michael.engel [☺] hs-coburg.de

Praktische Aufgaben zur Vorlesung
Eingebettete Betriebssysteme
Wintersemester 2017/18

Praktikumsaufgabe 4

Besprechung am Freitag, 24. November 2017

Hinweise: Praktikumsaufgaben müssen in der vorgegebenen Zeit in den eingeteilten Gruppen bearbeitet werden. Die Lösung der Aufgaben wird bewertet, jedes Gruppenmitglied soll Fragen zu den einzelnen Aufgaben beantworten können. Bei Unklarheiten und Rückfragen wird selbstverständlich eine Hilfestellung gegeben.

4.1 Präemptives Multitasking

Bauen Sie Ihren kooperativen Scheduler in einen präemptiven Scheduler um.

a. Implementieren Sie einen Timer-Interrupt via SYSCLK (Initialisierung, Handler). Der Interrupt-Handler kann entweder direkt das Scheduling übernehmen oder aber einen PendSV-Interrupt triggern, der die eigentliche Prozessumschaltung vornimmt (mit separatem PendSV-Handler können Sie wieder Policy – also Schedulingentscheidung – von der Implementierung der Prozessumschaltung trennen).

   Hinweis: Ihre Funktion zum Umschalten des Kontextes/Stacks kann mit Änderungen wiederverwendet werden.

b. Implementieren Sie Systemfunktionen zum dynamischen Erzeugen und Beenden von Prozessen. Können Sie einen Unix-artigen `fork()`-Systemaufruf damit erstellen? Wenn ja, welche Einschränkungen müssen Sie hier in Kauf nehmen und warum?

4.2 Gerätetreiber

Erweitern Sie Ihr Betriebssystem um einen Gerätetreiber für die seriellen Schnittstellen (UART) 0 und 2 des LPC1778. UART0 ist die Verbindung zum PC, an UART2 können Sie z.B. eine serielle Tastatur anschließen. Implementieren Sie dazu eine device-Switch-Tabelle und leiten Sie Zugriffsfunktionen wie `open()`, `read()`, `write()` und `close()` über diese an den entsprechenden Treiber weiter.

4.3 Shell

Über UART0 können Sie mit dem PC kommunizieren. Realisieren Sie eine über die serielle Schnittstelle steuerbare einfache Shell, mit der Sie Prozesse erzeugen, abbrechen usw. können.