

Grundlegende Übungen

Aufgabenblatt 2 - aperiodische Aufgaben

Fabian Scheler, Peter Ulbrich, Niko Böhm

11. November 2009

Ausgangspunkt für diese Aufgabenstellung ist ein rein zeitgesteuerter Ablaufplaner, der aperiodische Aufgaben im Hintergrund ausführen 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 Implementieren sie eine Ablauftabelle, in der die Aufgaben T1, T2 und T3 abgearbeitet werden. Die Anwendung soll auch die aperiodische Aufgabe T5 umfassen, die im Hintergrund bearbeitet werden soll. Aktiviert wird die Aufgabe T5 in einer Unterbrechungsbehandlung. **Hinweis:** Simulieren Sie das aperiodische Ereignis mit Hilfe eines `InterruptTimers` (siehe EZStubs Einführung, Folien 27 - 29 und 41), den Sie mit einem zufälligen Wert aufziehen.

- 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 die Bearbeitung dieser Aufgabe termingerecht sichergestellt werden kann. Man bezeichnet eine derartige Überprüfung als *Übernahmeprüfung*, skizzieren Sie, wie eine solche Übernahmeprüfung im vorliegenden Fall aussehen könnte. **Hinweis:** Sie dürfen auch weitere *Anforderungen* an das Betriebssystem stellen, die notwendig sind, um diese Übernahmeprüfung zu berechnen.
- Wie sollte sich das System verhalten, wenn die Übernahmeprüfung negativ verläuft? Die Übernahme verläuft negativ, wenn die Aufgabe also nicht zur Bearbeitung angenommen werden kann, weil nicht garantiert werden kann, dass der Termin eingehalten wird.

Aufgabe 2 Verbessern die Behandlung der aperiodischen Aufgabe, d.h. verkürzen Sie die Antwortzeit weiter. Verwenden Sie dazu ausschließlich die Mittel, die Ihnen der Ablaufplaner aus diesem Aufgabenblatt bietet.

- a) Zeigen Sie mit Hilfe einer Messung die negativen Konsequenzen dieser Verbesserung.
- b) Welches in der Vorlesung vorgestellte Verfahren vermeidet diese negativen Konsequenzen? **Hinweis:** Dieses Verfahren steht im Ablaufplaner aus Aufgabe 2 nicht zur Verfügung.
- c) 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

- Erstellen Sie im Unterverzeichnis `tests` der Vorgabe ein weiteres Unterverzeichnis `gue2`. Erstellen sie dort für jede der Aufgaben eine Datei `aufgabeX.cc` $X=1,2,3,4$, 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 27 - 29)).
- 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.