

Grundlegende Übungen

Aufgabenblatt 5 - Deferrable Server

Fabian Scheler, Peter Ulbrich, Niko Böhm

18. Januar 2010

Ausgangspunkt für diese Aufgabenstellung ist ein ereignisgesteuerter Ablaufplaner, der einen Deferrable Server unterstützt. Der Ablaufplaner hat dabei die Eigenschaften, wie sie in den Tafelübungen zu Aufgabe 5 besprochen wurden.

Aufgabe 1 Gegeben seien die vier Aufgaben in der folgenden Tabelle:

Aufgabe	p/a	Periode	Zwischenankunftszeit	WCET
T1	p	30 ms		8 ms
T2	p	20 ms		4 ms
T3	a	-	10 ms	4 ms
T4	a	-	10 ms	4 ms

Implementieren sie die Aufgaben T1 bis T4. Verwenden sie zur Aktivierung der periodischen Aufgaben T1 und T2 wie in den bisherigen Aufgaben einen Zeitgeber und vergeben sie die Prioritäten gemäß des *Rate Monotonic Algorithm*. Aktivieren sie die aperiodischen Aufgaben T3 und T4 ebenfalls mit Hilfe eines Zeitgebers, der in geeigneter Weise in zufälligen Abständen Unterbrechungen erzeugt. Behandeln sie T4 hierzu im Hintergrundbetrieb, dies kann simuliert werden, indem T4 mit der niedrigsten Priorität in diesem System versehen wird. T3 hingegen soll durch einen *Deferrable Server* behandelt werden. Dieser *Deferrable Server* hat ein Ausführungsbudget von 1 ms und eine Wiederherstellungsperiode von 10 ms. Der Server soll bei der Prioritätenvergabe gemäß des *Rate Monotonic Algorithm* mit einbezogen werden.

- Bestimmen sie mit Hilfe von Messungen jeweils die durchschnittlichen, die minimalen und die maximalen Antwortzeiten der Aufgaben T1 bis T4. Erläutern sie ihre Messergebnisse.
- Wie könnten sie die Antwortzeiten von Aufgabe T3 weiter verbessern? Welche Gefahr besteht dabei?
- Nehmen Sie an, die Aufgaben T1 und T3 würden auf ein gemeinsames Betriebsmittel zugreifen, das synchronisiert werden muss. T3 benötigt dieses Betriebsmittel für mindesten 1,5 ms. Welche Problematik entsteht hierdurch?

Allgemeine Hinweise

- Erstellen Sie im Unterverzeichnis `tests` der Vorgabe ein weiteres Unterverzeichnis `gue5`. Erstellen sie dort für jede der Aufgaben eine Datei

`aufgabe1.cc`, in der Sie die Aufgabe implementieren.

- Beantworten Sie die Zusatzfragen innerhalb der Dateien `aufgabe1.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.