

## Fallbeispiel: Linux

---

- Linux nutzt auch eine Tabelle mit Referenzen auf die Interrupt-Handler (ISR)
- Jeder Interrupt-Request wird auf eine Interrupt-Nummer (= Index in der Tabelle) abgebildet
- Meist wird in der ISR nur ein **Tasklet** erzeugt
- Tasklets dienen der schnellen Behandlung von Interrupts (ähnlich dem Windows-DPC-Mechanismus)

# Fallbeispiel: Linux, Interrupt-Vektor-Tabelle

Die Interrupt-Vektor-Tabelle ist im System wie folgt definiert:

```
extern irq_desc_t irq_desc [NR_IRQS];
```

Aufbau eines Tabelleneintrags:

```
typedef struct {  
    unsigned int status;           // IRQ-Status  
    hw_irq_controller *handler;    // Zeiger auf verantwortlichen  
                                   // (Hardware-)Controller  
    struct irqaction *action;      // Zeiger auf Action-Liste  
    unsigned int depth;            // Spezielles Feld zum  
                                   // Aktivieren und  
                                   // Deaktivieren des IRQ  
} ____cacheline_aligned irq_desc_t;
```

## Fallbeispiel: Linux, Action-Liste

- action = Action-Descriptor, Struktur mit Verweis auf eigentliche ISR
- Verkettete Liste

```
// vereinfacht
struct irqaction {

    // Verweis auf Interrupt-Service-Routine
    void (*handler)(int, void *, struct pt_regs *);

    unsigned long flags;           // Eigenschaften des Interrupt-Handlers
    const char *name;             // Name des Interrupt-Handlers
    void *dev_id;                 // Eindeutige Identifikation des
                                // Interrupt-Handlers
    struct irqaction *next;       // Verweis auf nächsten Eintrag in der
                                // Action-Liste bei shared Interrupts
};
```

# Fallbeispiel: Linux, Datenstrukturen im Kernel

