Beispiele zu Kapitel 2: Nebenläufigkeit (Concurrency)

Aus: Alan Burns, Andy Wellings: *Real-Time Systems and Programming Languages. Ada, Real-Time Java and C/Real-Time POSIX*. Addison Wesley, 2009. (Kapitel 4)

Beispiel 2-1: C/Real-Time POSIX-Schnittstelle für Thread-Attribute

Program 4.4 A C/Real-Time POSIX interface to thread attributes.

```
typedef ... pthread_t;  /* details not defined */
typedef ... pthread_attr_t;
typedef ... size_t;

int pthread_attr_init(pthread_attr_t *attr);
  /* initializes a thread attribute pointed at by attr to
     their default values */

int pthread_attr_destroy(pthread_attr_t *attr);
  /* destroys a thread attribute pointed at by attr*/

int pthread_attr_setstacksize(pthread_attr_t *attr,
                              size_t stacksize);
  /* set the stack size of a thread attribute */

int pthread_attr_getstacksize(const pthread_attr_t *attr,
                              size_t *stacksize);
 /* get the stack size of a thread attribute */

int pthread_attr_setdetachstate(pthread_attr_t *attr,
                                int detachstate);
  /* set the detach state of the attribute */

int pthread_attr_getdetachstate(const pthread_attr_t *attr,
                                int *detachstate);
  /* get the detach state of the attribute */

int pthread_attr_setguardsize(pthread_attr_t *attr,
                              size_t guardsize);
  /* set the guard size of a thread attribute */

int pthread_attr_getguardsize(const pthread_attr_t *attr,
                              size_t *guardsize);
 /* get the guard size of a thread attribute */

 ...
/* other attributes associated with scheduling */

/* Unless otherwise stated, all the above integer functions
   returns 0 if successful, otherwise an error number is returned
*/
```

Beispiel 2-2: C/Real-Time POSIX-Schnittstelle für Threads

Program 4.5 A C Real-Time POSIX interface to threads.

```
typedef ... pthread_t;  /* details not defined */
typedef ... pthread_attr_t;

int pthread_getconcurrency();
  /* returns the last set value of pthread_getconcurrency */

int pthread_setconcurrency(int level);
  /* sets the application's preferred thread concurrency level;
     returns the old level */

int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
                   void *(*start_routine)(void *), void *arg);
  /* create a new thread with the given attributes and call the
     given start_routine with the given argument */

int pthread_join(pthread_t thread, void **value_ptr);
  /* suspends the calling thread until the named thread has
     terminated, any returned values are pointed at by value_ptr */

void pthread_exit(void *value_ptr);
  /* terminate the calling thread and make the pointer value_ptr
     available to any joining thread */

int pthread_detach(pthread_t thread);
  /* the storage space associated with the given thread may be
     reclaimed when the thread terminates */

pthread_t pthread_self(void);
 /* return the thread id of the calling thread */

int pthread_equal(pthread_t t1, pthread_t t2);
  /* compare two thread ids
     return non 0 if equal, 0 otherwise */

int pthread_atfork(void (*prepare)(void), void(*parent)(void),
                   void (*child)(void));
  /* used for managing the resources shared by a multi-threaded
     program when a fork is performed */

/* Unless otherwise stated, all the above integer functions
   returns 0 if successful, otherwise an error number is returned
*/
```