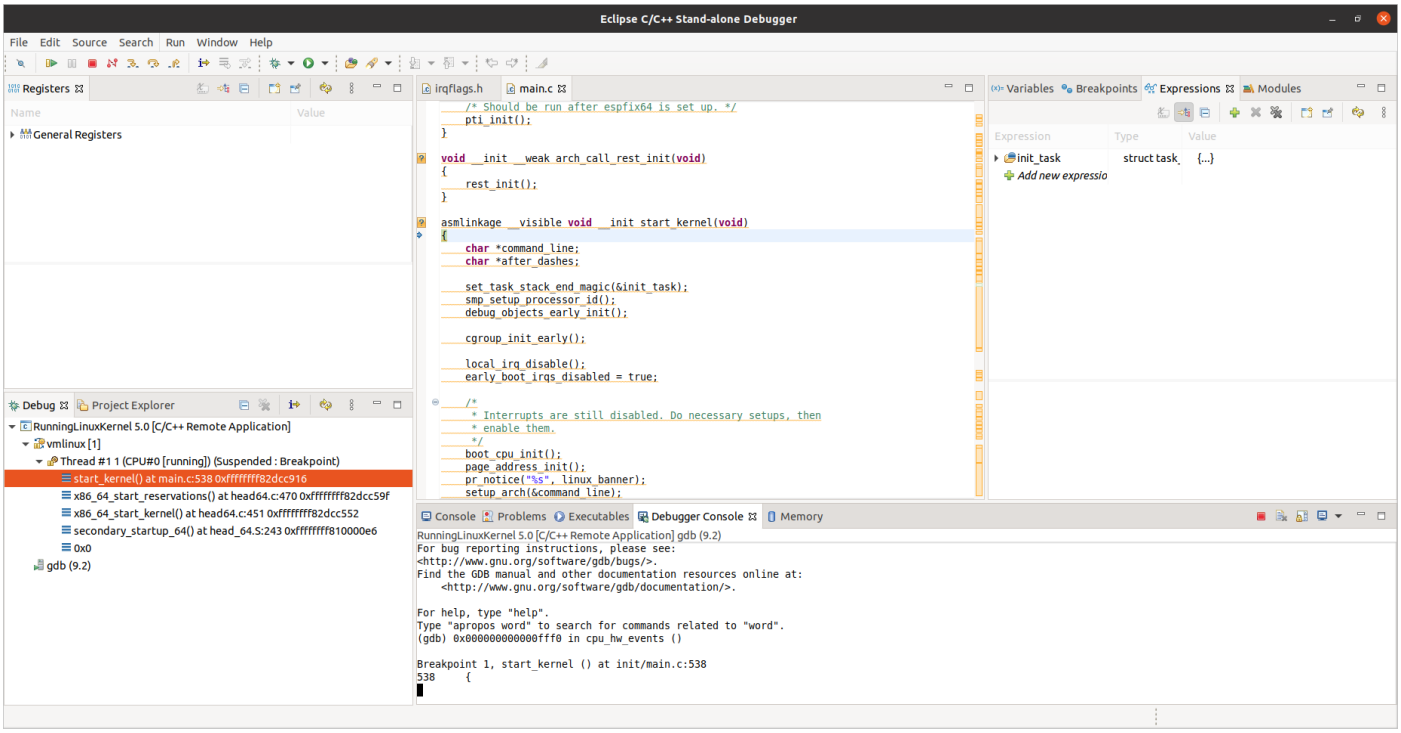
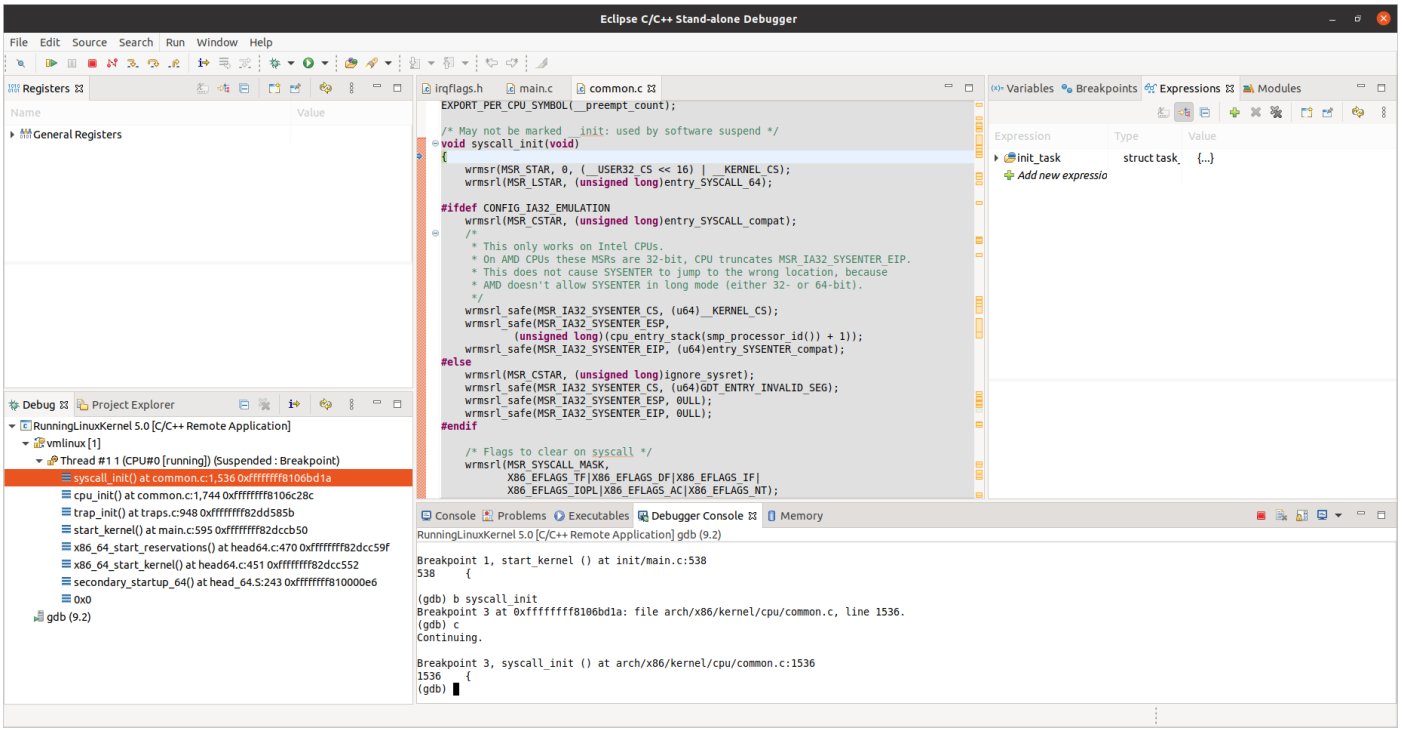
作業系統概論 hw1

學號: 408410113 姓名: 王 X 彥

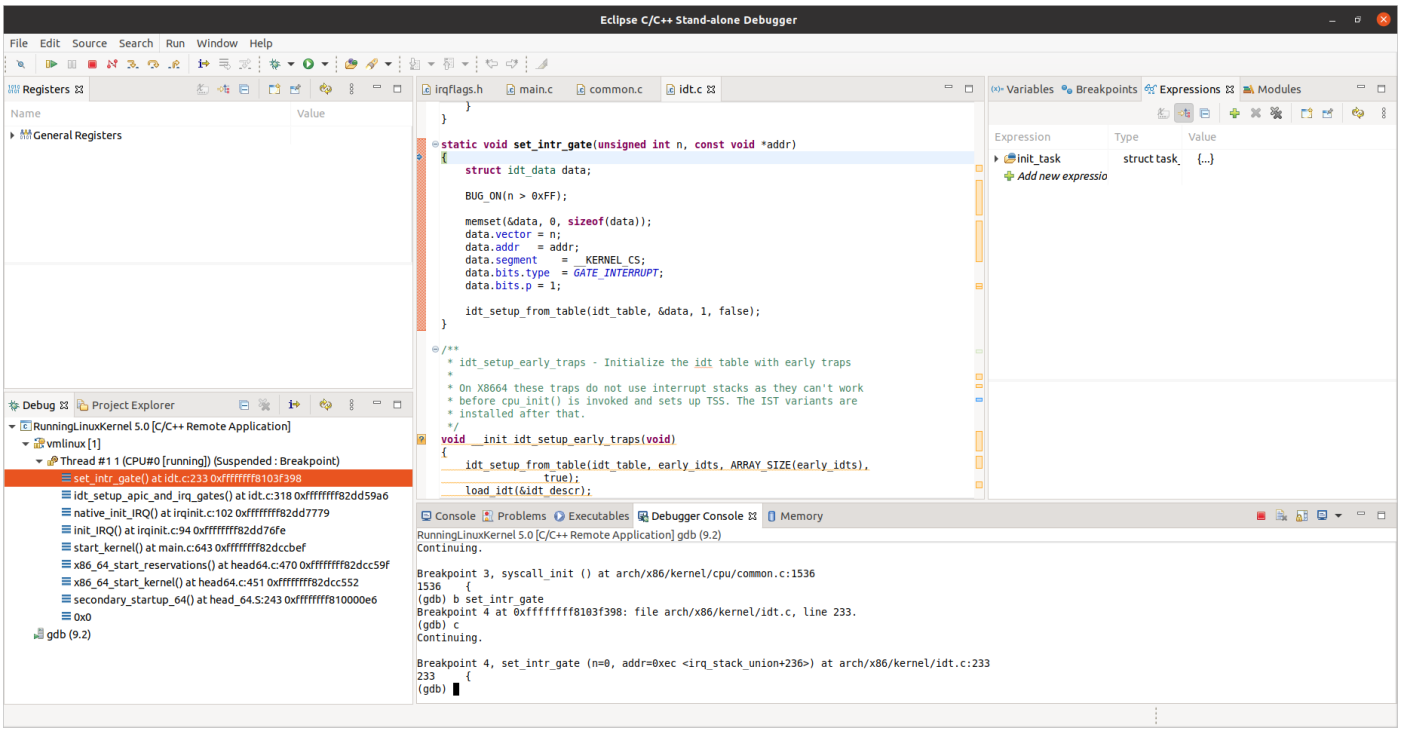
1. start_kernel



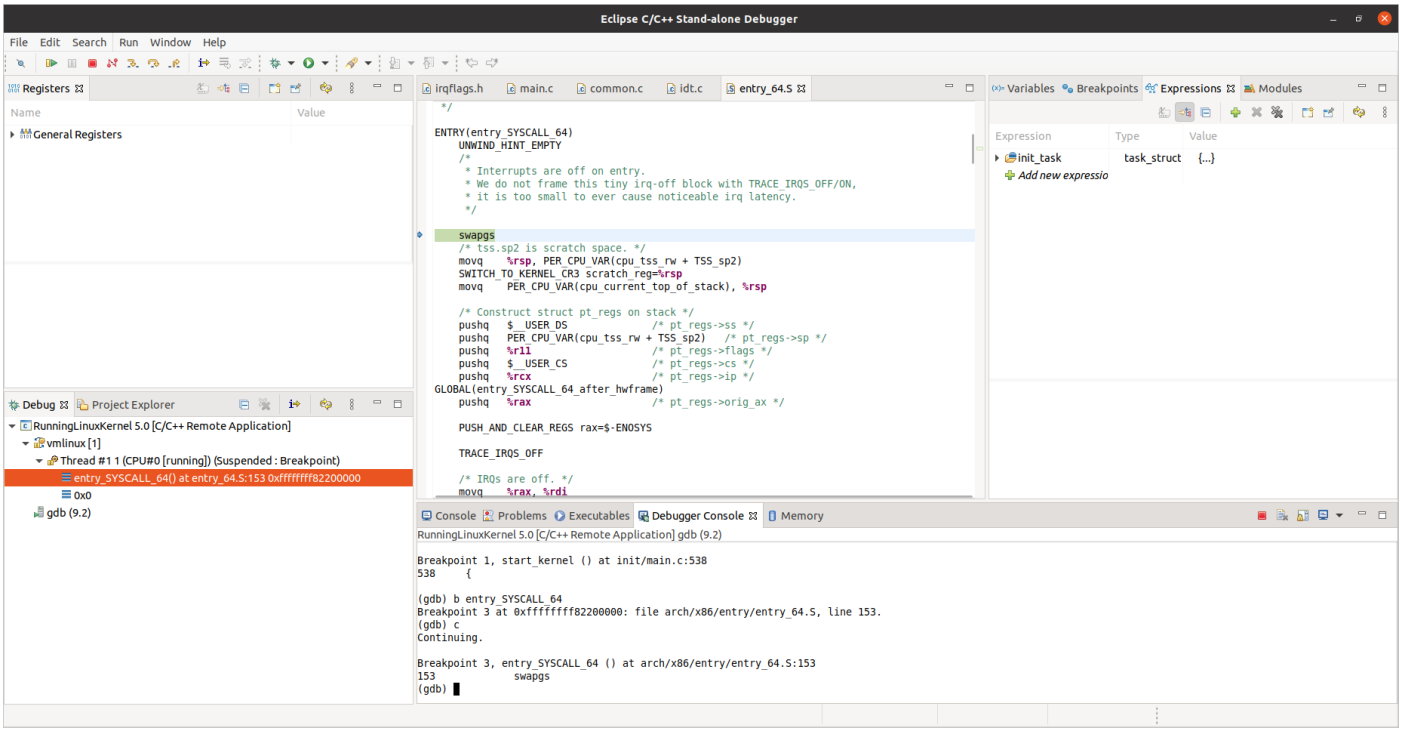
2. syscall_init



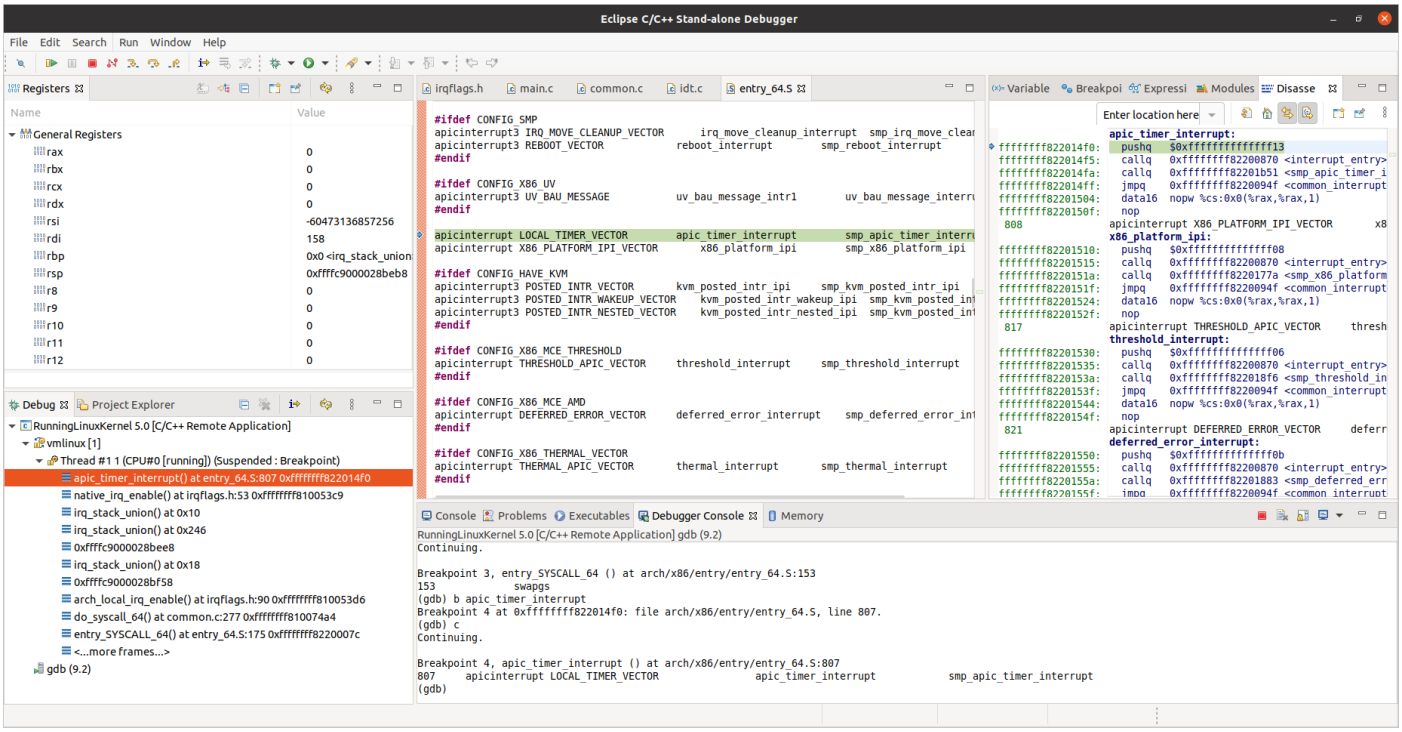
3. set_intr_gat



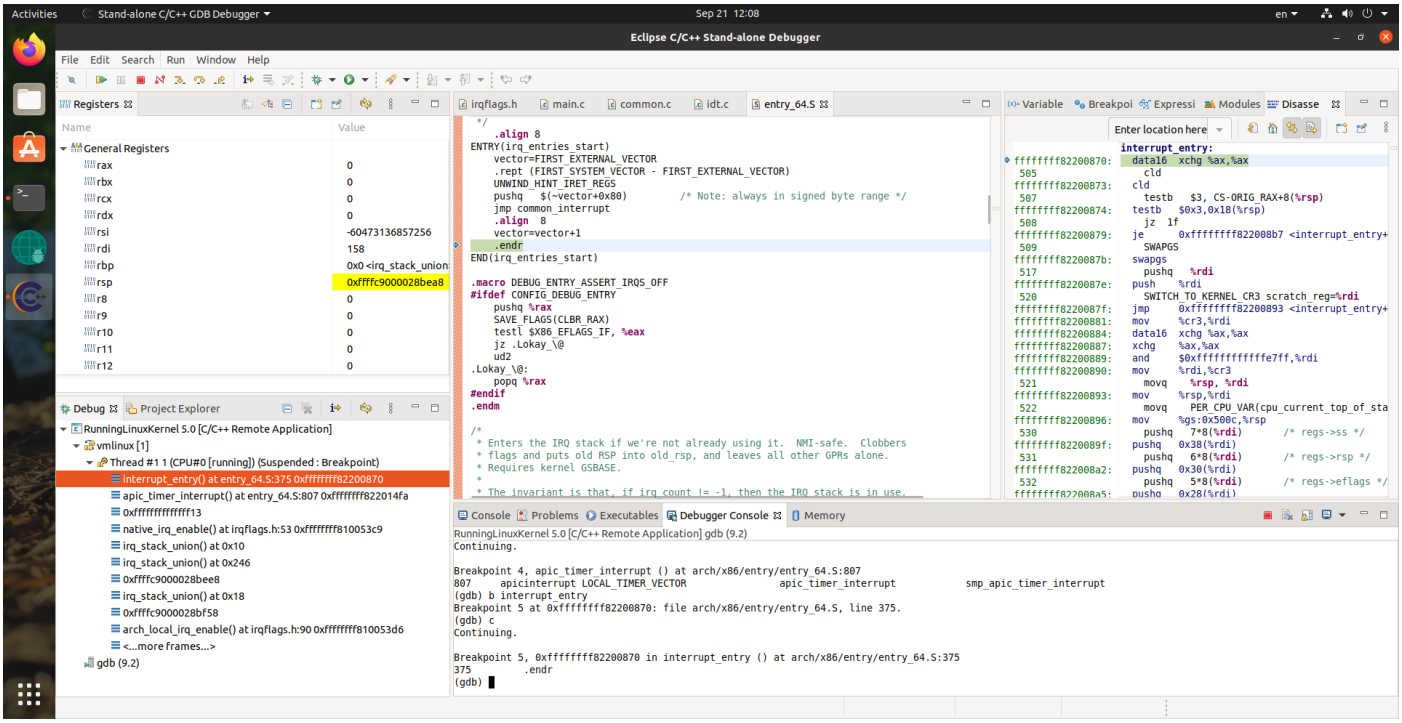
4. entry_SYSCALL_64



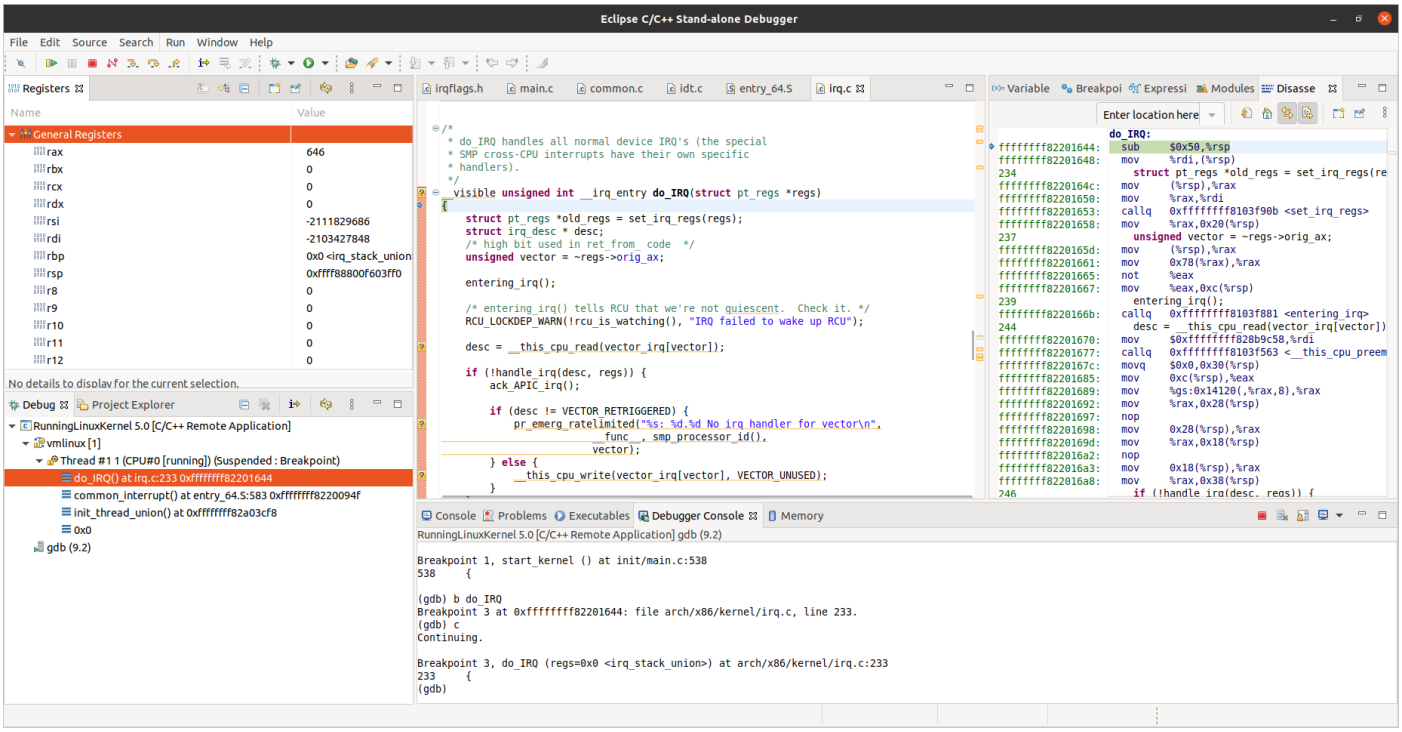
5. apic_timer_interrupt



6. interrupt_entry



7. do_IRQ


Eclipse C/C++ Stand-alone Debugger
File Edit Source Search Run Window Help
Registers
Name
Value
General Registers
rax 646
rbx 0
rcx 0
rdx 0
rsi -2111829686
rdi -2103427848
rbp 0x0 <irq_stack_union
rsp 0xffff88800f603ff0
r8 0
r9 0
r10 0
r11 0
r12 0
No details to display for the current selection.
Debug
Project Explorer
RunningLinuxKernel 5.0 [C/C++ Remote Application]
vmlinux [1]
Thread #1 1 (CPU#0 [running]) (Suspended : Breakpoint)
do_IRQ() at irq.c:233 0xffffffff82201644
common_interrupt() at entry_64.S:583 0xffffffff8220094f
init_thread_union() at 0xffffffff82a03cf8
0x0
gdb (9.2)
irqflags.h
main.c
common.c
idt.c
entry_64.S
irq.c
/*
* do_IRQ handles all normal device IRQ's (the special
* SMP cross-CPU interrupts have their own specific
* handlers).
*/
__visible unsigned int __irq_entry do_IRQ(struct pt_regs *regs)
{
struct pt_regs *old_regs = set_irq_regs(regs);
struct irq_desc * desc;
/* high bit used in ret_from_ code */
unsigned vector = ~regs->orig_ax;
entering_irq();
/* entering_irq() tells RCU that we're not quiescent. Check it. */
RCU_LOCKDEP_WARN(!rcu_is_watching(), "IRQ failed to wake up RCU");
desc = __this_cpu_read(vector_irq[vector]);
if (!handle_irq(desc, regs)) {
ack_APIC_irq();
if (desc != VECTOR_RETRIGGERED) {
pr_emerg_ratelimited("%s: %d.%d No irq handler for vector\n",
__func__, smp_processor_id(),
vector);
} else {
__this_cpu_write(vector_irq[vector], VECTOR_UNUSED);
}
Variable
Breakpoi
Expressi
Modules
Disasse
Enter location here
do_IRQ:
ffffffff82201644: sub $0x50,%rsp
ffffffff82201648: mov %rdi,(%rsp)
234 struct pt_regs *old_regs = set_irq_regs(re
ffffffff8220164c: mov (%rsp),%rax
ffffffff82201650: mov %rax,%rdi
ffffffff82201653: callq 0xffffffff8103f90b <set_irq_regs>
ffffffff82201658: mov %rax,0x20(%rsp)
237 unsigned vector = ~regs->orig_ax;
ffffffff8220165d: mov (%rsp),%rax
ffffffff82201661: mov 0x78(%rax),%rax
ffffffff82201665: not %eax
ffffffff82201667: mov %eax,0xc(%rsp)
239 entering_irq();
ffffffff8220166b: callq 0xffffffff8103f881 <entering_irq>
244 desc = __this_cpu_read(vector_irq[vector])
ffffffff82201670: mov $0xffffffff828b9c58,%rdi
ffffffff82201677: callq 0xffffffff8103f563 <__this_cpu_preem
ffffffff8220167c: movq $0x0,0x30(%rsp)
ffffffff82201685: mov 0xc(%rsp),%eax
ffffffff82201689: mov %gs:0x14120(,%rax,8),%rax
ffffffff82201692: mov %rax,0x28(%rsp)
ffffffff82201697: nop
ffffffff82201698: mov 0x28(%rsp),%rax
ffffffff8220169d: mov %rax,0x18(%rsp)
ffffffff822016a2: nop
ffffffff822016a3: mov 0x18(%rsp),%rax
ffffffff822016a8: mov %rax,0x38(%rsp)
246 if (!handle_irq(desc, regs)) {
Console
Problems
Executables
Debugger Console
Memory
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 0xffffffff82201644: 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)