

Echtzeitbetriebssysteme — Übung

Oliver Jack

Ernst-Abbe-Hochschule Jena
Fachbereich Elektrotechnik und Informationstechnik

Sommersemester 2025



Ernst-Abbe-Hochschule Jena
University of Applied Sciences

Übung 2: Threads

1 Erzeugung und Terminierung von Threads

2 Aufgabe

POSIX Threads

```
/* POSIX: neuer Thread im selben Adressraum */
int pthread_create (
    pthread_t *tid, /* Thread-Id */
    pthread_attr_t *attr, /* UL- oder KL-Thread, */
                        /* Stackposition und -groesse, */
                        /* Schedulinginfo */
    void * (*start) (void *), /* Startfunktion */
    void *arg) /* Argument */

/* POSIX: Terminierung eines Threads */
void pthread_exit ( void *retval )
```

Aufgabe: POSIX Threads

Bringen Sie folgendes Thread-Test-Programm unter VxWorks zum Laufen

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#define NUM_THREADS 5

void *PrintHello(void *threadid)
{
    int tid;
    tid = (int)threadid;
    printf("Hello World! It's me, thread # %d!\n", tid);
    pthread_exit(NULL);
    return 0;
}
```

Das brauchen wir für den Programmen

Aufgabe: POSIX Threads (Forts.)

```

int main(int argc, char *argv[])
{
    pthread_t threads[NUM_THREADS];
    int rc, t;
    for(t=0; t<NUM_THREADS; t++){
        printf("In main: creating thread %d\n", t);
        rc = pthread_create(&threads[t], NULL, PrintHello, (void *)t);
        if (rc){
            printf("ERROR; return code from pthread_create() "
                   "is %d\n", rc);
            exit(-1);
        }
    }
    pthread_exit(NULL);
}

```

← 5 stücke (oben definiert)

Threadterminierung aus einer anderen Task

Zur Beendigung einzelner Threads, außer mit `return` oder `pthread_exit`, gibt es noch die Funktion

```
int pthread_cancel(pthread_t thread);
```

Sie beendet den Thread mit der ID von `thread`.
Benutzung durch Einstellen des Abbruchstatus:

```
int pthread_setcancelstate(int state, int *oldstate);
```

Voreingestellt ist der Status `PTHREAD_CANCEL_ENABLE`, d. h. Abbruch eines Threads aus einem anderen Thread ist erlaubt. Verboten ist dies im Status `PTHREAD_CANCEL_DISABLE`

Aufgabe: Threadterminierung

↑
คำสั่งทำ

- Schreiben Sie ein Programm, das drei Threads erzeugt.
- Einer erzeugt eine Zufallszahl, die anderen zwei Threads reagieren entsprechend auf diese Zufallszahl.
- Je nach dem, ob die Zufallszahl kleiner bzw. größer als 25 ist, beendet der eine Thread den anderen mit `pthread_cancel`.