

Übung zur Vorlesung "Entwurf von Echtzeitsystemen"

Institut für Technische Informatik – TU Graz

6. Übung, 19.12.2014

Aufgabe 6.1 – Tasksynchronisation (Locking)

Warum und wann ist Tasksynchronisation nötig und was bedeuten Atomizität (atomic section) und Kritikalität (critical section) in diesem Zusammenhang? Erklären Sie das Problem anhand des folgenden Code-Beispiels:

```
// instructions before critical section
while (lock == 1); // OR yield();
lock = 1;

// critical section

lock = 0;

// instructions after critical section
```

Führt dieses Beispiel in jedem Fall zur korrekten Realisierung einer kritischen bzw. atomaren Sektion und wird dabei aktives Warten ("Spinlock") oder passives Warten verwendet? Wann ist es sinnvoll Spinlocks einzusetzen, wann nicht, und wie müsste der Code geändert werden, um die jeweils andere Variante zu realisieren?

Kann es unter Verwendung des gezeigten Konzepts durch mehrere Tasks zu Lifelocks oder Deadlocks kommen? Skizzieren Sie ggf. jeweils ein entsprechendes Szenario.

Aufgabe 6.2 – Tasksynchronisation (Producer-Consumer-Problem)

Erinnern Sie sich zurück an das in der Vorlesung besprochene Ereignissynchronisationsproblem: Ein Monitortask überwacht darin den Systemzustand und sendet bei Überschreitungen von Grenzwerten Nachrichten an einen Kommunikationstask. Diese werden in einem Puffer zwischen gespeichert.

Welche Problematik kann dazu führen, dass eine gepufferte Nachricht vom Kommunikationstask nicht weitergeleitet wird? Beschreiben Sie diese anhand des Beispiels aus der Vorlesung genauer.

Das zuvor betrachtete Problem enthält das Erzeuger-Verbraucher-Problem. Beschreiben Sie auch dieses genauer und präsentieren Sie Lösungsvarianten dazu.

Betreuer:

Prof. Dr. Marcel Baunach
(baunach@tugraz.at)

Sarah Haas
(sarah.haas@student.tugraz.at)

Wolfgang Schütz
(wolfgang.schuetz@student.tugraz.at)

Aufgabe 6.3 – Echtzeit-Programmiersprachen

Beschäftigen Sie sich mit den in der Vorlesung gezeigten sechs speziellen Anforderungen an Echtzeit-Programmiersprachen. Beschreiben Sie die einzelnen Anforderung und zeigen Sie welche Bedeutung den einzelnen Anforderungen in der Realität zukommt. Überlegen Sie sich hierzu ein fiktives Beispiel und zeigen Sie welche Auswirkung das Fehlen jeder speziellen Anforderung haben kann.

Aufgabe 6.4 – Aktivitätsorientierte Modelle

Sie nehmen mit einer Digitalkamera ein Bild auf. Dieses Bild wird in der Kamera geringfügig bearbeitet, z.B. Reduktion roter Augen, Entzerrung, Weißabgleich, etc. Stellen Sie den Ablauf von der Aufnahme des Bildes über die Bearbeitung bis hin zur Anzeige am Bildschirm als Datenflussgraph und Flussdiagramm dar. Sie können die Optimierungen, die am Bild vorgenommen werden, selbst wählen, es müssen aber mindestens 4 sein.

Aufgabe 6.5 – Multicore Scheduling

Abbildung 5 zeigt die Skizze eines Programmablaufs eines Echtzeitfähigen Programms mit einer Zykluszeit von 10ms. Um die Echtzeitfähigkeit des Programms zu garantieren, müssen alle Programmteile A bis J vor Erreichen der Deadline abgeschlossen werden.

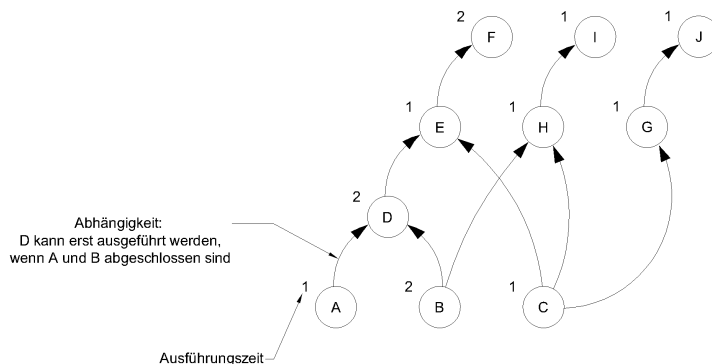


Abbildung 5: Beispiel eines Programmablaufs

- Skizzieren Sie den Programmablauf auf einer CPU bzw. mehreren CPU's.
- Wie viele CPU's werden benötigt?
- Wie viele CPU's machen Sinn?