

- **Echtzeitsysteme (EZS)**
- **Zeitpunkt:** Februar 2014
- **Dauer:** 20 Minuten (5 ECTS)
- **Prüfer:** Wolfgang Schröder-Preikschat
- **Beisitzer:** Isabella Stilkerich
- **Ergebnis:** sehr gut, faire Benotung

Vorbemerkung

Entspannte Atmosphäre. wosch stellt gezielte Fragen, manchmal etwas uneindeutig formuliert. Man bekommt aber weitere Hinweise, wenn man nicht gleich weiß was er meint.

Wie lassen sich Echtzeitsysteme anhand derer Termine untergliedern?

- × weiche Termine:
Terminüberschreitung wird toleriert, bedeutet aber konstante Abnahme des Nutzwerts
- × feste Termine:
Terminüberschreitung wird toleriert, Resultat nutzlos, Abbruch der Aufgabe
- × harte/strikte Termine:
Terminüberschreitung nicht tolerierbar, intransparent da sie zu Ausnahmebehandlung führt

Welche großen Klassifikationen von Echtzeitsystemen gibt es denn überhaupt?

Takt- und ereignisgesteuerte EZS.

Was sind da jeweils die Besonderheiten?

Bei taktgesteuert ist alles statisch. Ablaufplan steht a priori fest. Ablaufplan ist in gleich- oder verschieden große Zeitschlitze unterteilt. Behandlung von asynchronen Unterbrechungen ist bei Taktsteuerung nur bedingt möglich.

Bei ereignisgesteuert ist alles dynamisch. Ablaufplan steht a priori nicht fest, sprich Tasks werden anhand ihrer Priorität (statisch oder dynamisch) und ihrer Ankunftszeit eingeplant und eingelastet. Für die Behandlung von asynchronen Unterbrechungen stehen spezielle Bearbeitungskonzepte zur Verfügung (Zusteller, Unterbrecherbetrieb, Hintergrundbetrieb).

Wie könnte man denn taktgesteuerten EZS asynchrone Unterbrechungen behandeln?

Man könnte auch hier einen periodischen Zusteller einsetzen. Möchte man allerdings eine sehr kurze Reaktionszeit auf asynchrone Unterbrechungen (schnelle Bearbeitung), so muss dieser sehr hochfrequent eingeplant werden was im Ablaufplan leider viel Laufzeit verschwendet.

Was wären denn noch Vorteile von taktgesteuerten EZS gegenüber ereignisgesteuerten?

Durch das a priori Wissen kann der Ablaufplan quasi ohne zeitliche Beschränkungen optimiert werden. Da man die Aufgaben des Einplaners (scheduler) eigentlich vernachlässigen kann, fällt dieser auch in der Implementierung entsprechend einfach aus, was nicht nur Performanz bringt sondern auch bspw. Speicherplatz und Entwicklungszeit einspart.

Es können per Definition keine Blockierungen auftreten, da man entsprechende Verfahren einsetzt um potentielle Blockierungen a priori zu lösen!

Welche Möglichkeiten hat man denn bei taktgesteuerten EZS einen Ablaufplan zu gliedern?

Einsatz von Rahmen.

- × Rahmen muss mind. so groß wie die max. Ausführungszeit eines Tasks sein um dessen Verdrängung zu verhindern
- × Rahmen darf nicht größer als eine Periode eines Tasks sein um dessen rechtzeitige Auslösung zu garantieren
- × Rahmen muss zyklische Ausführung ermöglichen und somit die Hyperperiode teilen
- × Zwischen Auslösezeitpunkt und Termin muss mind. ein Rahmen passen
Man braucht ja 1 Rahmenanfang für die Einlastung und 1 Rahmenanfang für die Terminüberprüfung

Wie könnte man denn bei taktgesteuerten EZS unterschiedliche Situationen angehen?

Man erstellt für jede Situation eine separate Ablaufabelle. Bei einem Wechsel des Betriebsmodus wird dann einfach auf eine neue Tabelle atomar umgeschaltet.

(War wohl nicht prüfungsrelevanter Stoff laut Isa. Wusste es aber dennoch und hab die Frage halt mal mitgenommen)

Sie haben Zusteller erwähnt. Was gibt es denn da für welche?

Aufschiebbarer Zusteller der sein Budget in seiner Periode behält und nicht sofort verwirft wie der periodische Zusteller.

Das ermöglicht letztlich ein besseres Antwortzeitverhalten.

(Dann war die Zeit rum)