

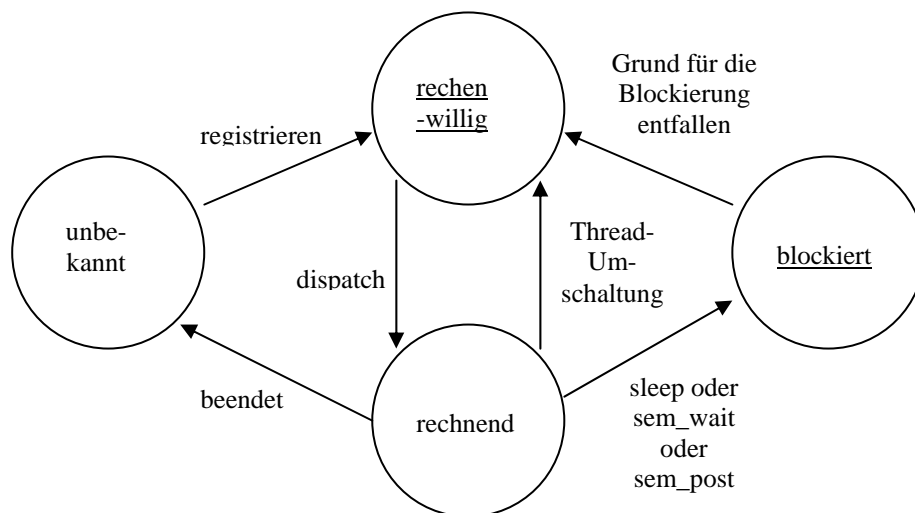
Überlegungen zur Realisierung eines rudimentären Multi-Threading-Betriebssystems mit Echtzeit-Scheduling

In der ersten Übungsaufgabe ist ein rudimentäres Betriebssystem(MTBS) für einen Mikrocontroller der 8051-Familie (hier 80C515C) mit den folgenden Eigenschaften zu realisieren:

- Multi-Threading-Betrieb mit einer vorgegebenen Anzahl (MAXTHREADS) von Threads
- Echtzeit-Scheduling: der (nicht blockierte) Thread der höchsten Priorität erhält die Rechenzeit zugeteilt; bei mehreren Threads der selben Priorität wird die Rechenzeit nach dem Zeitscheibenverfahren zugeteilt
- es sind die folgenden System-Calls zu realisieren:

```
sleep(t);          t in Einheiten von ms, kleinstes t z.B. 10 ms
sem_init(&s, n);    s ist eine Struktur vom Typ sem_t, n der
                   Initialisierungswert
. sem_wait(&s);
  sem_post(&s);
```

Dem MTBS liegt das folgende Thread-Zustandsdiagramm zu Grunde:



Die unterstrichenen Zustände "rechenwillig" und "blockiert" werden in Listen verwaltet.

Bei der Realisierung des MTBS sind die folgenden Aspekte zu beachten:

- Ein Thread mit der niedrigsten Priorität wird als idle-Thread registriert (möglichst "transparent" für den Nutzer des MTBS)-
- Es kann sinnvoll sein, eine Liste der rechenwilligen Threads, eine Liste für die durch sleep-Aufruf blockierten Threads und jeweils eine Liste der auf einen bestimmten Semaphor blockierten Threads zu verwalten.

- Aus Gründen der Speicherplatz- und Laufzeiteffizienz ist es sinnvoll, die Listen in Feldern fester Länge (entsprechend MAXTHREADS) zu verwalten. Die jeweilige Liste ist durch einen Listenkopf (z.B. für jede der Listen eine Variable des Typs `unsigned char`, wenn MAXTHREADS 255 ist) erreichbar, in dem die Thread-Id (TID) des ersten Elements der Liste steht. Ist die z.B. durch `#define LEER 255` vereinbarte Konstante im Listenkopf eingetragen, weiß man, dass die Liste leer ist.
- Da ein Thread sich jeweils nur in einem Zustand des Thread-Zustandsdiagramms befinden kann, ist es möglich, die Listen "innerhalb" der Thread-Control-Blöcke (TCB) zu verwalten. Dazu enthält der Inhalt des Listenkopfes (einer nicht leeren Liste) den Index des TCBs des ersten Threads in dieser Liste. In der TCB-Struktur wird ein Element `next` eingeführt, in dem der Index des nächsten Threads der Liste oder `LEER` steht.
- Die blockiert-Listen der Semaphore müssen ebenfalls nach Thread-Prioritäten sortiert sein.
- Der Code des MTBS steht im Wesentlichen innerhalb der Interrupt-Service-Routine für den Timer0, durch den zyklisch (z.B. alle 10 ms) Interrupts ausgelöst werden (Timer-Ticks).
- Es ist sinnvoll, zwischen dem Zeitintervall dieser Timer-Ticks und einer Zeitscheibe (z.B. 100 ms) zu unterscheiden.
- Scheduling-Punkte sind das Auftreten eines Timer-Ticks und die zu realisierenden System-Calls.
- Zur Realisierung der System-Calls wird eine System-Call-Schnittstelle realisiert, die über einen "Software-Interrupt" (durch Setzen des TF0 auf 1) gegebenenfalls den MTBS-Code aktiviert.