

Betriebssysteme und Rechnerarchitekturen

LV 4112

Übungsblatt 3

13.04.12

In dieser Übung soll die μ Thread (Micro Thread) Bibliothek, erstellt werden, die ein kooperatives Multithreading auf Anwenderebene ermöglicht.

Auf der Webseite der Veranstaltung finden Sie dazu das Verzeichnis `uthread` als komprimierte tar-Datei. Entpacken Sie diese Datei auf Ihrem Rechner. Sie finden darin die Unterverzeichnisse `html`, `include`, `lib` und `man`, sowie die Dateien `main.c`, `Makefile` und `output.txt`.

Aufgabe 3.1 (Dokumentation):

- (a) Im den Verzeichnissen `man` und `html` finden Sie die Dokumentation der μ Thread Bibliothek. Diese wurde mit Hilfe des Tools *doxygen* aus den Headerdateien im Verzeichnis `include` automatisch generiert. Schauen Sie sich diese Headerdateien und die daraus generierte Dokumentation an. Überlegen Sie, wie Sie ggf. Änderungen am Quellcode so vornehmen können, dass die Dokumentation automatisch mitgepflegt wird.
- (b) Lesen und verstehen Sie die Dokumentation zur den Headerdateien.
 - Die Headerdatei `jmp.h` beschreibt eine erweiterte Form des aus der Vorlesung bekannten Funktionspaares `setjmp()/longjmp()`. Die (Assembler-)Implementierung dieser Funktionen findet sich in der Datei `jmp.c` im Verzeichnis `lib`. Diese Funktionen sollen der μ Thread Bibliothek intern als Mechanismen zur Thread-Umschaltung dienen.
 - Die Headerdatei `uth.h` beschreibt die Schnittstelle der μ Thread Bibliothek. Die Implementierung dieser Funktionen soll in die z.T. leere Datei `uth.c` im Verzeichnis `lib` eingefügt werden.
- (c) Das Programm `main.c` ist ein Testprogramm für die zu erstellende μ Thread Bibliothek. Lesen und verstehen Sie den Quellcode. Die Datei `output.txt` enthält die von diesem Programm erzeugte Ausgabe als Referenz.
- (d) Die μ Thread Bibliothek implementiert das so genannte *kooperative Multithreading*. Worin liegt der Unterschied zum *präemptiven* Multithreading, wie es die meisten heutigen Betriebssysteme implementieren?

Aufgabe 3.2 (Implementierung der μ Thread Bibliothek):

- (a) Worin liegt der Unterschied zwischen Multitasking und Multithreading? Welche Adressbereiche werden beim Multithreading von allen Threads gemeinsam genutzt, welche sind für jeden Thread privat zu halten?
- (b) Überlegen Sie sich eine oder mehrere Datenstrukturen, in der/denen die Zustände der von der μ Thread Bibliothek verwalteten Threads abgespeichert werden können. Welche Daten repräsentieren den vollständigen Zustand eines Threads?
- (c) Die Stacks der Threads sollten vorzugsweise im für Stack-Operationen vorgesehenen Adressbereich des umschließenden Anwenderprogrammes liegen. Wie können Sie in der Funktion `uth_init()` einen solchen Stack-Speicherbereich reservieren?
- (d) Implementieren Sie die noch fehlenden Funktionen der μ Thread Bibliothek in der

Datei `uth.c`. Überprüfen Sie Ihr Ergebnis, indem Sie das Testprogramm `main.c` gegen Ihre Bibliothek linken und die von diesem Programm erzeugte Ausgabe mit dem Inhalt der Referenzdatei `output.txt` vergleichen.

Aufgabe 3.3 (Erweiterungen):

- (a) Die Threads des Testprogrammes `main.c` terminieren sich jeweils selbst mit Hilfe der Funktion `proc_suicide()`. Was würde geschehen, wenn das nicht der Fall wäre, d.h. wenn sie zu ihrem Aufrufer zurückkehren würden?
- (b) Erweitern Sie Ihre μ Threads Bibliothek derart, dass ein solches Zurückkehren zum Aufrufer den Thread terminiert.
- (c) Überlegen Sie, ob/wie die μ Thread Bibliothek ggf. für präemptives Multithreading erweitert werden könnte.