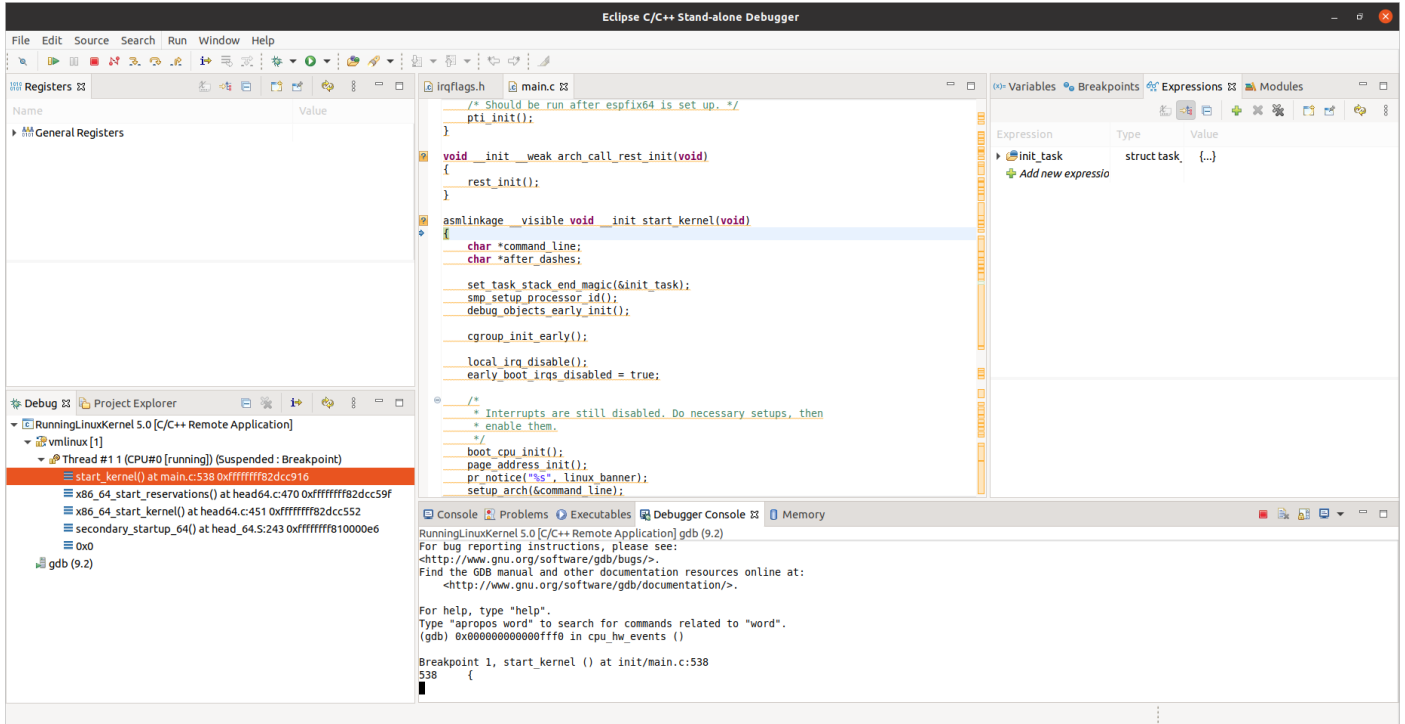


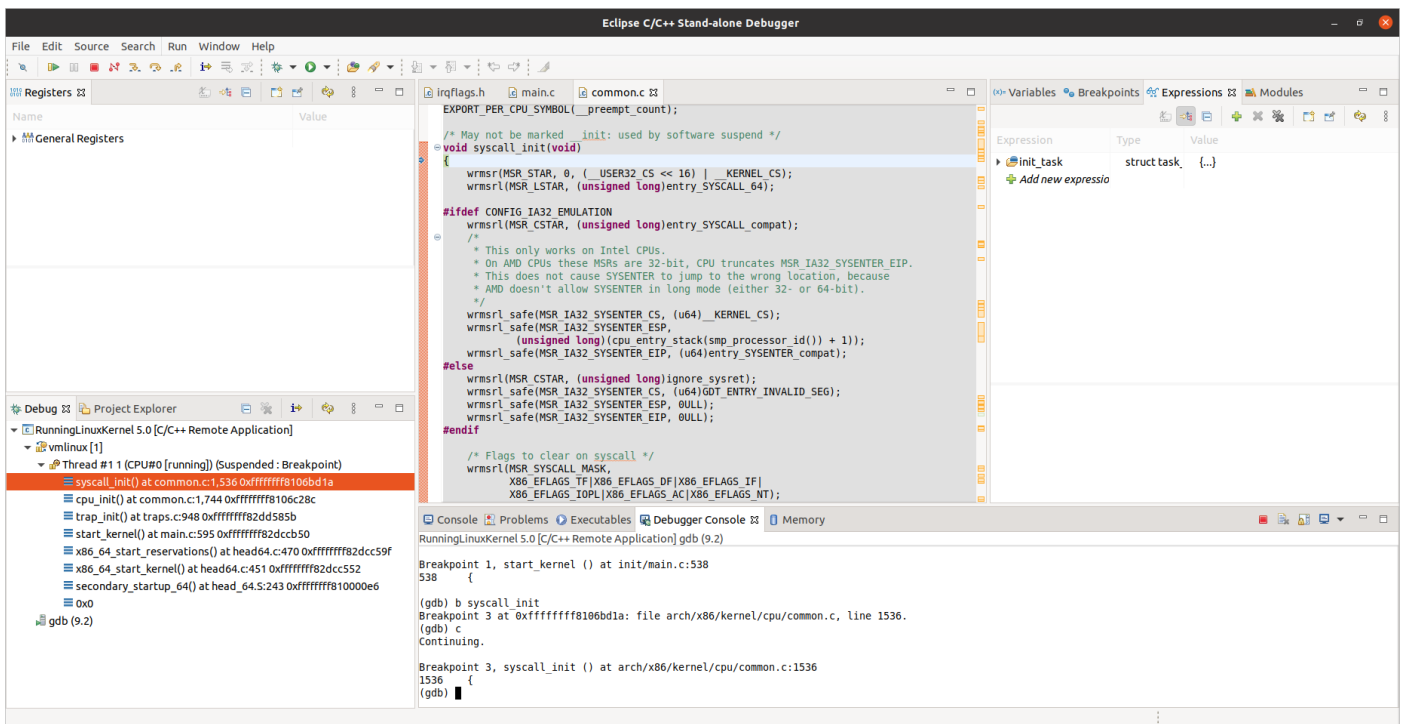
# 作業系統概論 hw1

學號: 408410113 姓名: 王 X 彥

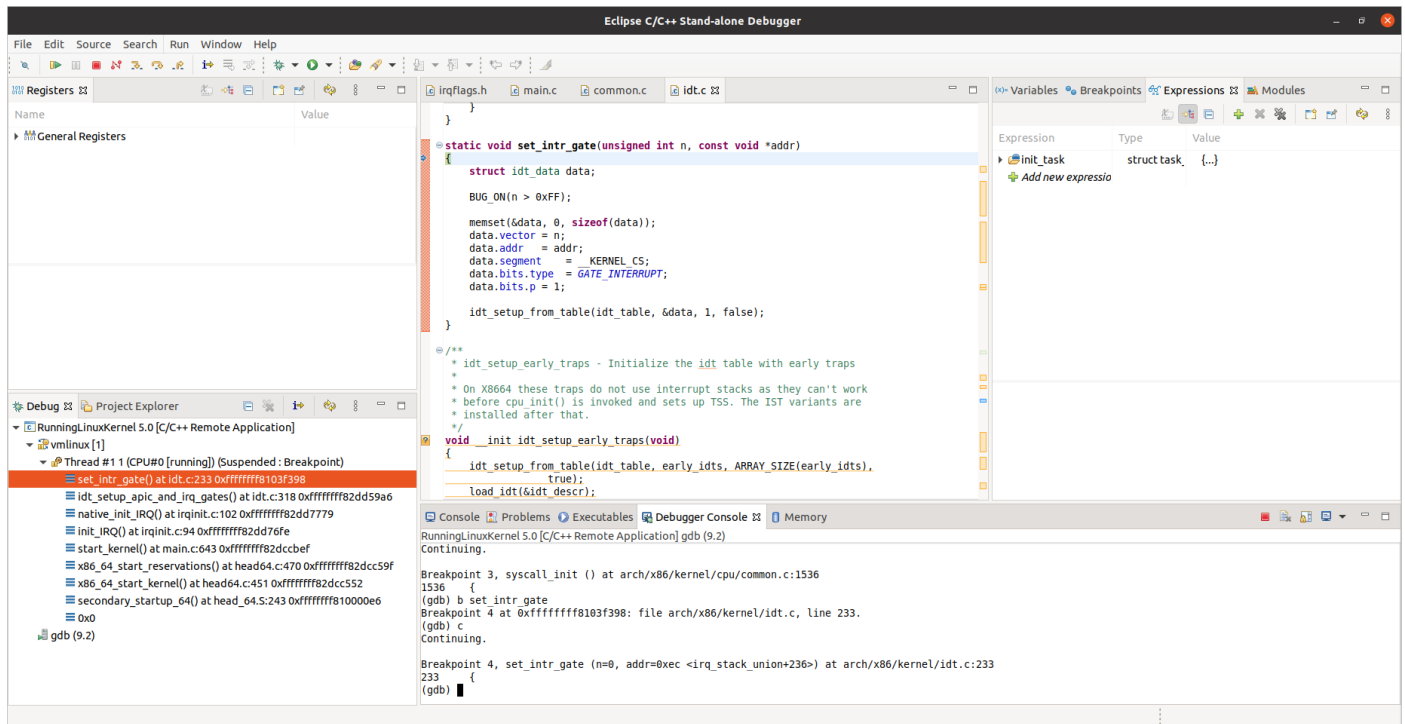
## 1. start\_kernel



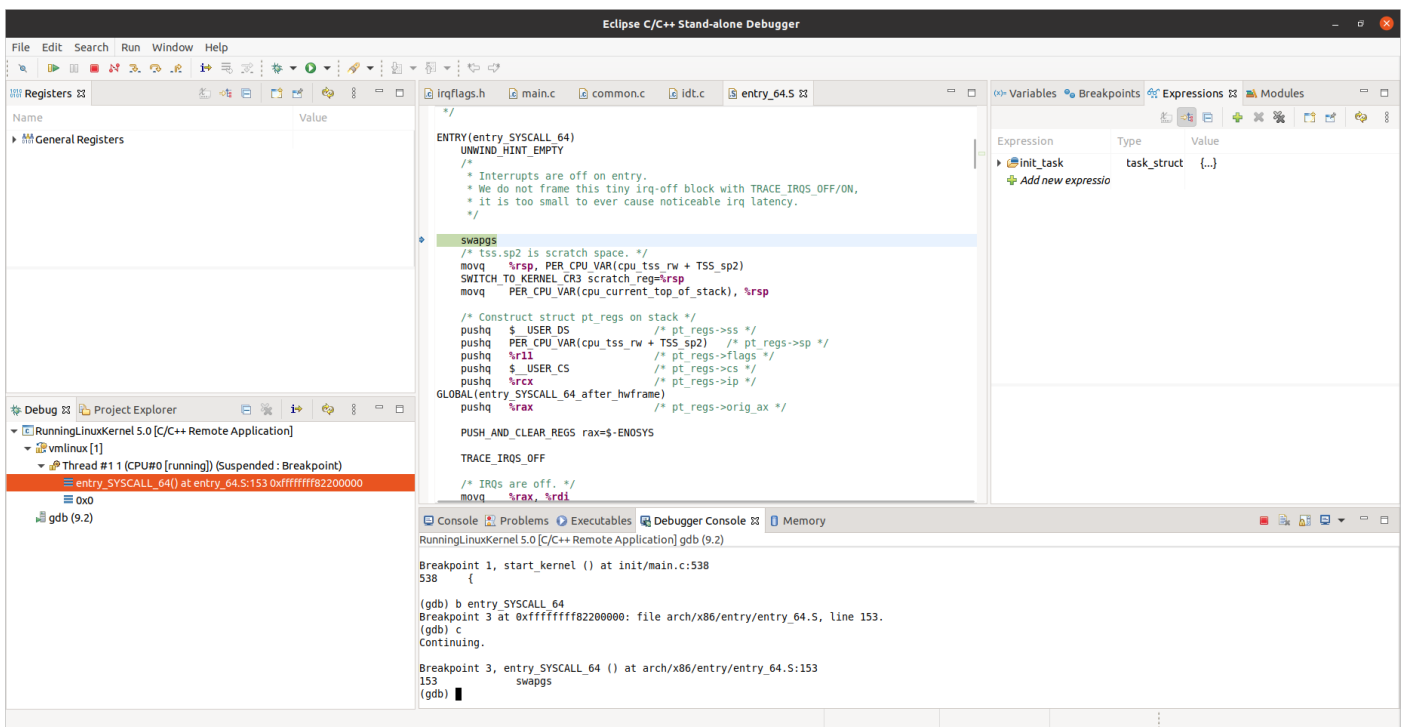
## 2. syscall\_init



### 3. set\_intr\_gat



### 4. entry\_SYSCALL\_64



## 5. apic\_timer\_interrupt

The screenshot shows the Eclipse C/C++ Stand-alone Debugger interface. The main window displays the source code of the `apic_timer_interrupt` function in `entry_64.S`. The function is defined as follows:

```
#ifndef CONFIG SMP
apicinterrupt3 IRQ_MOVE_CLEANUP_VECTOR irq_move_cleanup_interrupt smp_irq_move_clear
apicinterrupt3 REBOOT_VECTOR reboot_interrupt smp_reboot_interrupt
#endif

#ifdef CONFIG X86 UV
apicinterrupt3 UV_BAU_MESSAGE uv_bau_message_intr1 uv_bau_message_interrupt
#endif

apicinterrupt LOCAL_TIMER_VECTOR apic_timer_interrupt smp_apic_timer_interrupt
apicinterrupt X86_PLATFORM_IPI_VECTOR x86_platform_ipi smp_x86_platform_ipi

#ifdef CONFIG HAVE_KVM
apicinterrupt3 POSTED_INTR_WAKEUP_VECTOR kvm_posted_intr_wakeup_ipi smp_kvm_posted_intr_ipi
apicinterrupt3 POSTED_INTR_NESTED_VECTOR kvm_posted_intr_nested_ipi smp_kvm_posted_intr_ipi
#endif

#ifdef CONFIG X86 MCE_THRESHOLD
apicinterrupt THRESHOLD_APIC_VECTOR threshold_interrupt smp_threshold_interrupt
#endif

#ifdef CONFIG X86 MCE_AMD
apicinterrupt DEFERRED_ERROR_VECTOR deferred_error_interrupt smp_deferred_error_interrupt
#endif

#ifdef CONFIG X86 THERMAL_VECTOR
apicinterrupt THERMAL_APIC_VECTOR thermal_interrupt smp_thermal_interrupt
#endif
```

The left pane shows the Registers window with the following values:

Name	Value
rax	0
rbx	0
rcx	0
rdx	0
rsi	-60473136857256
rdi	158
rbp	0x0<irq_stack_union>
rsp	0xffffc9000028beeb
r8	0
r9	0
r10	0
r11	0
r12	0

The bottom pane shows the Console window with the following output:

```
RunningLinuxKernel 5.0 [C/C++ Remote Application] gdb (9.2)
Continuing.

Breakpoint 3, entry_SYSCALL_64 () at arch/x86/entry/entry_64.S:153
153 swaps
(gdb) b apic_timer_interrupt
Breakpoint 4 at 0xfffffff822014f0: file arch/x86/entry/entry_64.S, line 807.
(gdb) c
Continuing.

Breakpoint 4, apic_timer_interrupt () at arch/x86/entry/entry_64.S:807
807 apicinterrupt LOCAL_TIMER_VECTOR apic_timer_interrupt smp_apic_timer_interrupt
(gdb)
```

## 6. interrupt\_entry

The screenshot shows the Eclipse C/C++ Stand-alone Debugger interface. The main window displays the source code of the `interrupt_entry` function in `entry_64.S`. The function is defined as follows:

```
regs->rsp
regs->eflags
regs->cs
regs->ip
+-----+
+ | regs->orig_ax = ~(interrupt number) | +-----+
+-----+
+ | return address | +-----+
+-----+

ENTRY(interrupt_entry)
UNWIND_HINT_FUNC
ASM CLAC
cld

testb $3, CS-ORIG_RAX+8(%rsp)
jz 1f
SWAPGS

/* Switch to the thread stack. The IRET frame and orig_ax are
 * on the stack, as well as the return address. RDI..R12 are
 * not (yet) on the stack and space has not (yet) been
 * allocated for them.
 */
pushq %rdi

/* Need to switch before accessing the thread stack. */
SWITCH TO KERNEL CR3 scratch reg=%rdi
```

The left pane shows the Registers window with the following values:

Name	Value
rax	0
rbx	0
rcx	0
rdx	0
rsi	-60473136857256
rdi	158
rbp	0x0<irq_stack_union>
rsp	0xffffc9000028beeb
r8	0
r9	0
r10	0
r11	0
r12	0

The bottom pane shows the Console window with the following output:

```
RunningLinuxKernel 5.0 [C/C++ Remote Application] gdb (9.2)
Continuing.

Breakpoint 3, interrupt_entry () at arch/x86/entry/entry_64.S:375
375 .endbr
(gdb) n
505
(gdb) cld
(gdb)
```

## 7. do\_IRQ

The screenshot displays the Eclipse C/C++ Stand-alone Debugger interface. The main window shows the source code of the `do_IRQ` function in `kernel.c`. The function is responsible for handling normal device IRQs. The code includes comments about SMP cross-CPU interrupts and handlers. The function signature is `visible unsigned int __irq_entry do_IRQ(struct pt_regs *regs)`. The code defines a `struct pt_regs` and a `struct irq_desc`. It then enters a loop to handle the IRQ, including calling `ack_APIC_irq()` and `handle_irq(desc, regs)`. The code also includes a `pr_emerg_ratelimited` call to log the event.

The left sidebar shows the Registers window with the following values:

Name	Value
rax	646
rbx	0
rcx	0
rdx	0
rsi	-2111829686
rdi	-2103427848
rbp	0x0<irq_stack_union>
rsp	0xfffff8800f603ff0
r8	0
r9	0
r10	0
r11	0
r12	0

The bottom console shows the following output:

```
RunningLinuxKernel 5.0 [C/C++ Remote Application] gdb (9.2)
Breakpoint 1, start_kernel () at init/main.c:538
538 {
(gdb) b do_IRQ
Breakpoint 3 at 0xfffff82201644: file arch/x86/kernel/irq.c, line 233.
(gdb) c
Continuing.
Breakpoint 3, do_IRQ (regs=0x0 <irq_stack_union>) at arch/x86/kernel/irq.c:233
233 {
(gdb)
```