

Echtzeitbetriebssysteme — Übung

Oliver Jack

Ernst-Abbe-Hochschule Jena
Fachbereich Elektrotechnik und Informationstechnik

Sommersemester 2024



Ernst-Abbe-Hochschule Jena
University of Applied Sciences

Übung 3: Tasks

1 VxWorks Tasks, Prioritäten

Taskerzeugung und -aktivierung

- VxWorks erlaubt mit der Routine `taskSpawn()` die Erzeugung eines neuen Tasks.
- Syntax:
`id = taskSpawn(name,priority,options,stacksize,function, arg1,...,arg10);`
- Die Argumente zu `taskSpawn` sind der Taskname (ASCII-Zeichenkette), die Priorität, ein Hex-Wert für die Task-Optionen, die Stackgröße (int), die Hauptroutinenadresse (name der Funktion) und bis zu zehn Argumente der der Hauptroutine.

Beispiel

```
#include <stdio.h>
#include "taskLib.h"

#define ITERATIONS 10

void print(void);

void spawn_ten() /* Subroutine to perform the spawning */
{
    int i;
    TASK_ID taskId;
    for(i=0; i < ITERATIONS; i++) /* Creates ten tasks */
        taskId = taskSpawn("tprint",90,0x100,2000,(FUNCPTR) print,
                           0,0,0,0,0,0,0,0,0,0);
}

void print(void) /* Subroutine to be spawned */
{
    printf("Hello, I am task %d\n",taskIdSelf()); /* Print task Id */
}
```

Aufgaben

- 1 Verändern Sie die Priorität einheitlich für alle zehn Tasks so, dass sie niedriger ist als die Priorität des Tasks `spawn_ten`. Finden Sie zunächst heraus, was die höchste und was die niedrigste Priorität für VxWorks-Tasks ist. Was ändert sich an der Ausgabe?
- 2 Verändern Sie die Funktion `print()` so, dass sie Argumente übergeben bekommt und diese Argumente ausgibt.
- 3 Übergeben Sie die Argumente in der Routine `taskSpawn()`.
- 4 Weisen Sie jedem der zehn Tasks eine andere Priorität zu und beobachten Sie das Ergebnis des Programmlaufs. Ändert sich die Reihenfolge der Ausgabe?