

Interrupts (SW/HW)

Forklar de forskellige interrupttyper?

- **Software interrupts:**
 - **Event-driven code:** Kommer typisk fra **programmer** i applikationslageret (GUI)
 - **Exceptions:** Specielle SW interrupts genereret af **OS** eller **CPU** ved kritiske fejl såsom page fault
- **Hardware interrupts:**
 - **Internal:** Interrupts kommer af interne devices (**UART**)
 - **External:** Fra eksterne enheder, f.eks. tastatur eller mus
- **Level-triggered interrupts:**
 - Aktivt så længe IRQ er aktiv
 -
 -
 -
- **Edge-triggered interrupts:**
 -
 -
 -
 -
- **Shared interrupts**
 -
 -
 -
 -

Hvad er de typiske opgaver for en interrupt service routine?

-
-
-
-
-
-

Hvordan virker interrupthåndtering i Linux?

-
-
-
-
-

Hvilke funk. skal vi benytte i en driver for at understøtte interrupts?

- **request_irq:** Laver en interrupt handler.
- **free_irq:** Nedlægger handleren igen.
- **wake_up_interruptable:** Bliver kaldt når der er et interrupt.
- **wait_event_interruptable:** Sover indtil der er givet event i gennem en waitqueue (read-metode).
 - **waitqueue:** til at videregive events fra wake_up til wait

Hvordan kan vi optimere vores ISR?

- Interrupthandlers skal blive hurtige færdige for ikke at holde interruptet blokeret for længe.
- Linux løser dette ved at dele handleren op i to:
 - **Top half:** Routinen der svarer på interruptet (den der oprettes af request_irq)
 - **Bottom half:** Routine der skeduleres af top half for at blive eksekveret af bottom half. Dette sikrer at interrupts rutiner ikke afbryder registreringen af interrupts