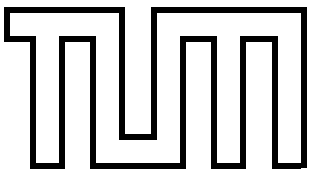


Technische Universität München

Fakultät für Informatik
Forschungs- und Lehreinheit Informatik VI

Übung zur Vorlesung Echtzeitsysteme

Einführung & Aufgabe 1-2

Simon Barner
barner@in.tum.de

Dominik Sojer
sojer@in.tum.de

Stephan Sommer
sommerst@in.tum.de

Wintersemester 2010/11

Übersicht über den voraussichtlichen Inhalt der Übung

1. Einführung in C unter VxWorks
2. Programmierung in Esterel / SCADE
3. Modellbasierte Entwicklung mit EasyLab
4. Code-Generierung mit OAW
5. Semaphore
6. Timer und Uhrensynchronization
7. Kugelfall-Versuch

Organisatorisches

Termine

	Zeit		Raum
Termin 1	Mo.	14:00-15:30	MI 03.05.012
Termin 2	Di.	08:30-10:00	MI 03.05.012
Termin 3	Mi.	10:15-11:45	MI 03.05.012
Termin 4	Mi.	14:00-15:30	MI 03.05.012

Tutoren

Simon Barner `barner@in.tum.de`
Dominik Sojer `sojer@in.tum.de`
Stephan Sommer `sommerst@in.tum.de`

Material

An den beiden folgenden Orten finden sich immer die aktuellen Versionen der Aufgabenblätter sowie von allen anderen Materialien.

- Webseite: http://www6.in.tum.de/Main/TeachingWs2010Echtzeitsysteme
- Freigabe: Als `Q:` verfügbar.
- Homeverzeichnis: Als `Z:` verfügbar. Legen Sie Ihre Dateien ausschließlich hier ab!

Aufgabe 1: Einführung in C unter VxWorks

Ziel

C ist eine weit verbreitete Programmiersprache, die sich durch Bibliotheken nahezu beliebig erweitern lässt. Die erste Aufgabe soll Ihnen helfen, sich in C einzuarbeiten und Ihnen einige Möglichkeiten und Besonderheiten nahe bringen, die bei der C-Programmierung unter VxWorks zu beachten sind. Zudem bietet die Aufgabe eine Einleitung in die Benutzung der POSIX-Bibliothek, die sie im Rahmen des Praktikums benutzen werden, und insbesondere in die Thread-Programmierung.
Eine Übersicht über die POSIX-Funktionalität sowie eine Einführung in VxWorks finden sie auf der Vorlesungs-Homepage.

Aufgabe

Verbessern Sie das fehlerhafte C-Programm `aufgabe1.c`, das sich im Verzeichnis `Uebung01` auf dem Laufwerk `Q:` befindet. Arbeiten Sie mit einer Kopie des Programms *aufgabe1.c* in Ihrem Projekt-Verzeichnis auf dem Netzlaufwerk `Z:`. Führen Sie das korrigierte Programm am Simulator aus. Die Aufrufreihenfolge der Prozesse hängt unter anderem von deren Priorität ab. Betrachten Sie deshalb bei dieser Aufgabe auch das Zusammenspiel der Prozesse unter Berücksichtigung der jeweils vergebenen Prioritäten!
Erläutern Sie zudem die einzelnen Programmschritte und fassen Sie die gefundenen Fehler zusammen. Senden Sie die verbesserte Version und die Programmbeschreibung mit einer Liste der Fehler an den Tutor.

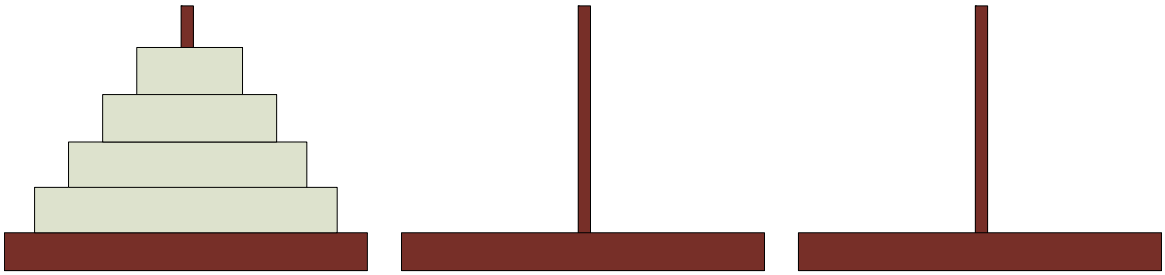
Implementierungshinweise

Zum Thema C-Programmierung gibt es eine fast unüberschaubare Vielfalt von Fachbüchern. Besonders empfehlenswert zum Erlernen von C ist die Lektüre von Kernighans & Ritchies *"The C Programming Language"*.
Einige Hinweise für die C-Programmierung in Verbindung mit VxWorks sind:

- die Startroutine muss nicht notwendigerweise `main` heißen
- (Anwendungs-)Routinen sind direkt aus der Kommandozeile aufrufbar, vorausgesetzt sie sind Programm-global (d.h. nicht `local` oder `static` deklariert)
- innerhalb eines Blockes dürfen Variablen nicht nach Anweisungen deklariert werden
- der doppelte Slash ist kein gültiges Kommentarzeichen
- Ausgaben von `printf`-Anweisungen bei Programmen mit nebenläufigen Prozessen erfolgen beim Start auf der *Host-Shell* nur teilweise und häufig nicht in der richtigen Reihenfolge. Ist die richtige Reihenfolge nötig, so ist ein Start aus der *Target-Shell* erforderlich.

Aufgabe 2: Türme von Hanoi

Ziel



Die Türme von Hanoi sind ein mathematisches Knobel- und Geduldsspiel bei dem ein Turm der Höhe n von der Stelle A nach Stelle C (mit Ablagestelle B) so transportiert werden, dass man immer nur eine Scheibe nehmen kann und niemals eine größere Scheibe über einer kleineren liegt. In der Vorlesung haben Sie bereits eine Implementierung mittels Ptolemy kennen gelernt. Ziel dieser Aufgabe ist es, die Kenntnisse der Programmiersprache C zu vertiefen und Sie mit der Entwicklungsumgebung *Wind River Workbench* vertraut zu machen.

Aufgabe

Sie finden ein Rahmenprogramm (*hanoi.c*) für diese Aufgabe im Verzeichnis `Uebung01` auf dem Laufwerk `Q:`.
Verwenden Sie zur Implementierung den nachfolgend im Pseudocode dargestellten Algorithmus.

```
funktion bewege (Zahl i, Stab a, Stab b, Stab c) {
    falls (i > 0) {
       bewege(i-1, a, c, b);
       verschiebe oberste Scheibe von a nach c;
       bewege(i-1, b, a, c);
    }
}
```

Testen Sie Ihr Programm für mehrere Turmhöhen.