

# Real-Time-Anwendungen unter Linux: Lessons Learned

Christian Seiler

2019-07-06

## Kurze Einführung

- Man kann unter Linux Threads sog. Real-Time-Priorität geben
- Es gibt einen speziellen Kernel-Patch, bei denen der Kernel selbst den Real-Time-Threads aus dem weg geht (PREEMPT\_RT), nähert das deutlich an echtes Real-Time-Verhalten an
- Allerdings gibt es etliche Fallstricke, die beim noramlen Programmieren nicht eintreten

## sched\_yield oft nicht hilfreich

- In Standardprogrammen oftmals schleifen in der Form (um Latenz zu verringern):  

```
while (!condition) sched_yield();
```

## sched\_yield oft nicht hilfreich

- In Standardprogrammen oftmals schleifen in der Form (um Latenz zu verringern):  

```
while (!condition) sched_yield();
```
- Im Fall SCHED\_FIFO, wenn ein anderer RT-Thread mit **gleicher** Priorität wartet, dann wird der Thread gewechselt

## sched\_yield oft nicht hilfreich

- In Standardprogrammen oftmals Schleifen in der Form (um Latenz zu verringern):  

```
while (!condition) sched_yield();
```
- Im Fall SCHED\_FIFO, wenn ein anderer RT-Thread mit **gleicher** Priorität wartet, dann wird der Thread gewechselt
- In allen anderen Fällen: RT-Thread blockiert alles andere auf der CPU, auf der er läuft, da Kernel ihn sofort wieder aufweckt

## Achtung vor langlaufenden RT-Threads (1)

- Nicht-Zeitkritische per-CPU-Aufgaben des Kernels (Aufräumarbeiten)
- RT-Thread blockiert eine CPU komplett  
⇒ Aufgaben werden nicht ausgeführt

## Achtung vor langlaufenden RT-Threads (1)

- Nicht-Zeitkritische per-CPU-Aufgaben des Kernels (Aufräumarbeiten)
- RT-Thread blockiert eine CPU komplett  
⇒ Aufgaben werden nicht ausgeführt
- Irgendwann warten dann andere CPU-Kerne auf die Erledigung dieser Aufgaben

## Achtung vor langlaufenden RT-Threads (1)

- Nicht-Zeitkritische per-CPU-Aufgaben des Kernels (Aufräumarbeiten)
- RT-Thread blockiert eine CPU komplett  
⇒ Aufgaben werden nicht ausgeführt
- Irgendwann warten dann andere CPU-Kerne auf die Erledigung dieser Aufgaben
- System wird immer träger und hängt sich auf



## Achtung vor langlaufenden RT-Threads (2)

- `/proc/sys/kernel/sched_rt_period_us`,  
`/proc/sys/kernel/sched_rt_runtime_us` können Abhilfe schaffen (RT-Threads dürfen maximal z.B. 95% der kompletten CPU-Zeit nutzen)

## Achtung vor langlaufenden RT-Threads (2)

- `/proc/sys/kernel/sched_rt_period_us`,  
`/proc/sys/kernel/sched_rt_runtime_us` können Abhilfe schaffen (RT-Threads dürfen maximal z.B. 95% der kompletten CPU-Zeit nutzen)
- Default-Setting ist plausibel, aber global, nicht pro CPU-Kern

## Achtung vor langlaufenden RT-Threads (2)

- `/proc/sys/kernel/sched_rt_period_us`,  
`/proc/sys/kernel/sched_rt_runtime_us` können Abhilfe schaffen (RT-Threads dürfen maximal z.B. 95% der kompletten CPU-Zeit nutzen)
- Default-Setting ist plausibel, aber global, nicht pro CPU-Kern
- Abhilfe:  

```
echo NO_RT_RUNTIME_SHARE | sudo tee  
/sys/kernel/debug/sched_features
```

## Achtung vor langlaufenden RT-Threads (2)

- `/proc/sys/kernel/sched_rt_period_us`,  
`/proc/sys/kernel/sched_rt_runtime_us` können Abhilfe schaffen (RT-Threads dürfen maximal z.B. 95% der kompletten CPU-Zeit nutzen)
- Default-Setting ist plausibel, aber global, nicht pro CPU-Kern
- Abhilfe:  

```
echo NO_RT_RUNTIME_SHARE | sudo tee  
/sys/kernel/debug/sched_features
```
- *Warnung*: man findet Online auch eine weitere Einstellung `RT_RUNTIME_GREED`; RedHat-only, wurde im Upstream-Kernel nicht angenommen

Danke

# Danke für die Aufmerksamkeit

Lizenz der Folien:



CC BY-SA 3.0 DE

Erstellt mit:

L<sup>A</sup>T<sub>E</sub>X Beamer, Theme: Antibes, Color Theme: beaver