

Grundlegende Übungen

Aufgabenblatt 2 - aperiodische Aufgaben

Peter Ulbrich, Martin Hoffmann

15. November 2010

Diese Aufgabe dreht sich um die Antwortzeit aperiodischer Aufgaben in zeitgesteuerten System und die sogenannten sporadischen Aufgaben. Welche im Gegensatz zu den aperiodischen Aufgaben harte Termine mit sich bringen und besonders behandelt werden müssen. Ausgangspunkt für diese Aufgabenstellung ist ein rein zeitgesteuerter Ablaufplaner, der aperiodische Aufgaben im Hintergrund nach der FCFS-Strategie ausführen kann. Der Ablaufplaner hat dabei die Eigenschaften, wie sie in den Tafelübungen zu Aufgabe 2 besprochen wurden. Gegeben sei weiterhin eine Menge von Aufgaben, die zu den Aufgaben aus Aufgabenblatt 1 identisch ist:

Aufgabe	p/a	Periode	min. Zwischenankunftszeit	WCET
T1	p	20 ms	NA	4 ms
T2	p	40 ms	NA	5 ms
T3	p	80 ms	NA	2 ms
T4	p	30 ms	NA	1 ms
T5	a	NA	120 ms	3 ms

Aufgabe 1 Die aperiodische Aufgabe T5 soll nun im Hintergrund ausgeführt werden, wofür der Scheduler nun die *Verdrängung* solcher Aufgaben implementiert. Aktiviert wird T5 in einer Unterbrechungsbehandlung. **Hinweis:** Simulieren Sie das aperiodische Ereignis, wie in Aufgabe 1, mit Hilfe eines `Interrupt_Timers` (siehe EZStubs Einführung, Folien 31ff).

- Implementieren Sie eine Ablauftabelle welche die periodischen Aufgaben T1-T4 umfasst, sowie eine Unterbrechungsbehandlung die die aperiodische Aufgabe T5 aktiviert. Zeigen Sie mit Hilfe einer Messung den Unterschied zu der Implementierung in Aufgabe 4 aus Aufgabenblatt 1.
- Nehmen Sie an, es handle sich bei Aufgabe T5 nicht um eine aperiodische Aufgabe sondern um eine sporadische Aufgabe mit einem relativen Termin von 80 ms. Eine solche Aufgabe darf nur dann zur Bearbeitung angenommen werden, wenn sichergestellt werden kann, dass sie termingerecht abgearbeitet werden kann. Man bezeichnet eine derartige Überprüfung als *Übernahmeprüfung*. Skizzieren Sie, wie eine solche Übernahmeprüfung im vorliegenden Fall aussehen könnte. **Hinweis:** Welche Größen müssen im Betriebssystem bekannt und welche Anforderungen erfüllt sein um diese Übernahmeprüfung zu berechnen.

- c) Wie sollte sich das System verhalten, wenn die Übernahmeprüfung negativ verläuft? Die Übernahmeprüfung verläuft negativ, wenn die Aufgabe also nicht zur Bearbeitung angenommen werden kann, weil nicht garantiert werden kann, dass der Termin eingehalten wird. **Hinweis:** Welcher Teil des Systems muss reagieren?

Aufgabe 2 Die Behandlung der aperiodischen Aufgabe lässt sich weiter verbessern, d.h. die Antwortzeit kann weiter verkürzt werden. Hierfür müssen Sie die aperiodischen Aufgabe an einer anderen Stelle implementieren / verschieben. **Hinweis:** Dieses Ziel lässt sich mit den Mitteln erreichen, die Ihnen der Ablaufplaner aus diesem Aufgabenblatt bietet.

- a) Implementieren sie eine alternative Ausführungen der aperiodischen Aufgabe T5 welche die Antwortzeit weiter verkürzt. Zeigen Sie mit Hilfe einer Messung die negativen Konsequenzen dieser Verbesserung auf die anderen Aufgaben.
- b) Welches in der Vorlesung vorgestellte Verfahren vermeidet diese negativen Konsequenzen? **Hinweis:** Dieses Verfahren steht im Ablaufplaner aus Aufgabe 2 **nicht** zur Verfügung. Erläutern Sie, wie man den Ablaufplaner erweitern müsste, um dieses Verfahren zu unterstützen.

Aufgabe 3 Nehmen Sie nun an, die periodische Aufgabe T1 und die nicht-periodische Aufgabe T5 verwenden ein gemeinsames Betriebsmittel.

- a) Skizzieren Sie ein Problemszenario, dass beim Zugriff auf das gemeinsame Betriebsmittel entstehen kann.
- b) Welche Möglichkeiten gibt es, dieses Problem lösen? **Hinweis:** Diese Lösungen können, müssen aber nicht unbedingt vom Betriebssystem zur Verfügung gestellt werden.

Allgemeine Hinweise

- Entpacken Sie die Vorgabe für Aufgabe 2 im Unterverzeichnis `aufgabe2`. Erstellen sie im Unterverzeichnis `tests` für jede der Aufgaben eine Datei `aufgabeX.cc` $X=1,2,3$, in der Sie die einzelnen Aufgaben implementieren.
- Beantworten Sie die Zusatzfragen innerhalb der Dateien `aufgabeX.cpp`. Geben Sie jeweils deutlich an, auf welche Zusatzaufgabe sich Ihre Erläuterungen beziehen.
- Für etwaige Zeitmessungen verwenden Sie die Klasse `Timer` (siehe EZ-Stubs Einführung, Folien 31ff).
- In manchen Fällen sind Zufallszahlen hilfreich - in `debug/random.h` findet ihr die Schnittstelle eines Generators für Pseudozufallszahlen (Mersenne-Twister). **Wichtig:** Vor der Verwendung mit `random_init()` initialisieren.