BS // Fragebogen

Nebenläufigkeit

1. Warum ist nicht davon auszugehen, dass logische und arithmetische Ebene$_2$-Befehle (z.B. `decl` beim x86) mit direkter Adressierung der Speicherzellen unteilbar ausgeführt werden? Was bedeutet es, wenn die CPU in der Lage ist, atomare *read-modify-write*-Zyklen zu fahren?

2. Welche besondere Problematik ergibt sich vor dem Hintergrund asynchroner Programmunterbrechungen z.B. mit den C-Operatoren `--`, `-=` bzw. `-` im Zusammenhang mit RISC-Prozessoren?

3. Warum sind Zuweisungsoperatoren als kritisch anzusehen, wenn Programme neben 32- auch auf 16- und 8-Bit-Technologie laufen sollen?

4. Wieso kann die nebenläufige Ausführung von Prozessen denselben Laufzeitfehler produzieren wie die überlappte Ausführung eines unterbrochenen Programms mit einer Unterbrechungsbehandlungsroutine?

5. Wozu dient in C/C++ die Typerweiterung `volatile`? Was bedeutet dies für die Codegenerierung von Zugriffen auf „unbeständige“ Variablen? Welche Konsequenzen erwachsen dadurch speziell für Programme, die auf CISC-Prozessoren zur Ausführung kommen sollen?

6. Durch welche Maßnahme kann eine Schlangenimplementierung erzielt werden, bei der die Einfügeoperation in deterministischer Zeit erfolgt? Weshalb bzw. in welchen Szenarien ist solch eine Operation bedeutsam?

7. Sind durch die nebenläufige Ausführung hervorgerufene Probleme in Hochsprachenprogrammen auf transparenter Weise allein z.B. auf Assemblerebene lösbar?

8. Welchen Schlangenzustand hinterlassen die diskutierten Überlappungsszenarien und welche Konsequenzen erwachsen daraus, wenn es sich bei der Schlange z.B. um die *ready*-Liste des Schedulers handeln würde? Welche Ablaufbeispiele auf der Ebene der Prozessverwaltung können angeführt werden?

9. Welche Verfahren und Techniken bieten sich vor dem Hintergrund asynchroner Programmunterbrechungen an, um die Synchronisation kritischer Bereiche bewerkstelligen zu können? Weshalb scheidet z.B. das Konzept des Semaphors in diesem Zusammenhang aus?