Domänenanalyse Threadverwaltung/Scheduling

Johannes Handl, Marc Rößler, Christian Strengert

15. Mai 2003

Domänendefinition

Die Erzeugung, Verwaltung, Umschaltung/Wechsel, und Terminierung von Threads auf einer CPU im Bereich der Microcontroller. Teil der Domäne sind insbesondere das Erstellen, Sichern und Wiederherstellen des Fadenkontextes.

Nicht Teil der Domäne sind:

- Speicherschutz
- CPU-Schutz
- Thread-Synchronisation

Domänenlexikon (1/3)

Coroutine: Aktivitätsträger mit eigenem Registersatz; Auruf anderer Coroutinen führt nicht zu Rekursion, d.h. der Stack wächst nicht an sondern die aufgerufene Coroutine setzt an der vorigen Stelle wieder auf. Basierend auf Couroutinen können Threads implementiert werden.

Thread: Gedachter Programmfluss, der quasi-parallel zu anderen Proramm-flüssen läuft (durch fortwährendes Umschalten zwischen Threads). Die Auswahl des nächsten zu aktivierenden Threads erfolgt im Gegensatz zur Coroutine nicht durch den Thread selbst, sondern üblicherweise durch einen Scheduler.

Domänenlexikon (2/3)

Scheduler: Programmteil, der den nächsten zu aktivierenden Thread bestimmt. Der zu verwendende Algorithmus ist dabei nicht festgelegt.

Statische Threaderzeugung: Erzeugen einer bestimmten Anzahl von Threads beim Systemstart.

dynamische Threaderzeugung: Erzeugen von Threads zur Laufzeit möglich.

Kontext: Registersatz, variabler Umfang, mindestens aber PC und SP.

Kontextwechsel: Austausch des festgelegten Kontext eines zu deaktivierenden Threads gegen den eines zu aktivierenden Threads (mindestens SP und PC, u.U. sichert/restored der Thread auch selbst).

Domänenlexikon (3/3)

Flüchtige Register: Register, die u.U. überschrieben werden.

Nichtflüchtige Register: z.B. SP, PC

Kooperatives Scheduling: Der Thread gibt die Kontrolle freiwillig ab

präemptives Scheduling: Dem Thread wird die Kontrolle durch einen Interrupt entzogen

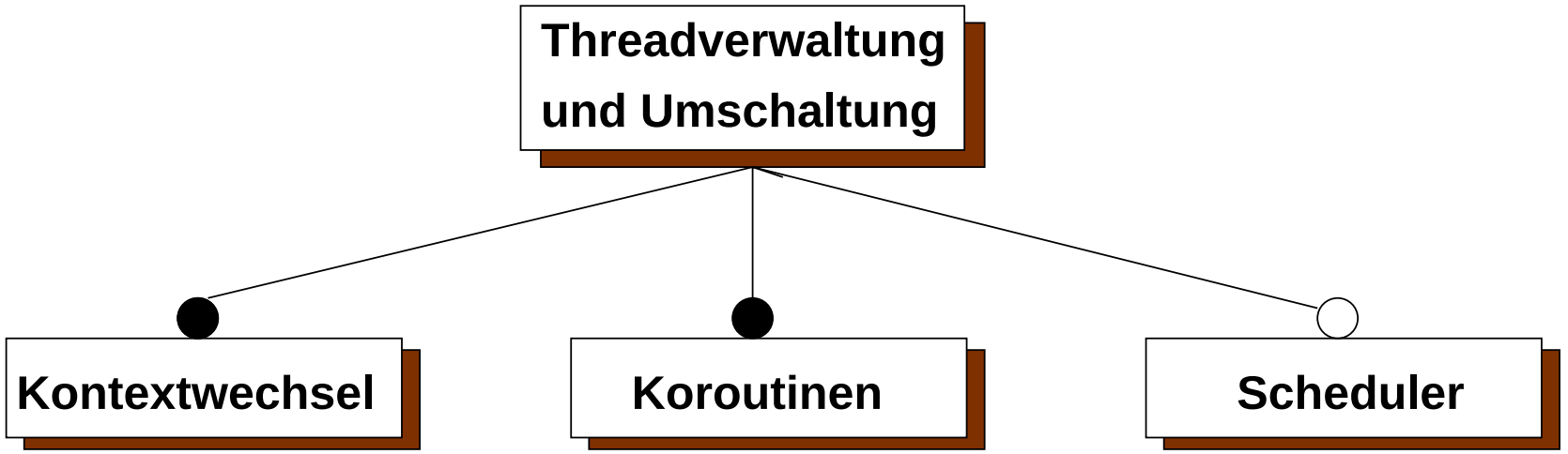
Dispatcher: Durchsetzen der Scheduling-Entscheidung

PCB: Process Control Block; Datenstruktur zur Verwaltung des Threads, enthält z.B. Prioritäten und u.U. auch den Kontext des Threads.

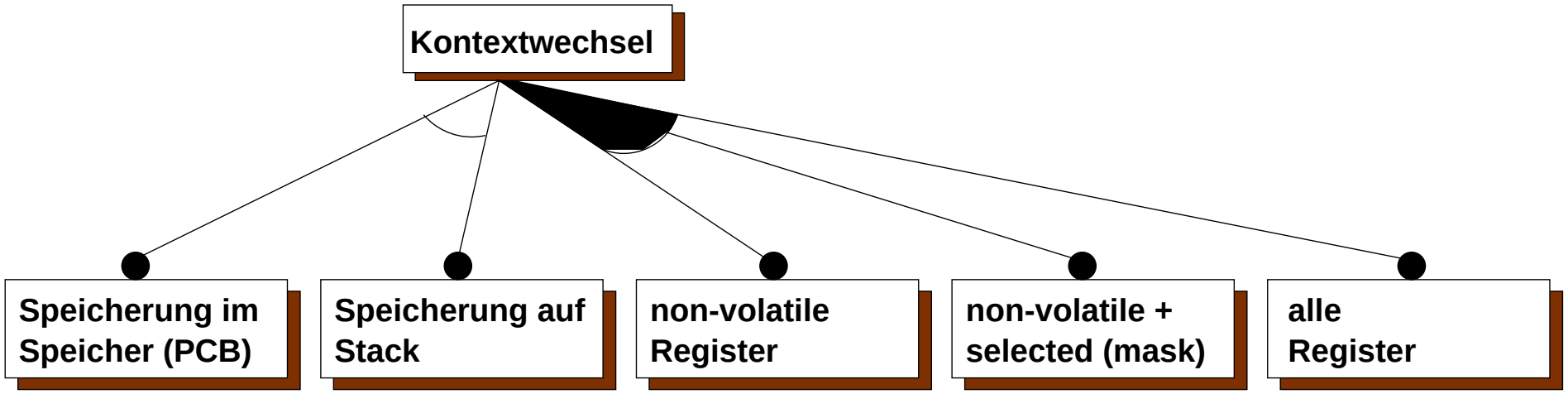
Konzeptmodell

Modelliert werden soll Programmcode, welcher Threads erzeugen, verwalten und terminieren kann. Die Threadverwaltung wird hauptsächlich durch den Scheduler durchgeführt, der nach bestimmten Kriterien (FCFS, SJF, RR, ...) den jeweils nächsten zu aktivierenden Thread auswählt und ihn mittels des Dispatchers zur Ausfürhung bringt. Der Threadwechsel kann dabei kooperativ - d.h. durch eigenständige Abgabe der CPU - oder präemptiv - durch Timer Interrups - erfolgen. Die PCBs enthalten dabei je nach Definition mindestens den minimalen Kontext, bestehend aus PC und SP, bis hin zum kompletten Registersatz der CPU. Die Threads bauen auf Couroutinen auf.

Featuremodell (1/3)



Featuremodell (2/3)



Featuremodell (3/3)