Linux Kernel Programming

Interrupted Interrupts Handlers & Shared Data

Pierre Olivier

Systems Software Research Group @ Virginia Tech

February 28, 2017

Virginia Tech

- In these slides, **handler == interrupt handler == top-half == ISR**

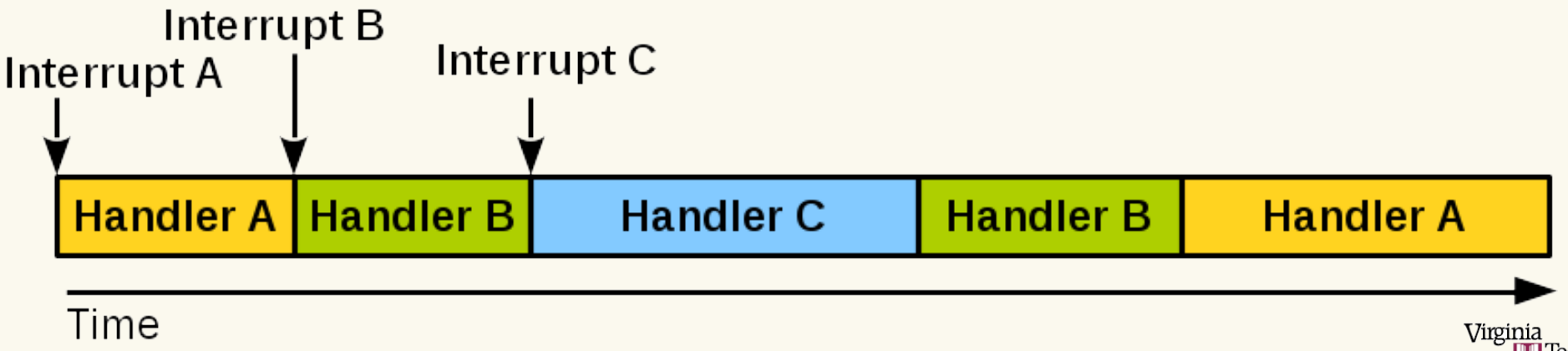
Virginia Tech

Outline

Virginia Tech

What can interrupt an interrupt handler?

- **What can interrupt an interrupt handler?**
  - Kernel **Preemption *cannot*** interrupt a handler
  - The interrupt (number) the handler is processing cannot interrupt it
    - This interrupt is disabled while its handler executes
- **Another interrupt can interrupt the currently executing handler**
  - (If this handler is configured to run with other interrupts enabled, which is generally the case)



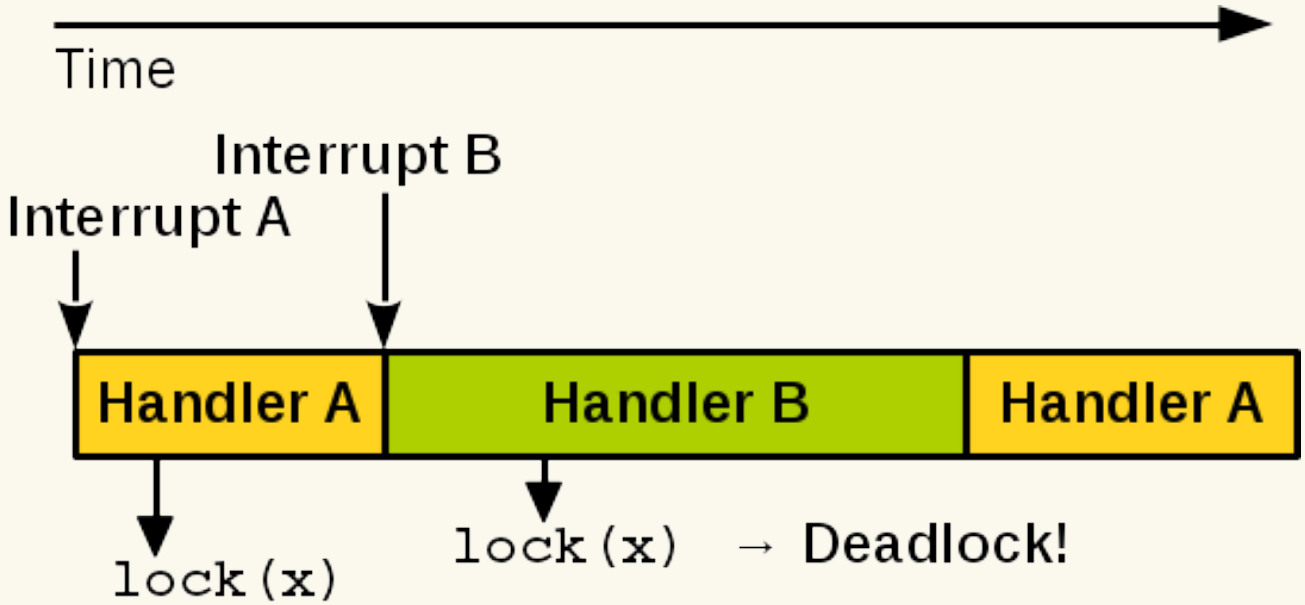
Virginia Tech

Outline

Virginia Tech

Handlers sharing data

- **If data is shared between two handlers, there is a risk of deadlock:**



- Solution: disabling interrupts before lock acquisition
  - We will see in the slides on synchronization the function `spin_lock_irqsave(lock, flags)`
    - As opposed to `spin_lock(lock)`

Virginia Tech

Outline

Virginia Tech

What about multicores?

- **An interrupt handler does not have to be thread-safe**
  - The corresponding interrupt is masked on all cores while the handler executes
- **Shared data protection against concurrent accesses**:
  - If data is shared *with another interrupt handler*, **use a lock** (and also disable interrupts on the local core, see previous slide example)
  - If data is shared with *another entity* (ex a kernel thread), just **use a lock**

Virginia Tech