

Vertiefung Systemtechnik

Prof. Dr. Peter Gerwinski

16. Oktober 2012

2 Theorie der Echtzeitsysteme

2.1 Multitasking

- *Kooperatives Multitasking*
Prozesse geben freiwillig Rechenzeit ab
- *Präemptives Multitasking*
Das Betriebssystem unterbricht laufende Prozesse
(englisch: *to pre-empt* – jemandem zuvorkommen)
- *Scheduler*
Steuerprogramm, das Prozessen Rechenzeit zuteilt
- *Kontextwechsel*
Umschalten zwischen zwei Prozessen → teuer

2 Theorie der Echtzeitsysteme

2.2 Ressourcen

Qualitätssicherung beim Multitasking

- Verschiedene Anforderungen:
Latenz vs. *Jitter* vs. *Durchsatz*
interaktiv vs. *Echtzeit* vs. *Stapel*
- Ressourcen reservieren:
Mutexe (= spezielle *Semaphore*)
und *Spinlocks*
- Verschiedene Methoden
der Priorisierung
- Umgehung der Probleme durch
speziell geschriebene Software
→ s. o. (MultiWii, RP6, ...)

Qualitätssicherung für Netzwerke:

- Verschiedene Anforderungen:
Latenz vs. *Jitter* vs. *Verluste*
vs. *Durchsatz*
- Ressourcen reservieren:
IntServ mit *Resource Reservation Protocol (RSVP)*
- Klassifizierung und Priorisierung:
DiffServ mit Type-of-Service-Bits
(IPv4) bzw. Traffic-Class-Bits (IPv6)
im IP-Header
- Eigenes Protokoll (Telefondienste):
Asynchronous Transfer Mode (ATM)

2 Theorie der Echtzeitsysteme

2.2 Ressourcen

Ressourcen reservieren

- *Semaphor*
griechisch: *sema* – Zeichen, *pherein* – tragen
„Eisenbahnsignal“
- *Mutex*
englisch: *mutual exclusion* – wechselseitiger Ausschluß
- *Spinlock*
englisch: *spin* – rotieren, *lock* Sperre
busy waiting auf etwas Schnelles, z. B. auf einen Semaphor

Hardware-Unterstützung: Prüfen, ob eine Variable einen bestimmten Wert hat; wenn ja, auf anderen Wert setzen; andere Prozessoren solange anhalten

2 Theorie der Echtzeitsysteme

2.2 Ressourcen

Ressourcen reservieren

- *Semaphor*
griechisch: *sema* – Zeichen, *pherein* – tragen
„Eisenbahnsignal“
[linux-3.7-rc1/kernel/semaphor.c](#)
- *Mutex*
englisch: *mutual exclusion* – wechselseitiger Ausschluß
[linux-3.7-rc1/kernel/mutex.c](#)
- *Spinlock*
englisch: *spin* – rotieren, *lock* Sperre
busy waiting auf etwas Schnelles, z. B. auf einen Semaphor
[linux-3.7-rc1/kernel/spinlock.c](#)

Hardware-Unterstützung: Prüfen, ob eine Variable einen bestimmten Wert hat; wenn ja, auf anderen Wert setzen; andere Prozessoren solange anhalten

2 Theorie der Echtzeitsysteme

2.2 Ressourcen

Ressourcen reservieren

- *Semaphor*

griechisch: *sema* – Zeichen, *pherein* – tragen
„Eisenbahnsignal“

[linux-3.7-rc1/kernel/semaphor.c](#)

[linux-3.7-rc1/drivers/usb/core/file.c](#)

- *Mutex*

englisch: *mutual exclusion* – wechselseitiger Ausschluß

[linux-3.7-rc1/kernel/mutex.c](#)

[linux-3.7-rc1/drivers/usb/serial/usb-serial.c](#)

- *Spinlock*

englisch: *spin* – rotieren, *lock* Sperre

busy waiting auf etwas Schnelles, z. B. auf einen Semaphor

[linux-3.7-rc1/kernel/spinlock.c](#)

[linux-3.7-rc1/kernel/semaphor.c](#), [linux-3.7-rc1/kernel/mutex.c](#)

Hardware-Unterstützung: Prüfen, ob eine Variable einen bestimmten Wert hat; wenn ja, auf anderen Wert setzen; andere Prozessoren solange anhalten

Zombies

Wikipedia:

Als Zombie wird die fiktive Figur eines zum Leben erweckten Toten (Untoter) oder eines seiner Seele beraubten, willenlosen Wesens bezeichnet. Der Begriff leitet sich von dem Wort nzùmbe aus der zentralafrikanischen Sprache Kimbundu ab und bezeichnet dort ursprünglich einen Totengeist.

Ein Zombie-Prozeß ist bereits beendet („tot“), bekommt keine Rechenzeit mehr („seiner Seele beraubt“), hat alle belegten Ressourcen wieder freigegeben („willenlos“), wird aber noch in der Prozeßliste geführt („untot“).

- Warum? —> Ein anderer Prozeß (Elternprozeß) wartet noch auf den Rückgabewert des beendeten Prozesses.
- Schadet das? —> Nein.
- Aber? —> Wenn sich Zombie-Prozesse anhäufen, deutet dies auf einen Prozeß hin, der andere Prozesse erzeugt und anschließend „hängt“.
- Beispiel: Datenträger entfernt, zugreifender Prozeß „hängt“.
—> Tochterprozesse werden zu Zombies.