

Betriebssysteme (BS)

VL 5.4 – Unterbrechungen, SoftIRQs – Verwandte Konzepte

Volkmar Sieh / Daniel Lohmann

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

Friedrich-Alexander-Universität
Erlangen Nürnberg

WS 20 – 23. November 2020



https://www4.cs.fau.de/Lehre/WS20/V_BS

Agenda

Einordnung

Beispiele

Aufteilung Interrupt-Bearbeitung

Implementierung

SoftIRQs

SoftIRQ-Beispiel

Verwandte Konzepte

Zusammenfassung



Idee:

- Interrupt-Handler liest Daten aus Gerät in Puffer und **setzt Bit**
- vor **Rückkehr in den User-Mode** werden Puffer ausgelesen und die Daten verarbeitet, **wenn Bit gesetzt**

(Linux)



Idee:

- `while (pending) { pending = false; handler(); }`-Schleife wird nur begrenzte Anzahl von Malen durchlaufen.
- Danach wird „`ksoftirqd`“ (**Kernel-Thread**) aufgeweckt, der SoftIRQ-Abarbeitung übernimmt.

Vorteil: Interrupts können CPUs nicht monopolisieren.
(Linux)



Idee:

Hard-Interrupt-Handler triggert nicht (per Nummer) die Ausführung einer bestimmten Methode, sondern fügt eine **beliebige Methode** in eine **Liste** ein. Liste wird dann vor **Rückkehr in User-Mode** abgearbeitet.
(Linux)



Idee:

- Interrupt-Handler liest Daten aus Gerät in Puffer
- weckt **Thread** auf
- **Thread** liest Puffer aus und ...

Threads können **Prioritäten** besitzen.

Threads können **warten**.



Idee:

- Interrupt-Handler fügen Aufträge in **Work-Queue** ein.
- **Work-Queue** wird von **Worker-Thread-Pool** abgearbeitet.



Agenda

Einordnung

Beispiele

Aufteilung Interrupt-Bearbeitung

Implementierung

SoftIRQs

SoftIRQ-Beispiel

Verwandte Konzepte

Zusammenfassung



- Viele verschiedene Möglichkeiten, Interrupts abzuarbeiten.
- Immer Ziel: schnelle Reaktion auf (wichtige) Interrupts.
- Fast immer: „Hard“- und „Soft“-Interrupt-Teil.
- SoftIRQs: „Interrupts“ in Software.
- In aktuellen BS existieren weitere, von SoftIRQs abgeleitete Konzepte.

