Interrupts

What to Expect?

- W's & How's of Interrupts?
- The IRQ
- Interrupt Handling & Handlers
- Soft IRQs

W's of an Interrupt?

- What is an Interrupt?
  - Intervention to get Attention
  - Here, from the Devices to the CPU
- Why is an Interrupt required?
  - Mismatch of Device & CPU speeds
  - Get overall better efficiency & latency

How does interrupt work?

- Interrupt is a Hardware thing
- It needs Physical connection
- In Processors
  - Typically have one or two interrupt lines
  - Handled by a Interrupt Controller, called PIC
  - And PIC informs the Processor through its one line
- In Micro-controllers
  - Each GPIO can act as an interrupt line
  - Interrupt Controller is in-built
- In both cases, the CPU then decodes the IRQ
- And calls the corresponding registered interrupt handler

Interrupt ReQuest (IRQ)

- Number derived by CPU & Board
- Programmed or Hard Coded with Interrupt Controller
- Hence, one key component of LSPs
- In x86
  - CPU Specific: 0x00 to 0x1F (Hard coded)
  - Board Specific: 0x20 to 0xFF
    - Header: <asm/interrupt.h> → <asm/irq.h> → irq*.h
- In Microcontrollers
  - Depends on the controller
    - Header: <asm/interrupt.h> → <asm/irq.h> → irq*.h

Programming Interface

- Header: <asm/interrupt.h>
- Type:
  - typedef irqreturn_t (*irq_handler_t)(int, void *);
- APIs
  - int request_irq(unsigned int irq, irq_handler_t handler, unsigned long flags, const char *name, void *dev_id);
  - void free_irq(unsigned int irq, void *dev_id);
  - int can_request_irq(irq, flags);
- Flags
  - IRQF_TRIGGER_RISING, ..., IRQF_TRIGGER_HIGH, ...
  - IRQF_SAMPLE_RANDOM
  - IRQF_SHARED, ...

IRQ Handler Do's & Don'ts

- No sleeping – direct or indirect
  - Example: schedule_timeout, input_register_device
- No mutexes
  - Rather use spin_locks, if you must
- Can't exchange data with User Space
  - In their own context. current is invalid
- If lot to be done, break it into two
  - Top Half & Bottom Half
- Need not be re-entrant
- May get interrupted by higher priority interrupt handlers

The Additional Info

- IRQ Control
  - enable_irq(irq);
  - disable_irq(irq);
- IRQ Handler returns
  - IRQ_NONE
  - IRQ_HANDLED
- Check for execution state
  - in_interrupt();
- Synchronous interrupts, treated alike
  - Exceptions to report grave runtime errors
  - Software interrupts such as the int 0x80 used for system calls in x86 architecture

Message Signaled Interrupts

- Specifically for PCI Devices
- Advantages
  - No sharing, No sync issues, More interrupts
- Modes: MSI or MSI-X (only one at a time)
- MSI (since PCI 2.2)
  - Special Address: PCI config space
  - Interrupts / Device: Upto 32 in powers of 2
- MSI-X (since PCI 3.0)
  - Special Address: Bus address
  - Interrupts / Device: Sparse & upto 2048

MSI/MSI-X APIs

- MSI
  - int pci_enable_msi(strcut pci_dev *);
  - int pci_enable_msi_block(strcut pci_dev *, int cnt);
  - void pci_disable_msi(strcut pci_dev *);
- MSI-X
  - int pci_enable_msix(strcut pci_dev *, struct msix_entry *, int nvec);
    - struct msix_entry

      { u16 vector /* allocated irq */, entry; };
  - void pci_disable_msix(strcut pci_dev *);

Soft IRQs

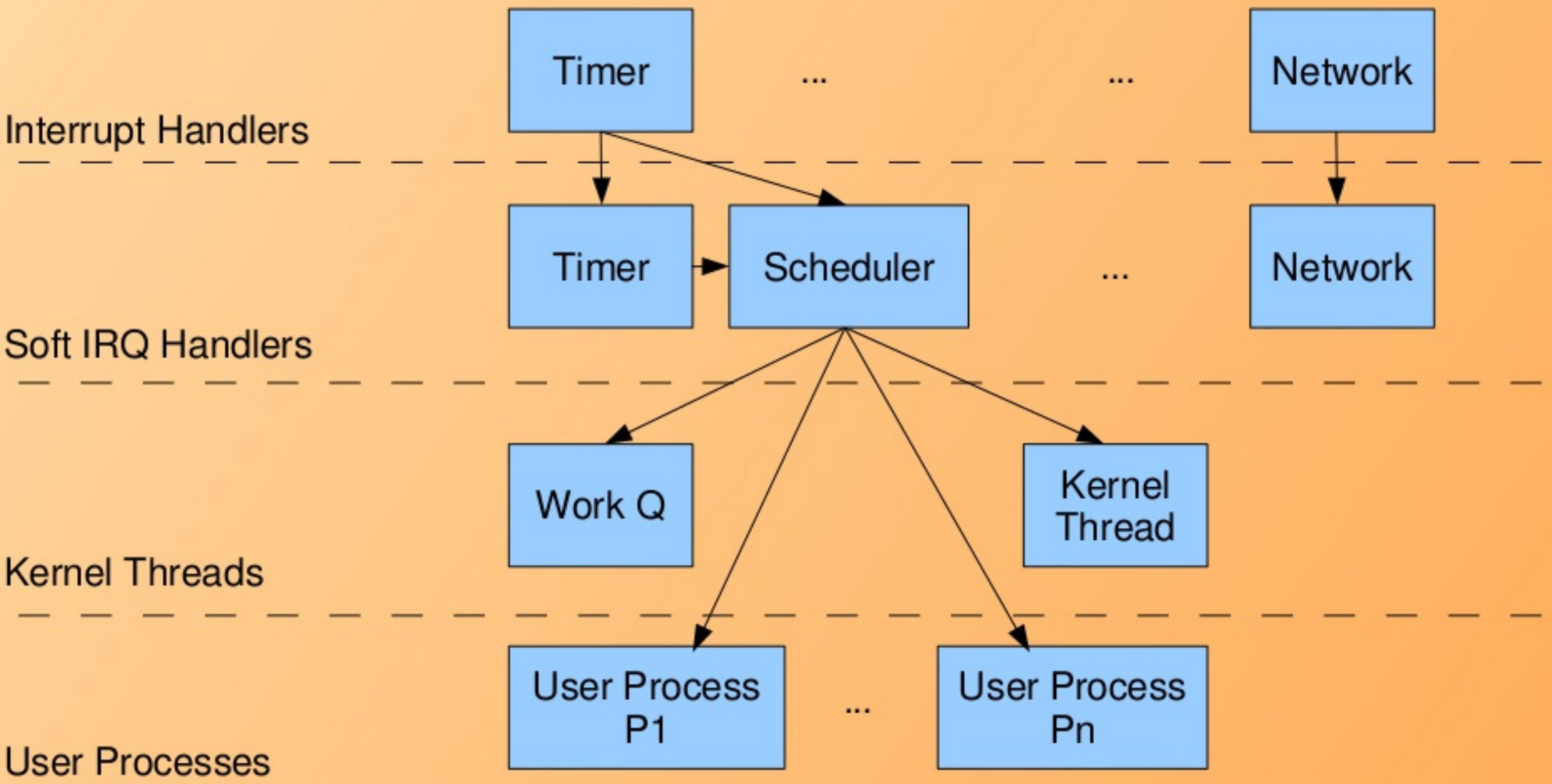
What is a Soft IRQ?

- Basic Bottom Half
- Synchronous Interrupt
- Have strong locking requirements
- Used for the performance sensitive subsystems
- Not same as Software Interrupt

Typical Soft IRQs

- Timer
- Network
- Block
- Tasklet
- Scheduler
- High Resolution Timer

Linux Execution Model



Programming

- Header: <linux/interrupt.h>
- APIs
  - void open_softirq(int nr, void (*action)(struct softirq_action *));
  - void raise_softirq(unsigned int nr);

Top & Bottom Halves

- Top Half
  - Registered through request_irq
- Bottom Half
  - Tasklet
    - Soft IRQ Context
    - Fast & Atomic
    - Only different tasklets can run on different CPUs
  - Work Queue
    - Special Kernel Process Context
    - Allowed to sleep
    - Can run simultaneously on different CPUs

What all have we learnt?

- W's & How's of Interrupts?
- The IRQ
- Interrupt Handling & Handlers
- Message Signalled Interrupts
- Soft IRQs: The Bottom Halves

Any Queries?