

Linux wird echtzeitfähig: RT-Extension vs. PREEMPT_RT

Carsten Emde

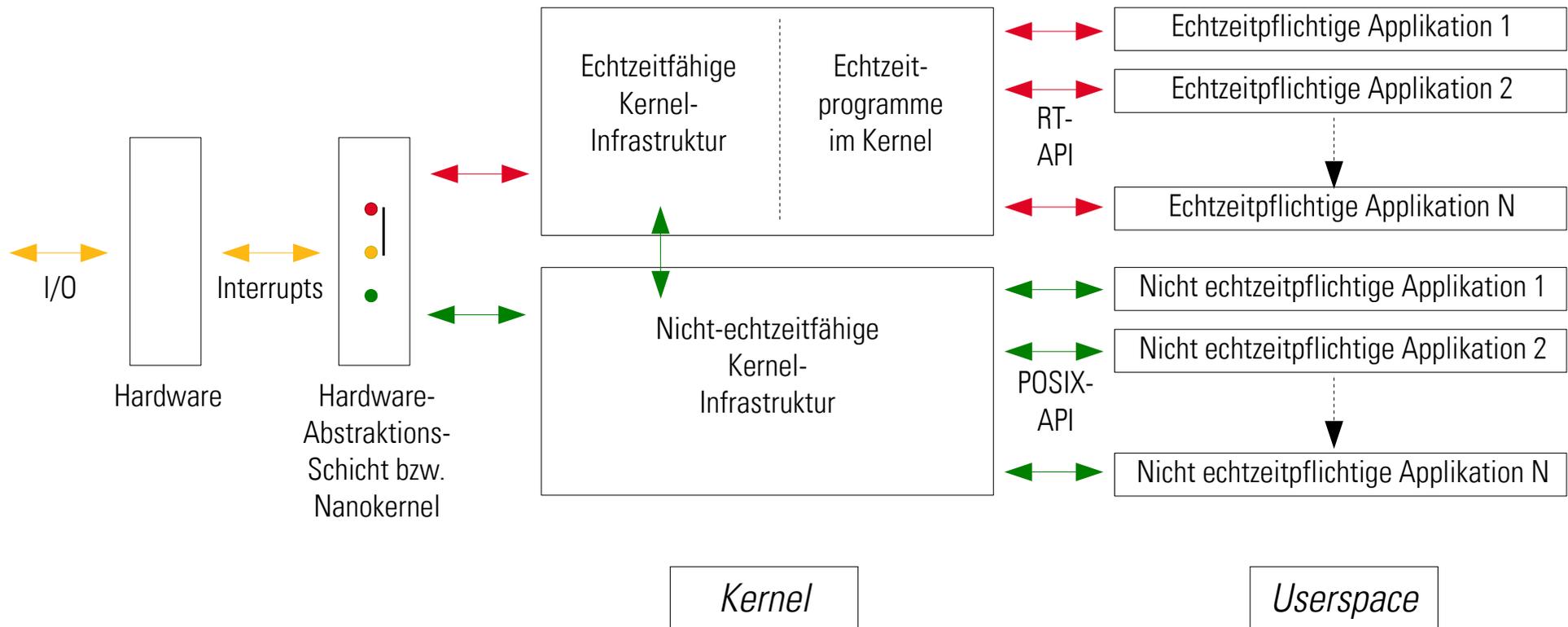
Open Source Automation Development Lab

(OSADL) eG

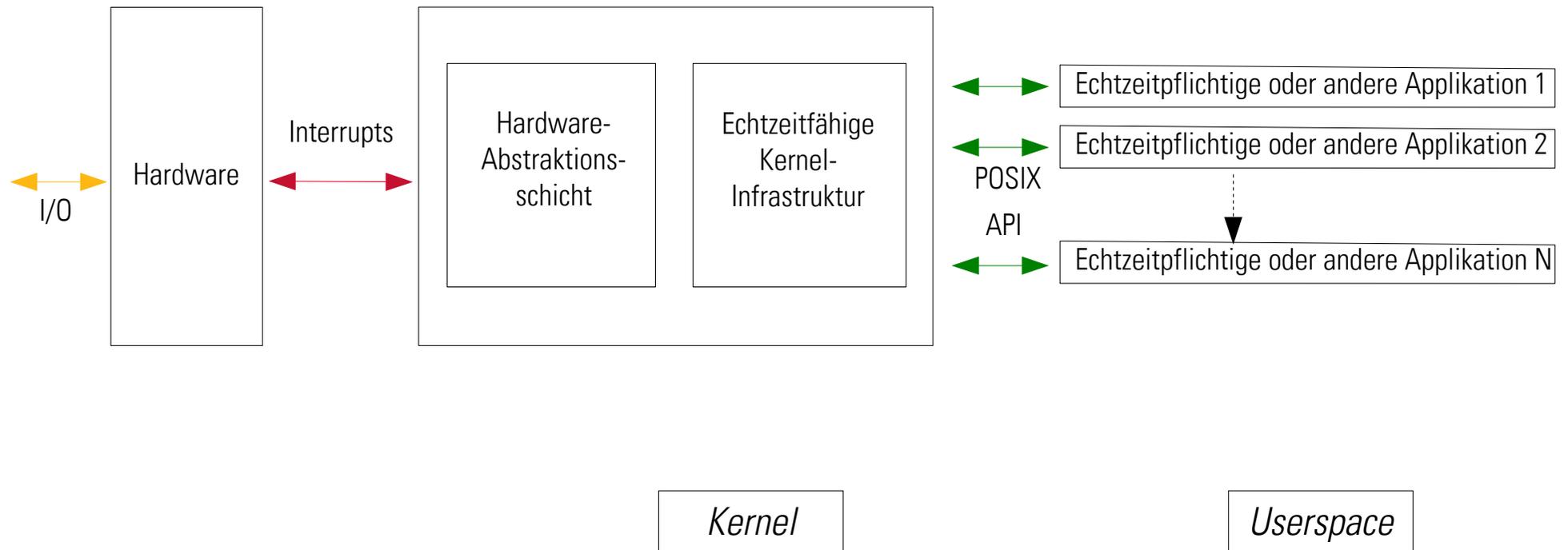
Zitat aus dem Jahre 2004, unbekannte Quelle

***“It's impossible to turn a General Purpose
Operating System (GPOS) Kernel into a Real
Time Operating System (RTOS) Kernel”***

Was ist eine Real Time Extension?



Was ist ein Real Time Operating System?



Terminologie

- RTLinux (*Dual-kernel approach*) verwendet einen getrennten Kernel (sogenannten Nanokernel):

RTLinux, RTAI, Xenomai, RTCore etc.

- CONFIG_PREEMPT_RT (*Single-kernel approach*):

Linux mainline realtime, "Linux RTOS"

Ziele

- Vollständig präemptibler Kernel
- Echtzeitgarantie (Determinismus) für die überwiegende Mehrzahl von Einsatzgebieten
- POSIX-Kompatibilität (Standard-API):
sched_setscheduler()
sched_setaffinity()
und verwandte Aufrufe

Treibende Kräfte für Echtzeit-Linux

- SMP
- (Auffinden von Race-Conditions)
- Audio
- Video
- Banking
- Automation

Autoren von Echtzeit-Linux

- Doug Niehaus, University of Kansas
- Thomas Gleixner, Linutronix
- Ingo Molnár, Red Hat
- Peter Zijlstra, Red Hat
- Paul E. McKenney, IBM
- Steven Rostedt, Red Hat
- viele andere

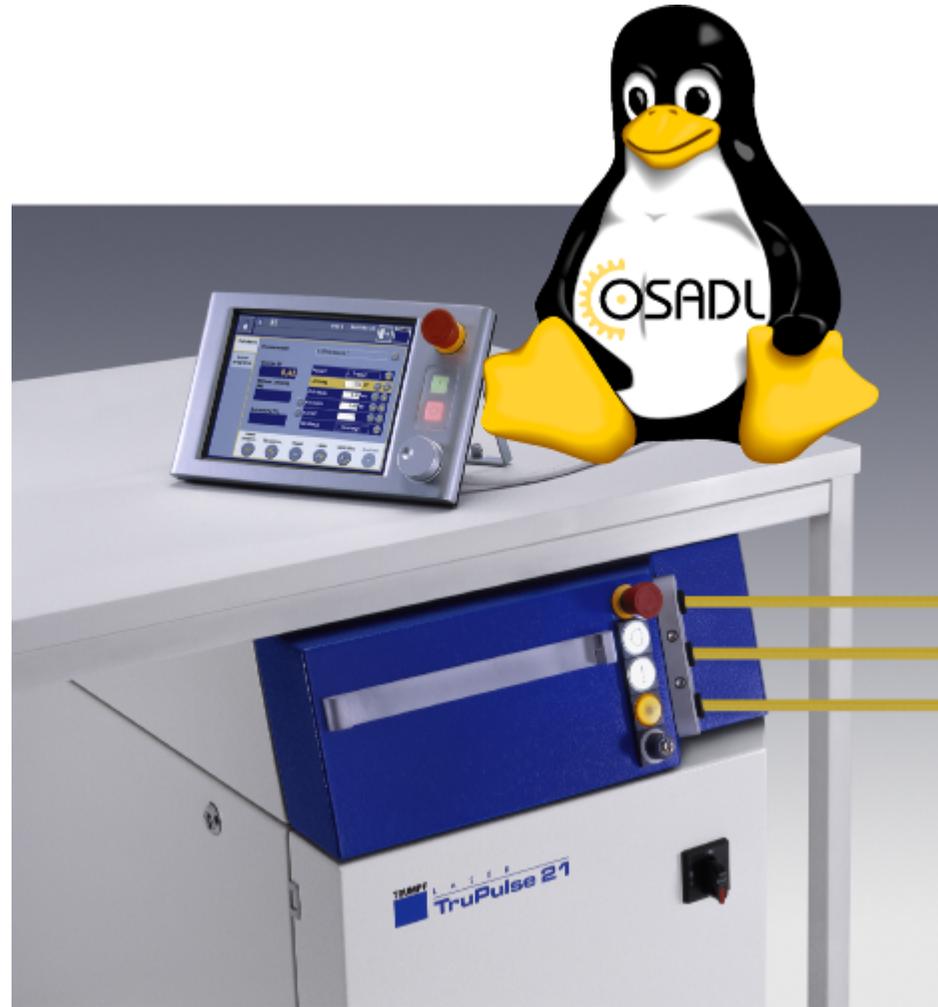
Komponenten von Echtzeit-Linux

- Deterministic Scheduler
- Preemption Support
- PI Mutexes
- High-Resolution Timer
- Preemptive Read-Copy Update
- IRQ Threads (selected, forced)
- Raw Spinlock Annotation
- Preemptive Memory Management
- Full Realtime Preemption Support

Geschichtlicher Ablauf

- Herbst 2004: MontaVista, Timesys und Lynuxworks veröffentlichen erste Teile von Echtzeit-Patches für den Mainline-Linuxkernel.
- Anfang 2005: Ingo Molnár reimplementiert die verschiedenen Komponenten und veröffentlicht den ersten kompletten Realtime-Preemption-Patch (PREEMPT_RT).
- Mitte 2005: Ein Team von Echtzeit-Spezialisten formiert sich.
- Kernel Summit 2006 in Ottawa: Es wird ein Plan zur sukzessiven Aufnahme der Patches in den Mainline-Linuxkernel beschlossen.

Kernel-Summit, Ottawa, August 2006



Veranstaltung 24576 „Echtzeitsysteme“ – Part 1
Linux wird echtzeitfähig: RT-Extension vs. PREEMPT_RT
Karlsruher Institut für Technologie, Sommersemester, 11. Juli 2023

Kernel-Summit, Ottawa, August 2006

"Controlling a laser with Linux is crazy, but everyone in this room is crazy in his own way. So if you want to use Linux to control an industrial welding laser, I have no problem with your using PREEMPT_RT."

Linus Torvalds

Realtime Linux Road Map

This is the current status of Realtime Linux using the Realtime-Preempt patches:

Architecture	x86	x86/64	powerpc	arm	mips	68knommu
Deterministic Scheduler	●	●	●	●	●	●
Preemption Support	●	●	●	●	●	●
PI Mutexes	●	●	●	●	●	● ³
High-Resolution Timer	●	● ¹	● ¹	● ¹	● ¹	●
Preemptive Read-Copy Update	● ²					
IRQ Threads	● ⁴	● ^{3,4,5}				
Raw Spinlock Annotation	● ⁶					
Full Realtime Preemption Support	●	●	●	●	●	● ³

● Available in mainline Linux

● Available when Realtime-Preempt patches applied

Kernel-Konfiguration

.config - Linux/x86 3.12.14 Kernel Configuration
→ Processor type and features

Preemption Model

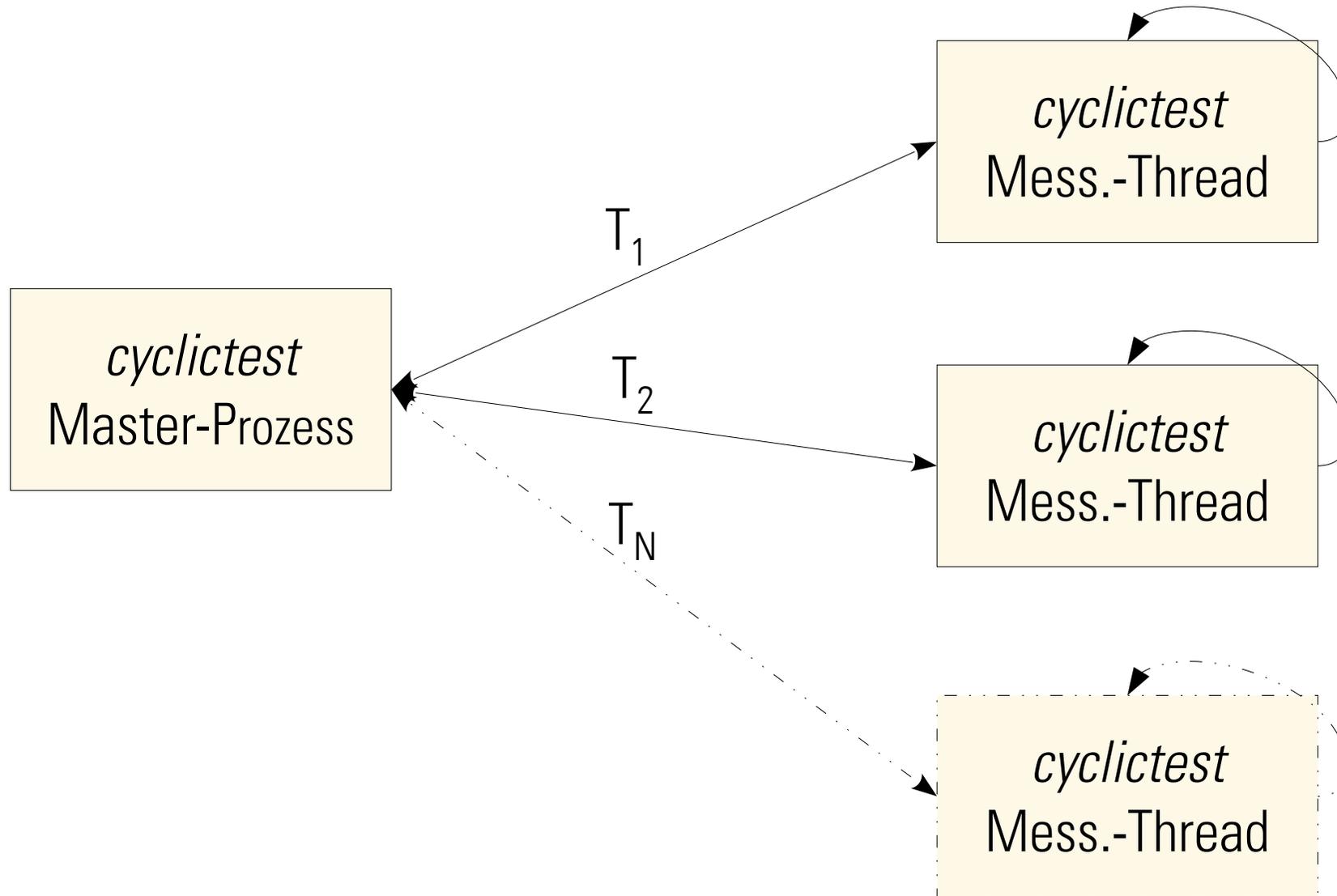
Use the arrow keys to navigate this window or press the hotkey of the item you wish to select followed by the <SPACE BAR>. Press <?> for additional information about this

- No Forced Preemption (Server)
- Voluntary Kernel Preemption (Desktop)
- Preemptible Kernel (Low-Latency Desktop)
- Preemptible Kernel (Basic RT)
- Fully Preemptible Kernel (RT)

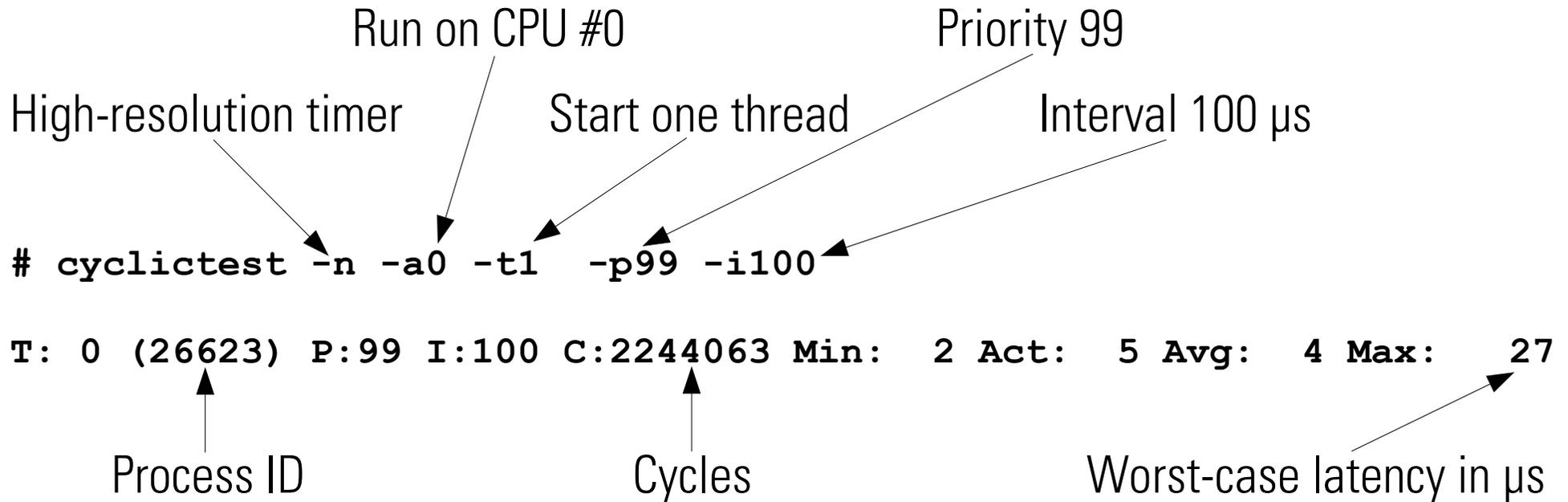
<Select>

< Help >

Latenz-Messung (1)



Latenz-Messung (2)



Latenz-Messung (3)

High-resolution timer Run on CPU #0 Priority 99, 98, 97, ... 88
Start 12 threads Interval 100 μ s No delay

```
# cyclicttest -n -a0 -t12 -p99 -i100 -d0
```

T: 0	(2910)	P:99	I:100	C:3217008	Min: 2	Act: 6	Avg: 4	Max: 32
T: 1	(2911)	P:98	I:100	C:3217008	Min: 1	Act: 4	Avg: 3	Max: 59
T: 2	(2912)	P:97	I:100	C:3217007	Min: 2	Act: 4	Avg: 3	Max: 47
T: 3	(2913)	P:96	I:100	C:3217007	Min: 2	Act: 11	Avg: 3	Max: 53
T: 4	(2914)	P:95	I:100	C:3217007	Min: 2	Act: 9	Avg: 4	Max: 53
T: 5	(2915)	P:94	I:100	C:3217007	Min: 3	Act: 9	Avg: 7	Max: 89
T: 6	(2916)	P:93	I:100	C:3217007	Min: 2	Act: 5	Avg: 4	Max: 85
T: 7	(2917)	P:92	I:100	C:3217006	Min: 2	Act: 10	Avg: 5	Max: 119
T: 8	(2918)	P:91	I:100	C:3217006	Min: 2	Act: 13	Avg: 9	Max: 148
T: 9	(2919)	P:90	I:100	C:3217007	Min: 1	Act: 4	Avg: 4	Max: 178
T:10	(2920)	P:89	I:100	C:3217006	Min: 1	Act: 4	Avg: 3	Max: 1413
T:11	(2921)	P:88	I:100	C:3217006	Min: 3	Act: 7	Avg: 10	Max: 27331

Latenz-Messung (4)

High-resolution timer Run on all CPUs Start 12 threads Priority 99, 98, 97, ... 88 Interval 100 μ s No delay

```
# cyclicttest -n -a -t12 -p99 -i100 -d0
```

T: 0	(11009)	P:99	I:100	C:3748311	Min: 2	Act: 3	Avg: 4	Max: 28
T: 1	(11010)	P:98	I:100	C:3748311	Min: 2	Act: 2	Avg: 4	Max: 27
T: 2	(11011)	P:97	I:100	C:3748311	Min: 2	Act: 10	Avg: 4	Max: 31
T: 3	(11012)	P:96	I:100	C:3748310	Min: 2	Act: 3	Avg: 3	Max: 14
T: 4	(11013)	P:95	I:100	C:3748310	Min: 2	Act: 3	Avg: 4	Max: 40
T: 5	(11014)	P:94	I:100	C:3748310	Min: 2	Act: 3	Avg: 5	Max: 29
T: 6	(11015)	P:93	I:100	C:3748310	Min: 2	Act: 3	Avg: 4	Max: 28
T: 7	(11016)	P:92	I:100	C:3748310	Min: 2	Act: 3	Avg: 4	Max: 20
T: 8	(11017)	P:91	I:100	C:3748309	Min: 2	Act: 7	Avg: 4	Max: 32
T: 9	(11018)	P:90	I:100	C:3748308	Min: 2	Act: 6	Avg: 4	Max: 27
T:10	(11019)	P:89	I:100	C:3748309	Min: 2	Act: 3	Avg: 4	Max: 24
T:11	(11020)	P:88	I:100	C:3748309	Min: 2	Act: 2	Avg: 5	Max: 46

Latenz-Messung (5)

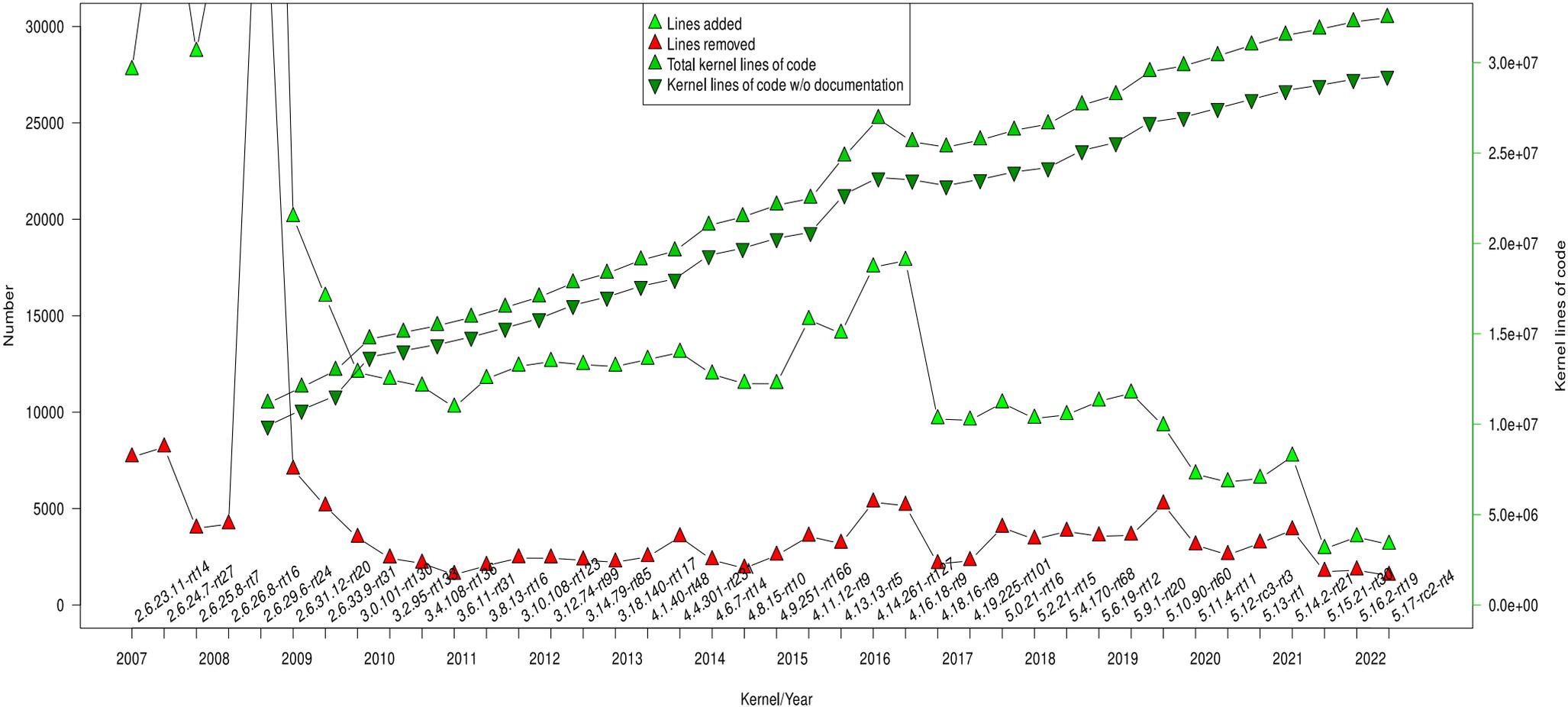
High-resolution timer Run on all CPUs Start 12 threads Priority 99 of all threads Interval 100 μ s No delay

`cyclictest` `-s` `-p99` `-i100` `-d0`

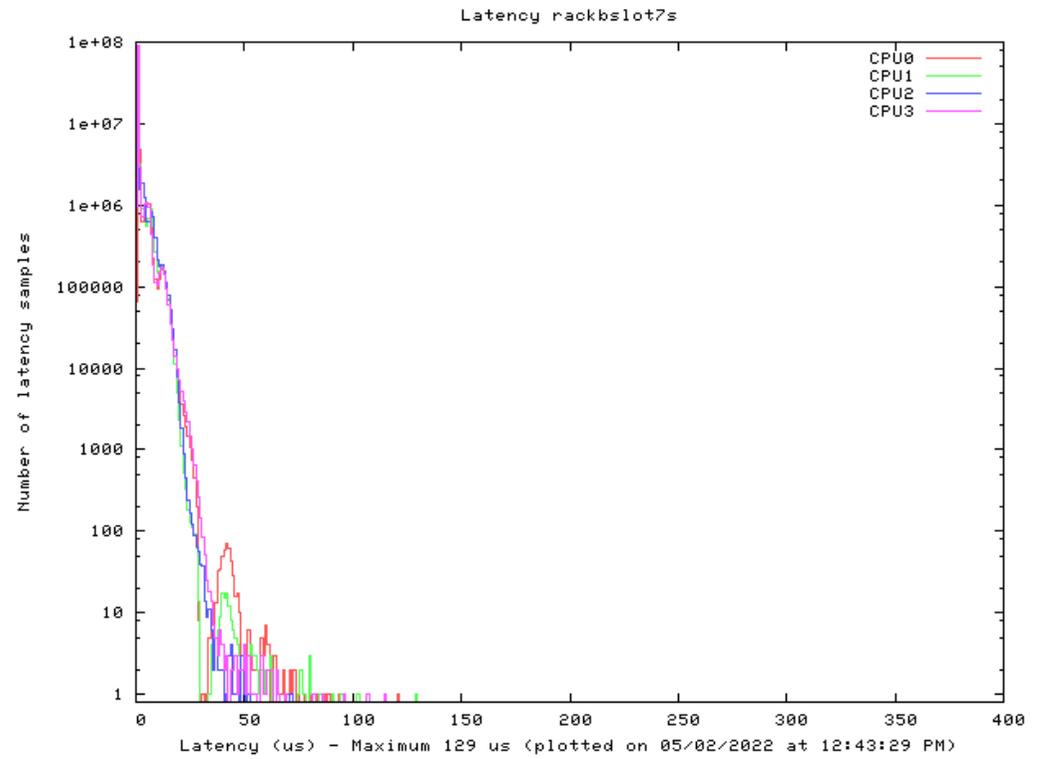
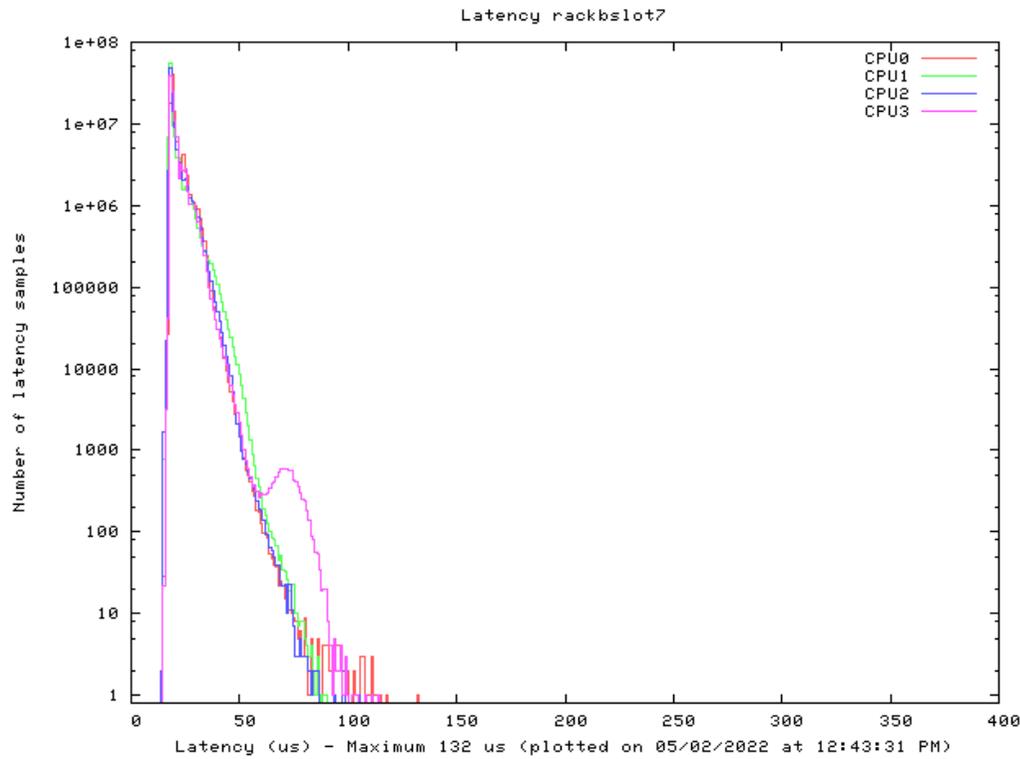
T: 0	(15350)	P:99	I:100	C:3839755	Min: 2	Act: 6	Avg: 3	Max: 24
T: 1	(15351)	P:99	I:100	C:3839755	Min: 2	Act: 7	Avg: 4	Max: 19
T: 2	(15352)	P:99	I:100	C:3839755	Min: 2	Act: 8	Avg: 4	Max: 27
T: 3	(15353)	P:99	I:100	C:3839755	Min: 2	Act: 5	Avg: 4	Max: 24
T: 4	(15354)	P:99	I:100	C:3839755	Min: 2	Act: 5	Avg: 3	Max: 20
T: 5	(15355)	P:99	I:100	C:3839755	Min: 2	Act: 5	Avg: 5	Max: 52
T: 6	(15356)	P:99	I:100	C:3839755	Min: 2	Act: 5	Avg: 4	Max: 20
T: 7	(15357)	P:99	I:100	C:3839755	Min: 2	Act: 5	Avg: 3	Max: 17
T: 8	(15358)	P:99	I:100	C:3839755	Min: 2	Act: 10	Avg: 4	Max: 28
T: 9	(15359)	P:99	I:100	C:3839754	Min: 2	Act: 5	Avg: 4	Max: 22
T:10	(15360)	P:99	I:100	C:3839755	Min: 2	Act: 5	Avg: 4	Max: 42
T:11	(15361)	P:99	I:100	C:3839755	Min: 2	Act: 5	Avg: 5	Max: 34

Status quo auf dem Weg zu Linux-Mainline

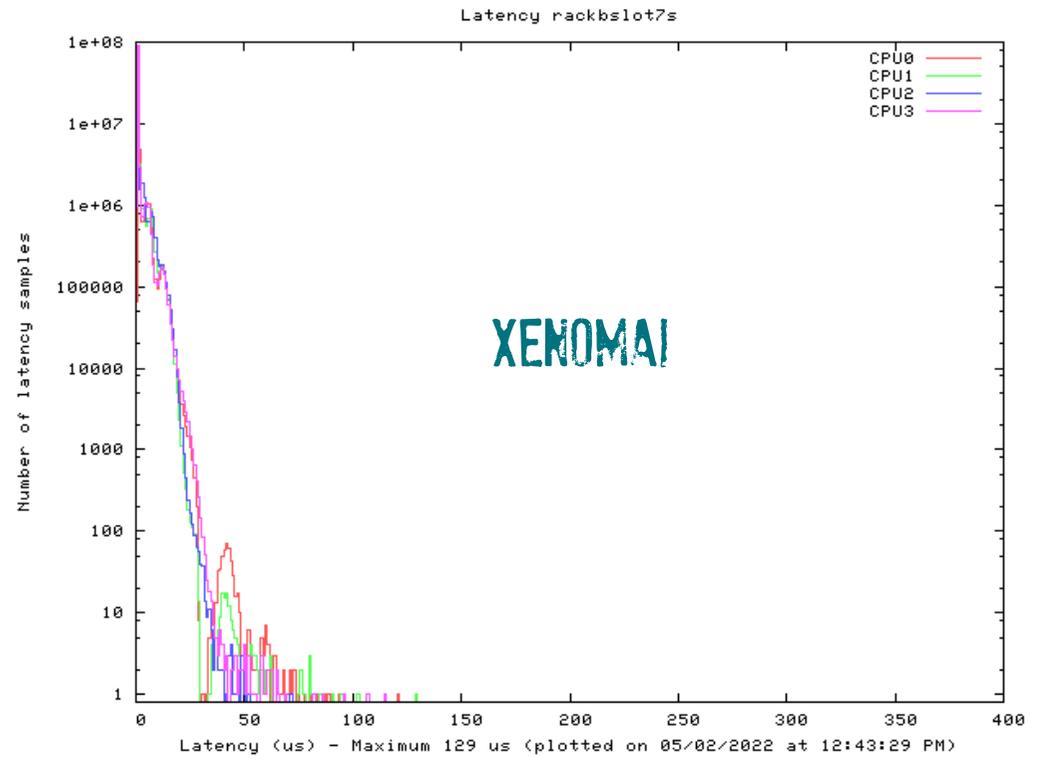
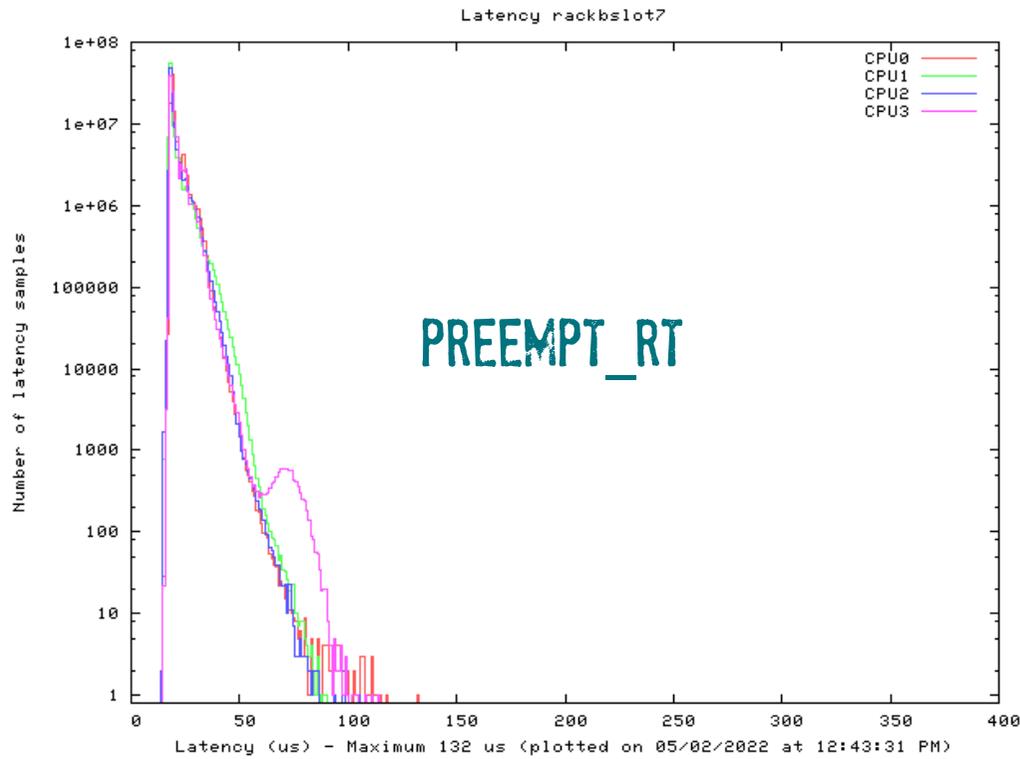
PREEMPT_RT Patch: Lines added/Lines removed



Vergleich PREEMPT_RT mit Xenomai



Vergleich PREEMPT_RT mit Xenomai



Zusammenfassung (1)

Es ist tatsächlich möglich, ein ursprünglich als General Purpose Operating System (GPOS) geplantes und entwickeltes Betriebssystem nachträglich in ein Real Time Operating System (RTOS) umzuwandeln.

Zusammenfassung (2)

Dabei ist es sogar gelungen, alle Eigenschaften des GPOS zu erhalten und entsprechende Echtzeitfähigkeit herzustellen.

Die Parametrisierung aus dem Userspace erfolgt ausschließlich über die Standard-POSIX-Schnittstelle.