

librtipc - Bibliothek für echtzeitfähige Interprozesskommunikation

Josef Raschen

Diplomarbeit: *Hardwareunabhängige Interprozesskommunikation
für Echtzeitanwendungen*

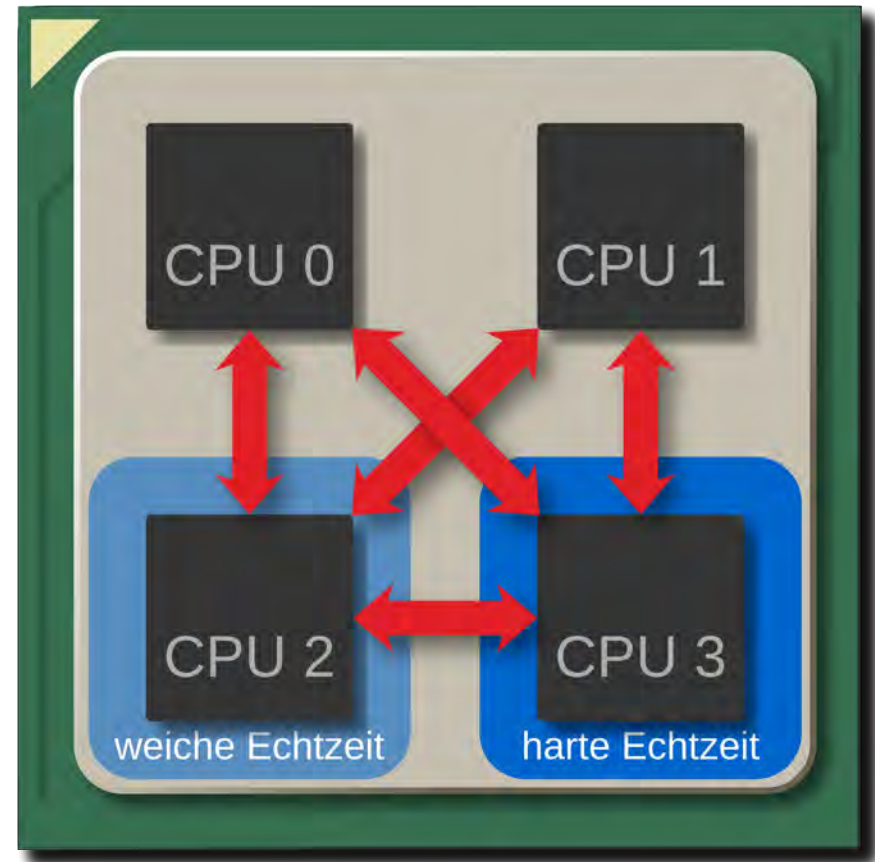
Lehrstuhl für Betriebssysteme
RWTH Aachen University

Workshop Echtzeit 2012

22.11.2012

Idee

- harte Echtzeit auf gewöhnlichem Betriebssystem
- Realisierungsansatz: CPU-Isolation auf Mehrprozessorsystemen
- Kommunikation zwischen echtzeitfähigen und nicht echtzeitfähigen Teilen des Systems erforderlich



Interprozesskommunikation

- betriebssystemeigene Methoden nicht echtzeitfähig
- Ansatz: Implementierung unter Nutzung von Shared-Memory
- Koordinierung des Zugriffs auf den Shared-Memory nötig
- Problem: Beeinflussung der Kommunikationsteilnehmer untereinander durch Synchronisation

Prozesssynchronisation

- blockierende Synchronisationstechniken
 - Implementierung einfach
 - Verhalten aller Teilnehmer muss abschätzbar sein
- nicht-blockierende Synchronisationstechniken
 - lock-free
 - benötigt keine Locks für Zugriff auf geteilte Ressourcen
 - Implementierung häufig mit Retry-Loops
 - systemweit wird ein Prozess erfolgreich weiterarbeiten
 - wait-free
 - spezielle Variante von lock-free
 - Algorithmen garantieren Erfolg innerhalb vorgegebener Frist

IPC-Objekte

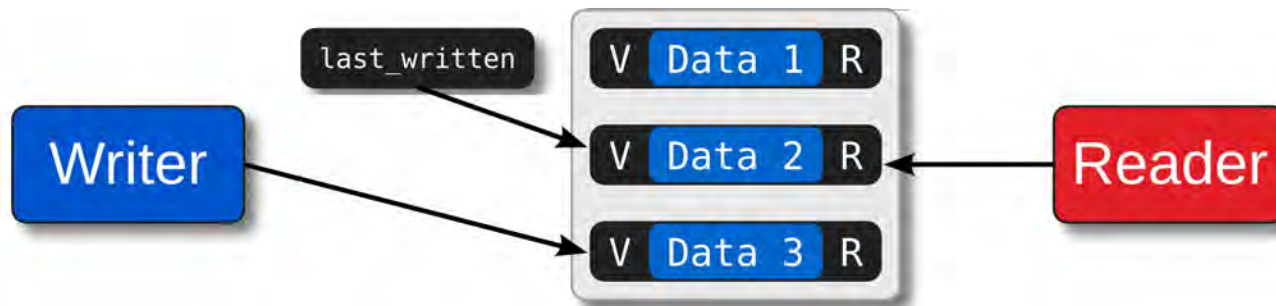
IPC-Objekt	Intel x86	ARMv7
Mutex	blockierend	
Barrier		
Flag		
Wait-Free SPSC Queue	wait-free	wait-free
Wait-Free MPSC Queue	wait-free	lock-free producer
		wait-free consumer
Lock-Free MPSC Queue	lock-free producer	lock-free producer
	wait-free consumer	wait-free consumer
Lock-Free MPMC Queue	lock-free	lock-free
Sensor-Buffer	wait-free writer	wait-free writer
	lock-free reader	lock-free reader

Sensor-Buffer

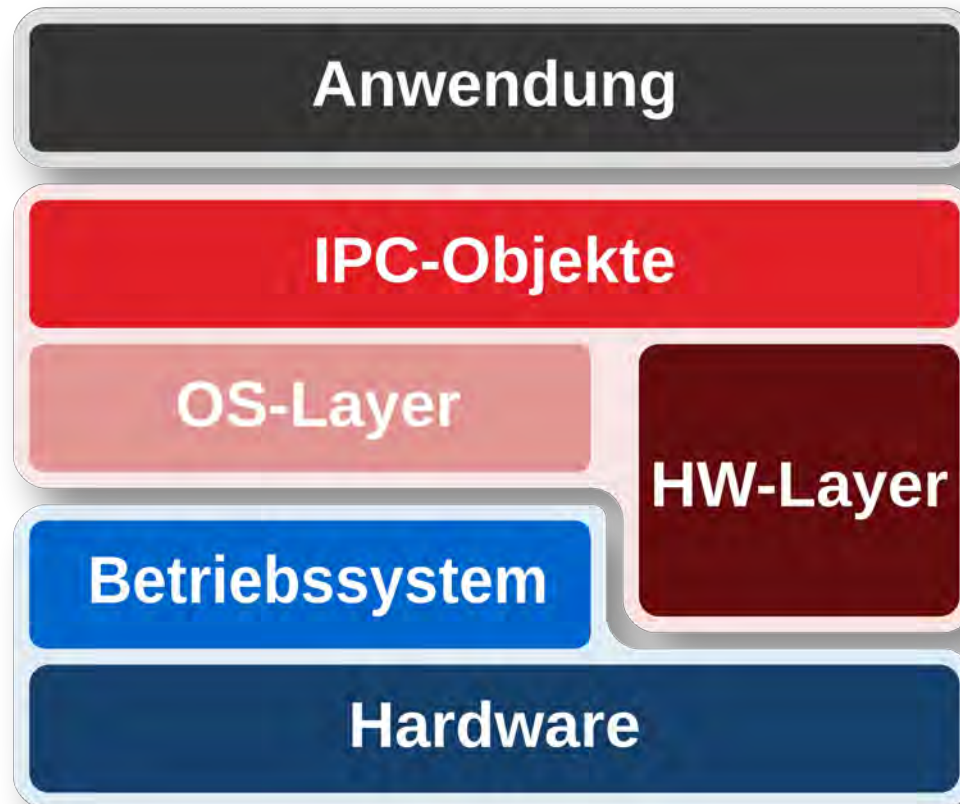
- Szenario
 - unter harten Echtzeitanforderungen laufender Prozess soll Daten an nicht echtzeitfähigen Prozess übermitteln
 - jeweils nur die aktuellsten Daten sind von Interesse (z.B. Sensordaten)
- Problem
 - echtzeitfähiger Prozess muss immer innerhalb eines vorgegebenem Zeitfensters schreiben können
 - Gültigkeit der gelesenen Daten muss sichergestellt werden
 - Queue-Implementierung hier ungeeignet

Implementierung Sensor-Buffer

- Ringpuffer zur Speicherung der Daten
- Valid-Flag:
 - Zurücksetzen vor Schreibzugriff
 - Setzen, wenn Daten gültig
- Reader-Flag
 - Zurücksetzen vor Schreibzugriff
 - Setzen bei Beginn des Lesezugriffs und Überprüfen nach Beendigung des Lesens
- gemeinsame Variable *last_written* zur Signalisierung des zuletzt geschriebenen Eintrags
- schreibender Zugriff: immer erlaubt, wait-free
- lesender Zugriff: lock-free



librtipc - Struktur



Zusammenfassung

- echtzeitfähige, hardwareunabhängige IPC-Lösung
- basiert auf Shared-Memory
- verschiedene IPC-Objekte mit den Charakteristiken blockierend, nicht-blockierend und wait-free
- Implementierung für Linux auf x86 und ARMv7
- leicht portierbar auf weitere Betriebssysteme und Hardwarearchitekturen

Literatur

Wassen, G., Lankes, S. und Bemmerl, T.: *Harte Echtzeit für Anwendungsprozesse in Standard-Betriebssystemen auf Mehrkernprozessoren*, Herausforderungen durch Echtzeitbetrieb - Echtzeit 2011, 2012

Raschen, J.: *Hardwareunabhängige Interprozesskommunikation für Echtzeitanwendungen*, Diplomarbeit, Lehrstuhl für Betriebssysteme, RWTH Aachen University, 2011

Kontakt

Josef Raschen

josef@raschen.org