Hands-on Session

- Create a Linux device driver

© 2015 Altera Corporation—**Public**

Exercise 8: Creating a Linux Device Driver

- We will use the red LEDs and pushbuttons
- The program increments the value displayed on the LEDs when a pushbutton is pressed

© 2015 Altera Corporation—**Public**

Kernel Module - Initialization

```
void * lwbridgebase;

static int __init intitialize_pushbutton_handler(void)
{
    // get the virtual addr that maps to 0xff200000
    lwbridgebase = ioremap_nocache(0xff200000, 0x200000);
    // Set LEDs to 0x200 (the leftmost LED will turn on)
    iowrite32(0x200,lwbridgebase);
    // Clear the PIO edgecapture register (clear any pending interrupt)
    iowrite32(0xf,lwbridgebase+0x5c);
    // Enable IRQ generation for the 4 buttons
    iowrite32(0xf,lwbridgebase+0x58);

    // Register the interrupt handler.
    return request_irq(73, (irq_handler_t)irq_handler,
              IRQF_SHARED, "pushbutton_irq_handler",
              (void *)(irq_handler));
}

module_init(intitialize_pushbutton_handler);
```


© 2015 Altera Corporation—Public

Kernel Module – Interrupt Handler

```
irq_handler_t irq_handler(int irq, void *dev_id, struct pt_regs *regs)
{
    // Increment the value on the LEDs
    iowrite32(ioread32(lwbridgebase)+1,lwbridgebase);
    // Clear the edgecapture register (clears current interrupt)
    iowrite32(0xf,lwbridgebase+0x5c);

    return (irq_handler_t) IRQ_HANDLED;
}
```


© 2015 Altera Corporation—Public

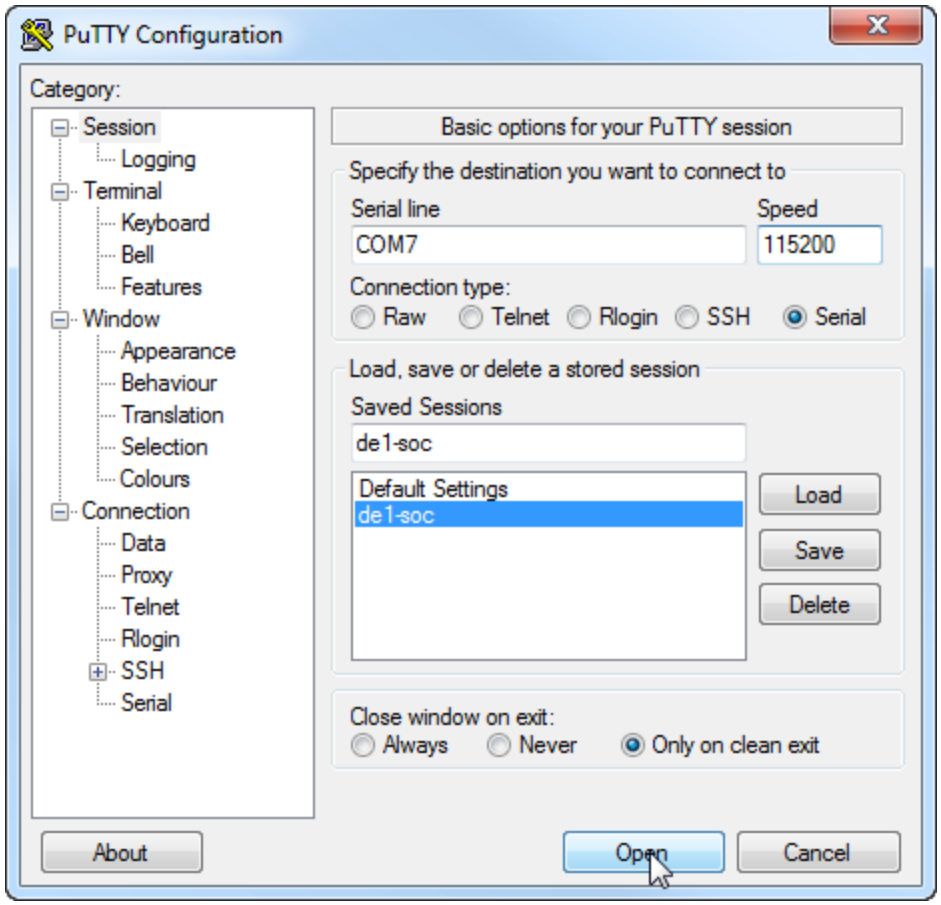
Kernel Module – Exit Routine

```
static void __exit cleanup_pushbutton_handler(void)
{
    // Turn off LEDs and de-register irq handler
    iowrite32(0x0,lwbridgebase);
    free_irq(73, (void*) irq_handler);
}

module_exit(cleanup_pushbutton_handler);
```

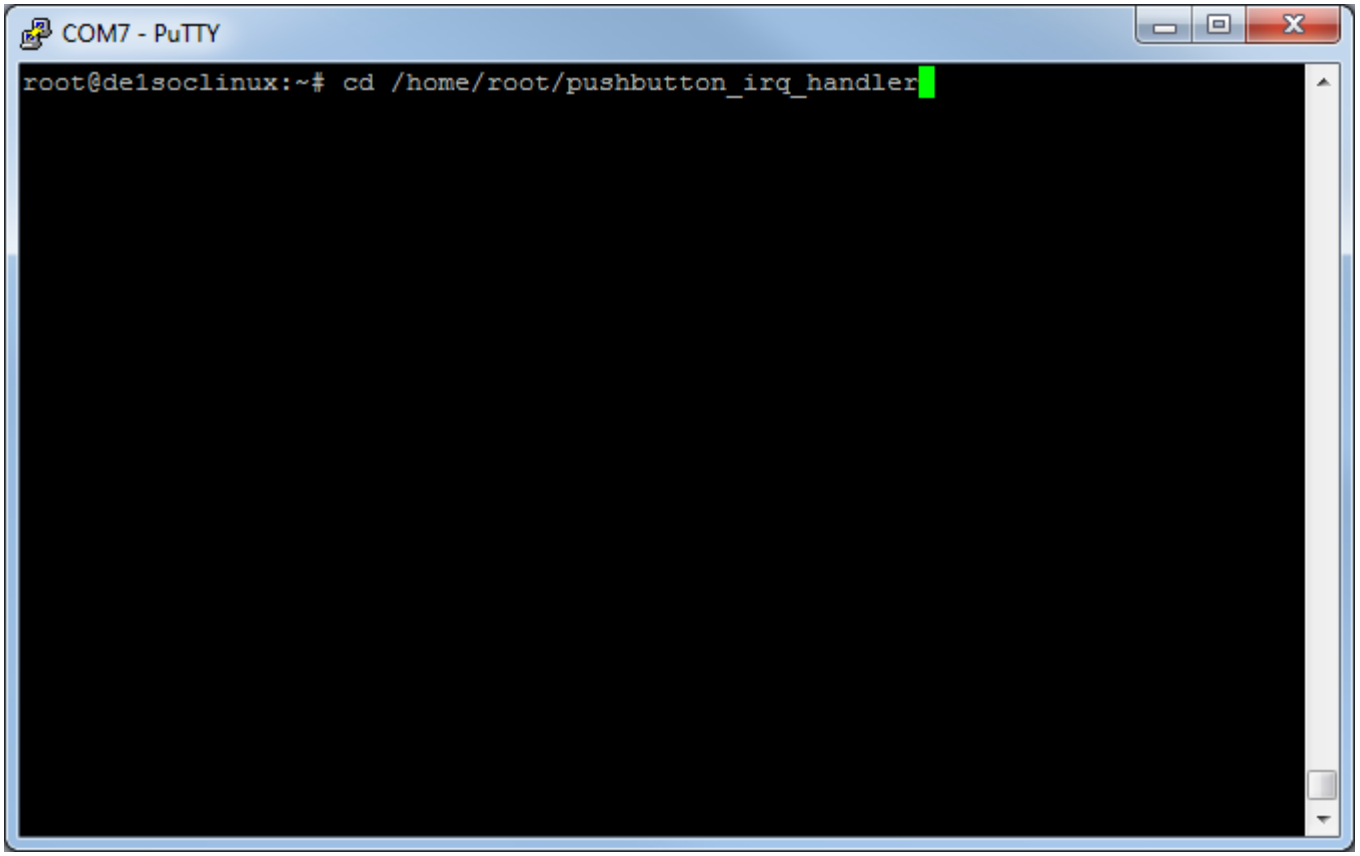

© 2015 Altera Corporation—Public

Step 1: Open the Connection to the DE1-SoC using Putty



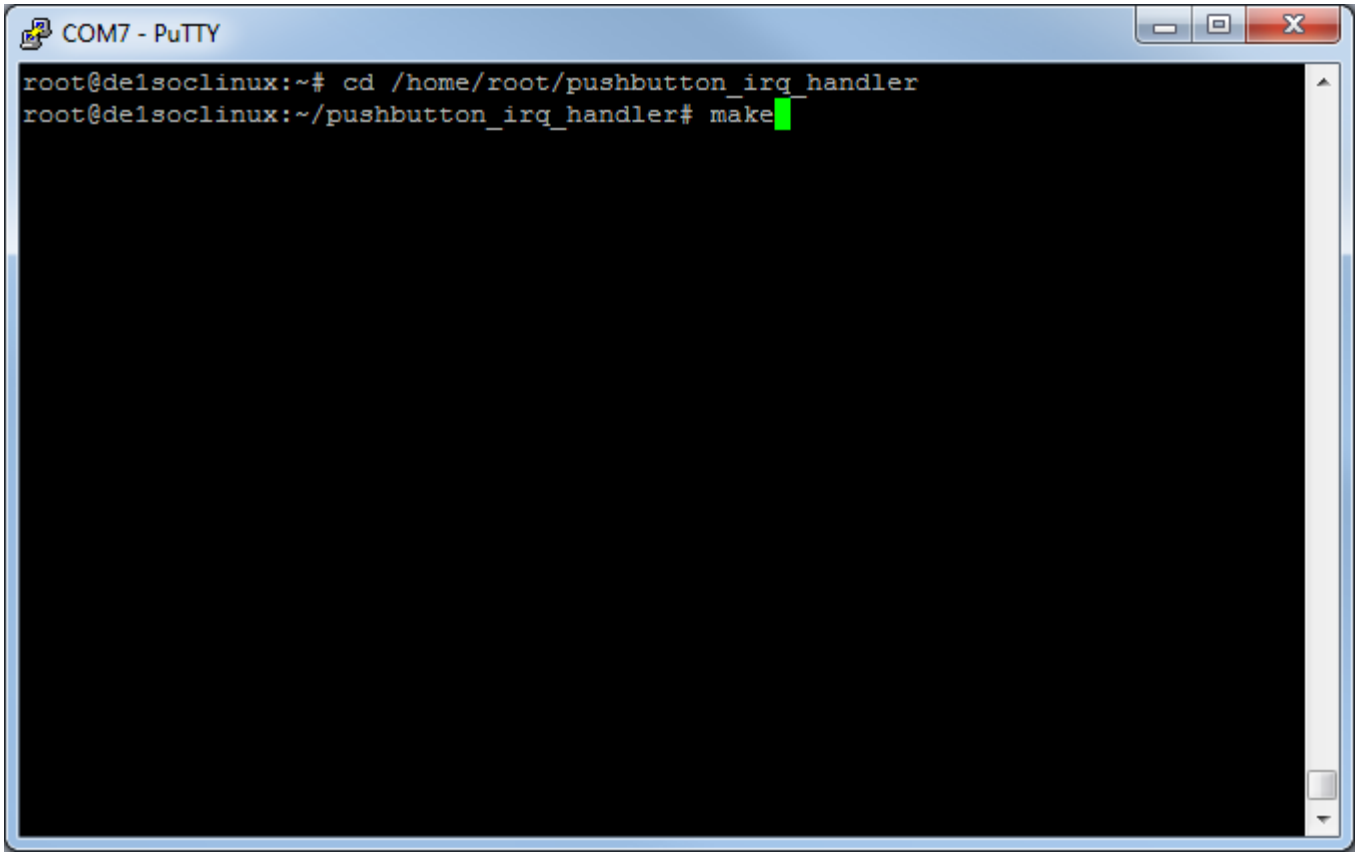
© 2015 Altera Corporation—**Public**

Step 2: Change to the Example Directory



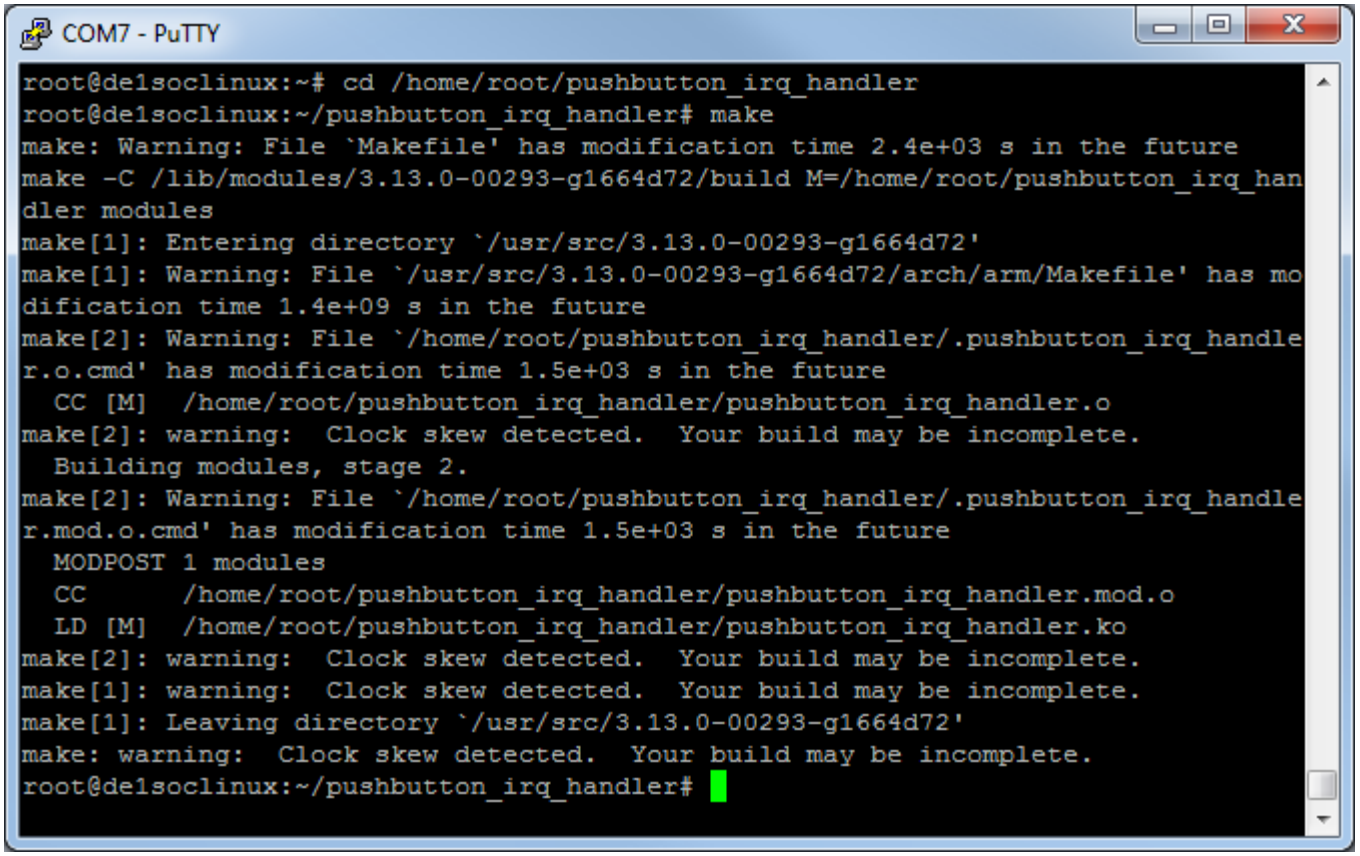
© 2015 Altera Corporation—**Public**

Step 3: Compile the Sample Program



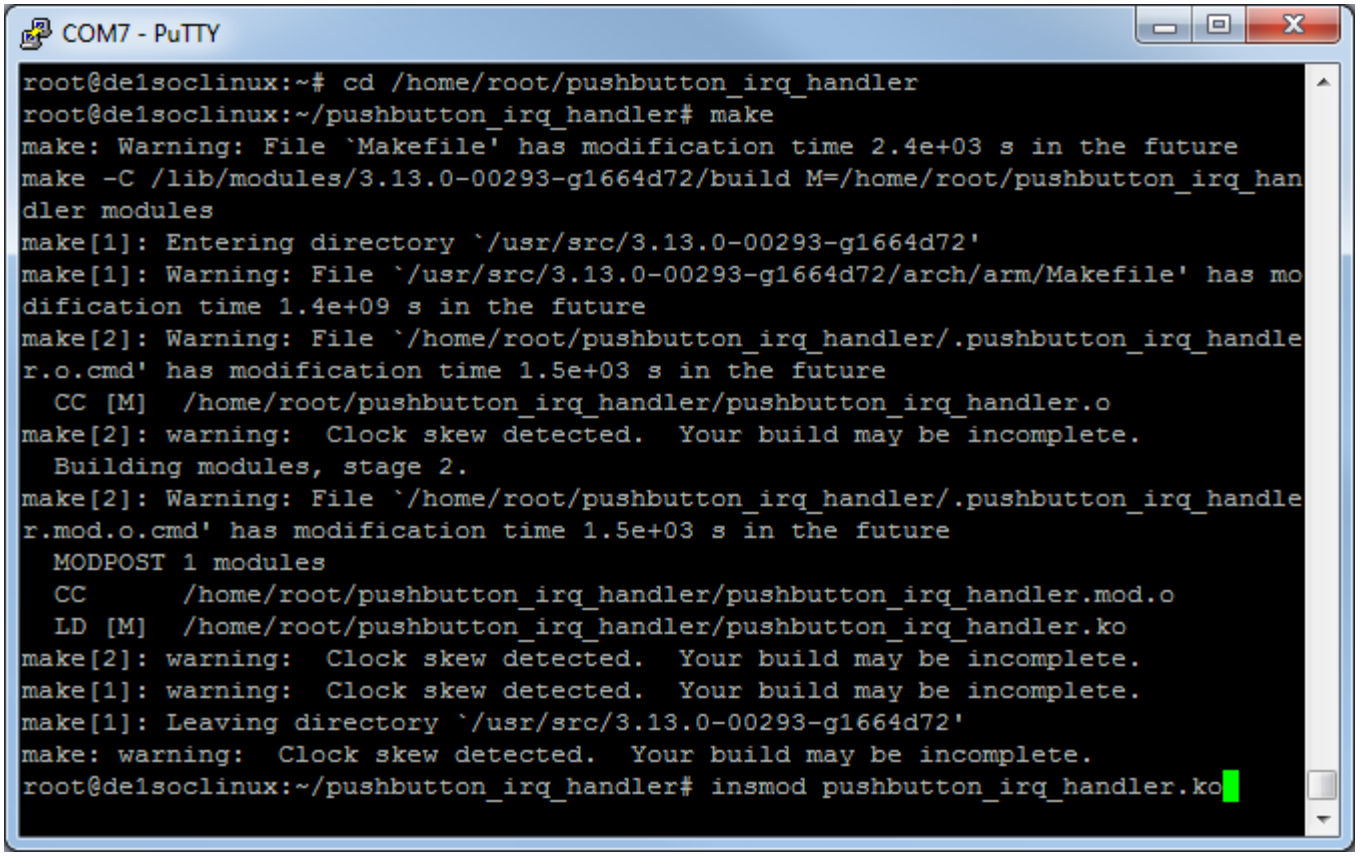
© 2015 Altera Corporation—**Public**

Step 4: Compilation Complete



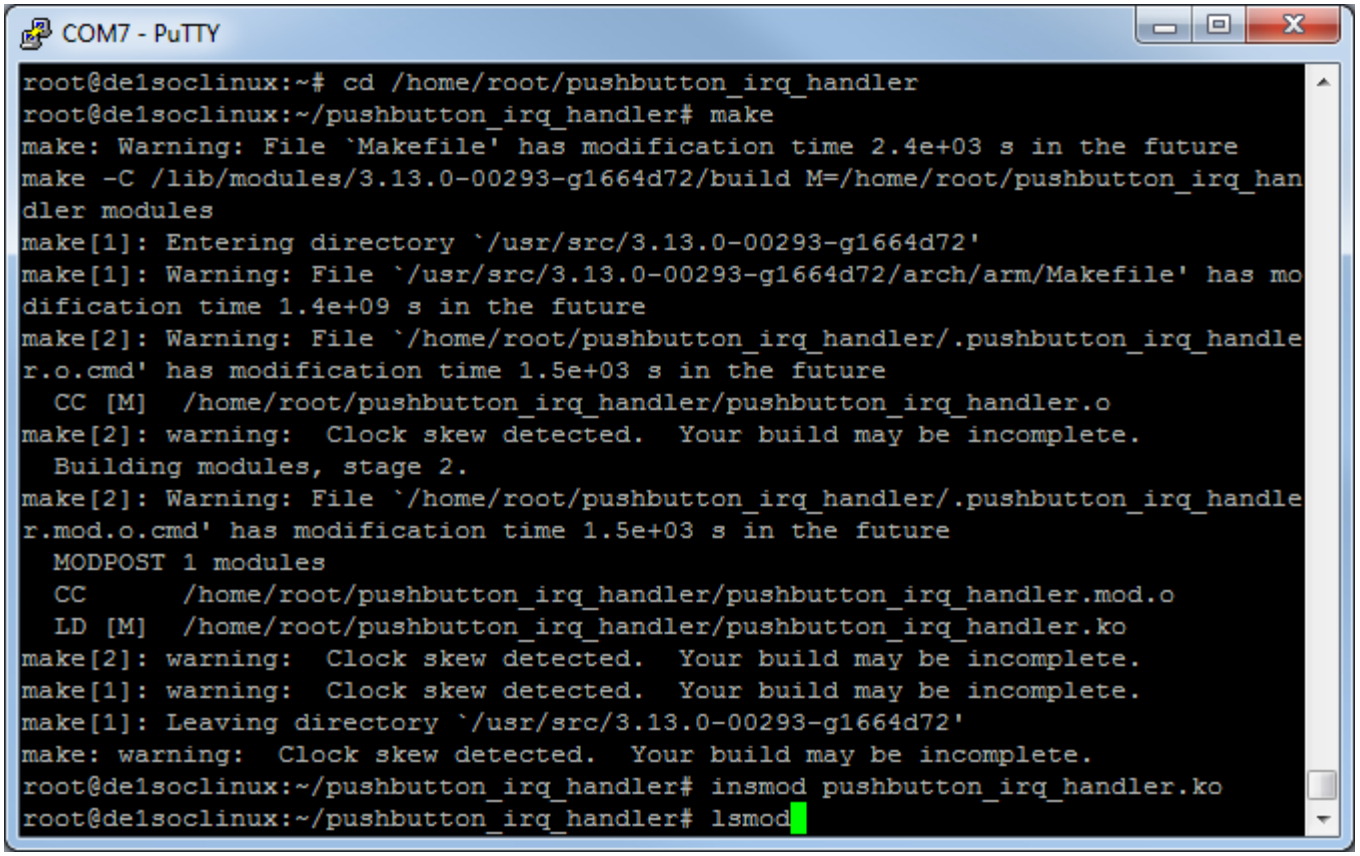
© 2015 Altera Corporation—**Public**

Step 5: Insert Newly Compiled Kernel Module into Linux



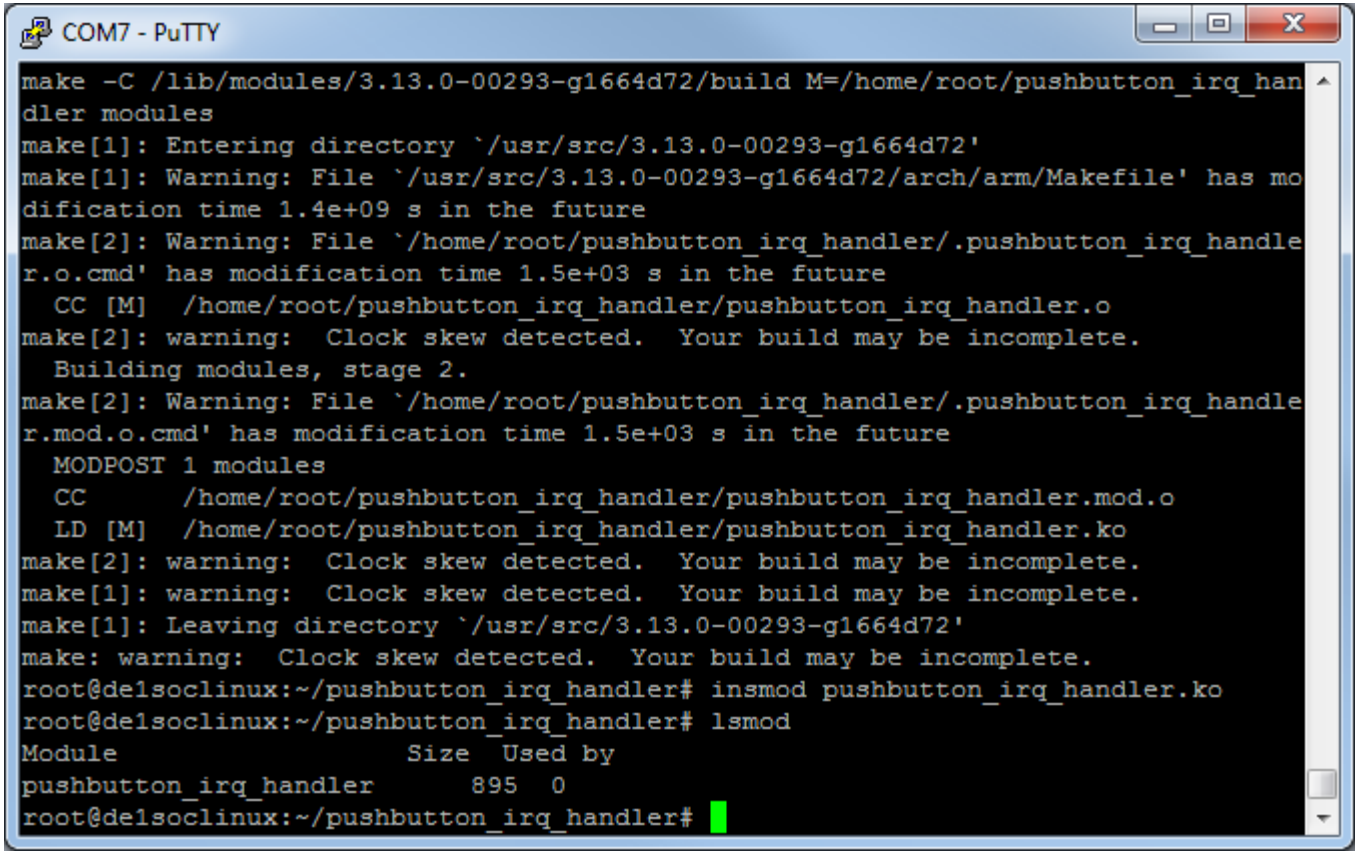
© 2015 Altera Corporation—Public

Step 6: Run ‘lsmod’ to See All Loaded Kernel Modules



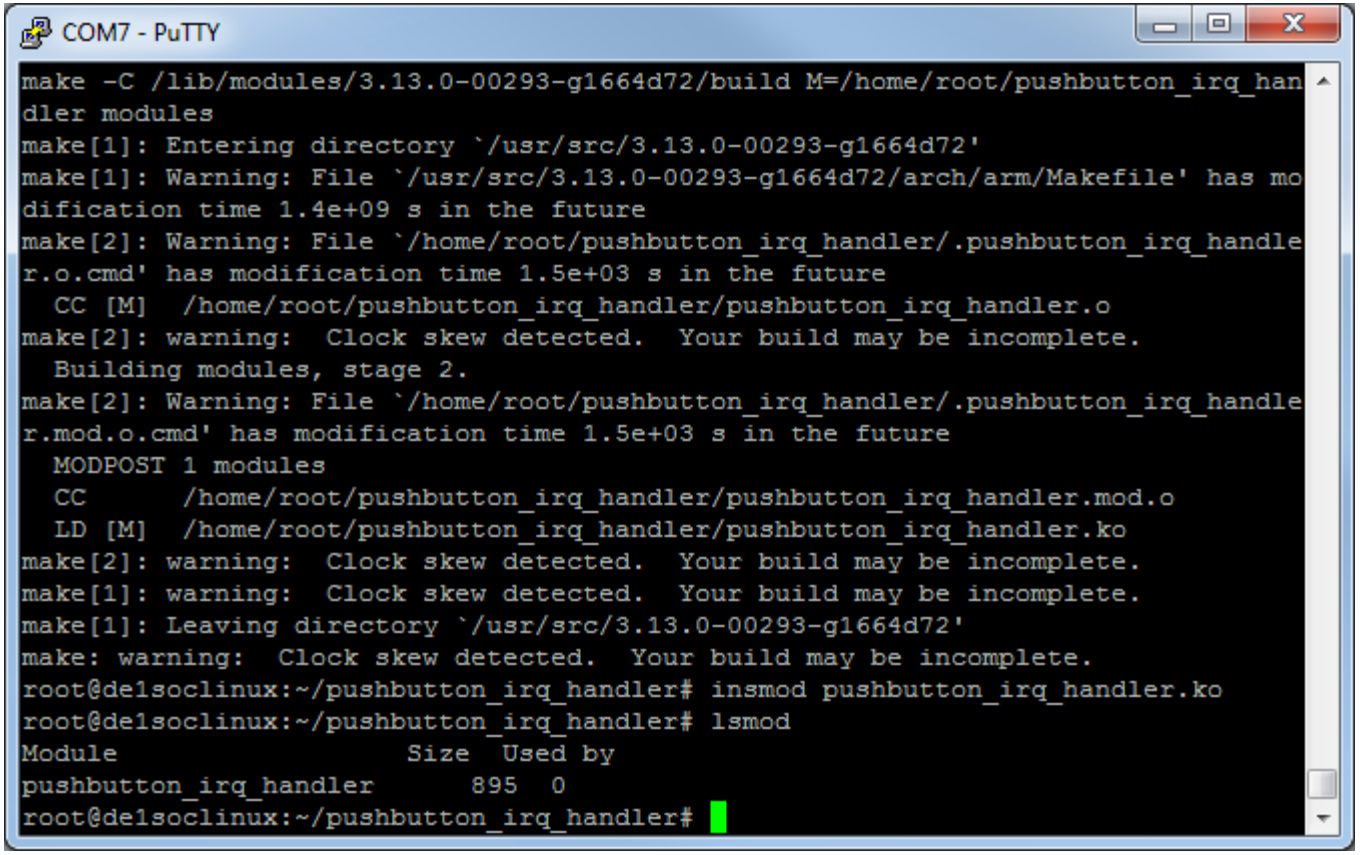
© 2015 Altera Corporation—**Public**

Step 6: The Pushbutton IRQ Handler was Loaded



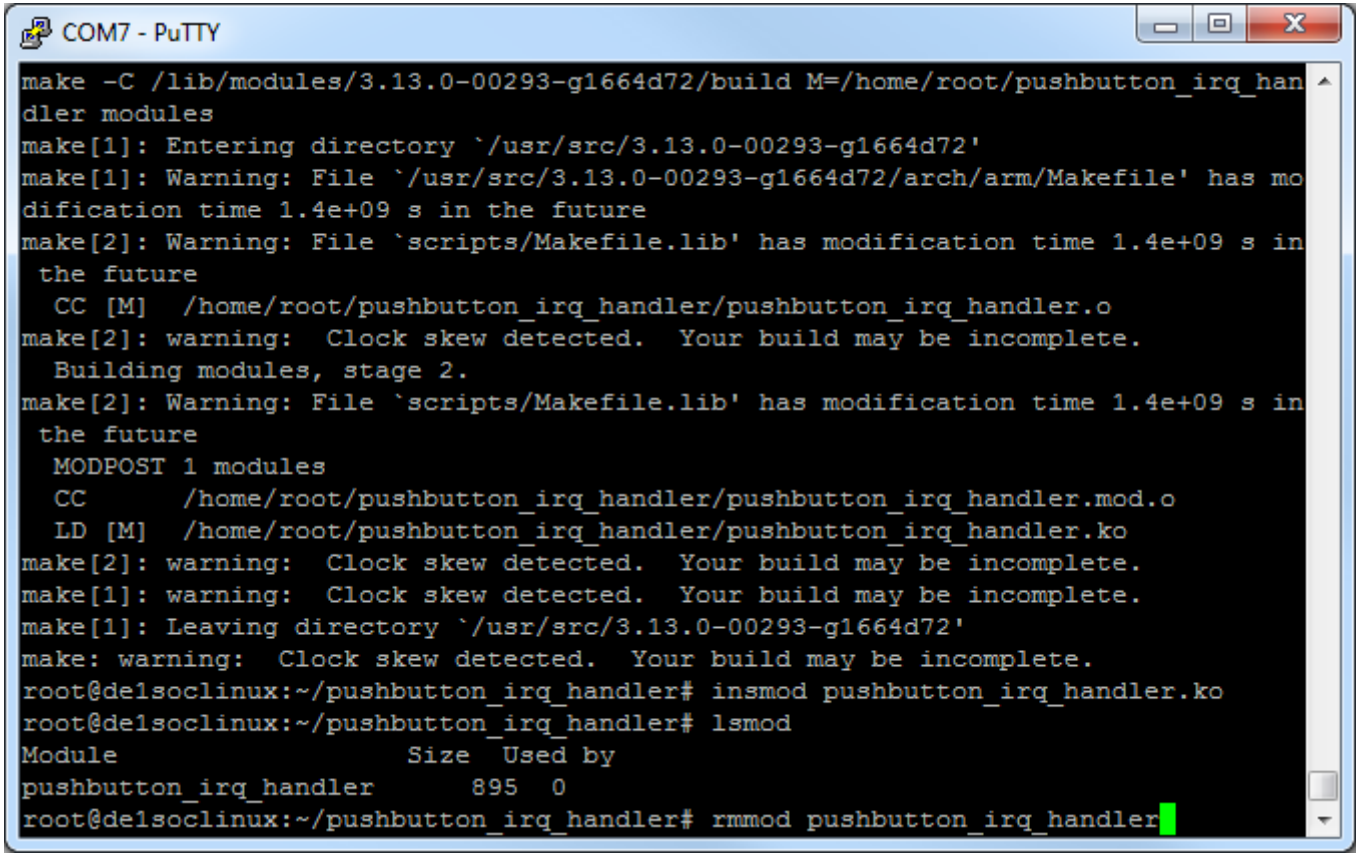
© 2015 Altera Corporation—Public

Step 7: Try Toggling the Pushbuttons



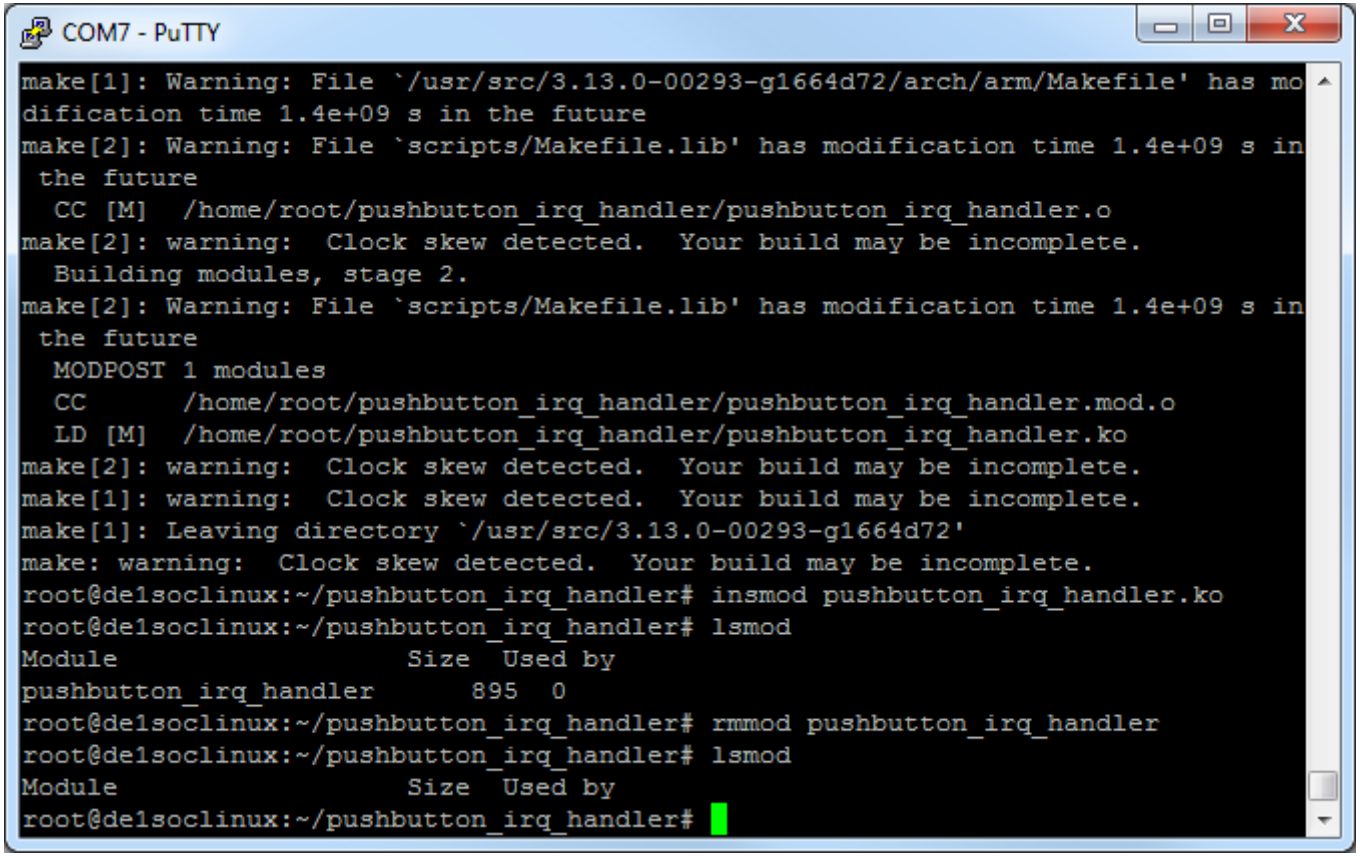
© 2015 Altera Corporation—Public

Step 8: Remove the Kernel Module



© 2015 Altera Corporation—Public

Step 9: Run ‘lsmod’ to See All Loaded Kernel Modules



© 2015 Altera Corporation—Public

Step 10: Examine the 'pushbutton_irq_handler.c' file

U:\External_Documentation\Prof_Course_2016_Korea\exercise9\pushbutton_irq_handler.c - Notepad++

pushbutton_irq_handler.c

```
1   #include <linux/kernel.h>
2   #include <linux/module.h>
3   #include <linux/init.h>
4   #include <linux/interrupt.h>
5   #include <asm/io.h>
6
7   MODULE_LICENSE("GPL");
8   MODULE_AUTHOR("Altera University Program");
9   MODULE_DESCRIPTION("DE1SoC Pushbutton Interrupt Handler");
10
11  void * lwbridgebase;
12
13  irq_handler_t irq_handler(int irq, void *dev_id, struct pt_regs *regs)
14  {
15      // Read value of switches, write it to LEDs
16      iowrite32(ioread32(lwbridgebase+0x40),lwbridgebase);
17      // Clear the edgecapture register (clears current interrupt)
18      iowrite32(0xf,lwbridgebase+0x5c);
19
20      return (irq_handler_t) IRQ_HANDLED;
21  }
22
23  static int __init intitialize_pushbutton_handler(void)
24  {
25      // get the virtual addr that maps to 0xff200000
26      lwbridgebase = ioremap_nocache(0xff200000, 0x200000);
27      // Set LEDs to 0
28      iowrite32(0x0,lwbridgebase);
29      // Clear the PIO edgecapture register (clear any pending interrupt)
30      iowrite32(0xf,lwbridgebase+0x5c);
```

C so length : 1528 lines : 49 | Ln : 1 Col : 1 Sel : 0 | 0 | Dos\Windows | UTF-8 w/o BOM | INS

© 2015 Altera Corporation—Public