

Recap: Queue, Semaphore, Mutex

Silvio Emmenegger Pascal Häfliger

11. November 2015

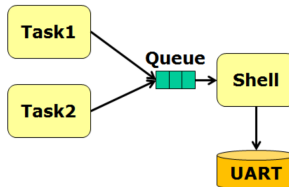
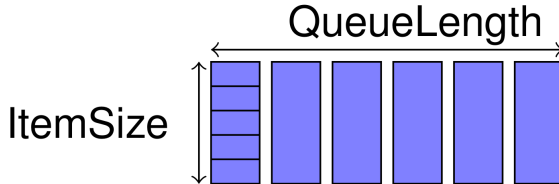
Inhaltsverzeichnis

- 1 Queue
- 2 Semaphore
- 3 Mutex

FRTOS:
Höhere Priorität = Wichtiger

- 1 Queue
- 2 Semaphore
- 3 Mutex

Queue



- 1 Queue
- 2 Semaphore
- 3 Mutex

Übersicht

- Primitive Lösung für Synchronisation
- Thread übergreifende Datenaustausch
- Nicht geschützt gegen Interrupts!!!

Prioritäten Inversion - Vererbung

Problem: Priorität Inversion

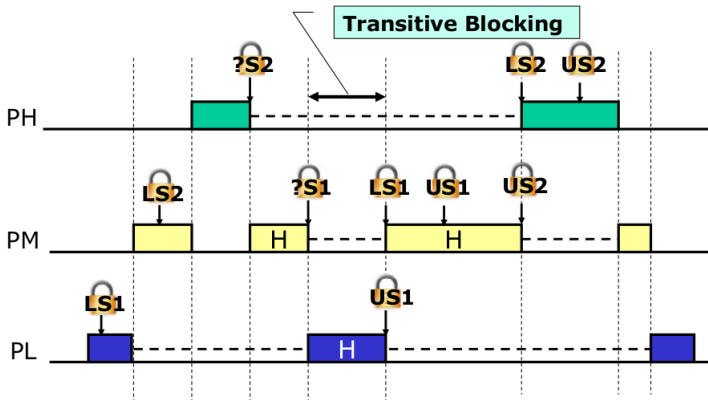
- Tiefere Priorität blockiert höhere Priorität

Lösung:

- Prioritäten Vererbung: "Lower-prio task inherits the prio of any higher-prio task pending on resource"

Deadlocks weiterhin möglich

Prio Inheritance



Prioritäten Inversion - "Ceiling"

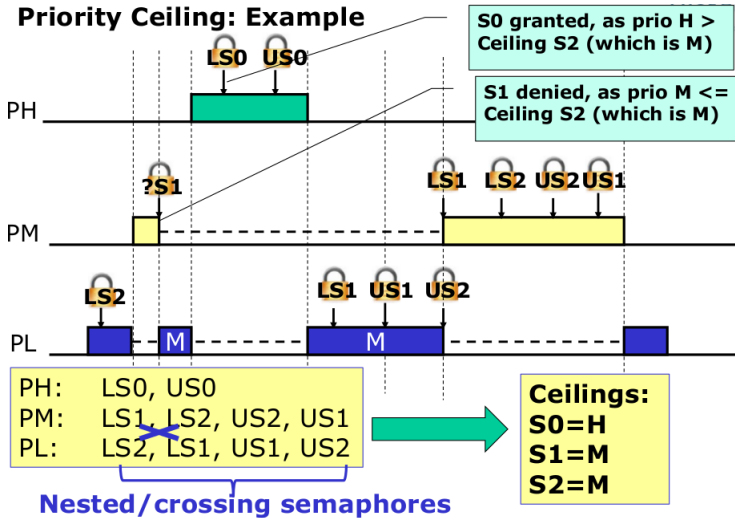
Problem: Priorität Inversion

- Tiefere Priorität blockiert höhere Priorität
- Deadlocks

Lösung:

- Prioritäten Vererbung: "Lower-prio task inherits the prio of any higher-prio task pending on resource"

Prio Ceiling



- 1 Queue
- 2 Semaphore
- 3 Mutex**

Mutex

- Binärer Semaphore
- Ressourcen schützen, 1 Thread in kritischer Section
- Muss immer freigegeben werden!!!
- Priority Inheritance \rightarrow Deadlock möglich
- Nicht geschützt gegen Interrupts!!!