

Lutz Schröder
Christoph Lüth

Techniken zur Entwicklung Korrekter Software
Übungsblatt 2
Abgabe: 10.12.2007

Ein priorisierter Round-Robin Scheduler

In einem einfachen Round-Robin Scheduler bekommen Prozesse, die auf eine bestimmte Ressource warten, dieselbe der Reihe nach zugeteilt; am Ende wird beim ersten Prozess wieder angefangen. Eine Variante ist das priorisierte Scheduling, in dem jeder Prozess eine „angeborene“ Priorität besitzt, und gemäß dieser Priorität häufigeren oder weniger häufigen Zugriff auf die Ressource bekommt. Dies lässt sich auf verschiedene Weisen realisieren, etwa indem man die Ressource zu gegebenen Anteilen auf die Prioritätsklassen aufteilt und innerhalb jeder Prioritätsklasse per normalem Round Robin verteilt, oder indem man Prozesse nach jeder Zuteilung um eine Prioritätsklasse herabstuft und bei Erreichen der untersten Klasse wieder ihre auf ihre angeborene Priorität setzt; in letzterem Falle wird die Ressource stets an einen Prozess der höchsten Prioritätsklasse zugeteilt.

Führe eine Entwicklung zunächst eines einfachen Round-Robin-Schedulers und dann eines priorisierten Schedulers in CASL durch; der Einfachheit halber reicht es aus, jeweils eine Anforderungs- und eine Entwurfspezifikation anzugeben. Bei geschickter Strukturierung können Teile der einfacheren Spezifikation in der komplexeren wiederverwendet werden. Argumentiere auf Papier (aber durchaus im üblichen mathematischen Sinne formal), warum jeweils der Entwurf die Anforderungen erfüllt.

Besonders zu berücksichtigende Punkte:

- Da CASL intern zustandslos ist, müssen Zustände und das Verstreichen von (diskreter) Zeit explizit modelliert werden.
- Prozesse werden stark abstrahiert, sollten aber zwischen Zuständen wie *waiting*, *running*, *sleeping* wechseln können.
- Die Anforderungen sollten unter anderem beinhalten, dass wartende Prozesse irgendwann Zugriff auf die Ressource erhalten, und dass höher priorisierte Prozesse in geeigneter Weise bessergestellt sind.