Aufgabenblatt 1
Echtzeitsysteme

Institut: Berliner Hochschule für Technik
Dozent: Prof. Dr. Christian Forler
Url: `https://lms.bht-berlin.de/`
Email: `cforler(at)bht-berlin.de`
SoSe 2022

Aufgabe 1 (16 Punkte) 2-Zustands-Prozessmodell Simulator
Erstellen Sie in C-Programm welches die Änderung von Prozesszuständen innerhalb des 2-Zustands-Prozessmodells simuliert. Gehen Sie dazu wie folgt vor.

1. Erstellen Sie eine Datei `process.h` mit den folgenden Inhalten
   - Eine Enum `enum state` welches die beiden Werte `READY` und `RUNNING` annehmen kann.
   - Ein Verbund `struct process` welches über die Member `uint32_t p_id` und `enum state p_state` verfügt.
   - Den Funktionsprototypen `void p_switch_state(struct process *p)`. Die Implementierung soll den Zustand eines Prozesses von `READY` auf `RUNNING` bzw. von `RUNNING` auf `READY` verändert.
   - Den Funktionsprototypen `void p_print(struct process *p)`. Die Implementierung soll den übergeben Prozesses `p` in menschenlesbarer Form ausgeben.
2. Schreiben Sie ein Programm, welches die implementierten Funktionen testet.
3. Implementieren Sie die Warteschlangen (`queue`) als Verkette Listen. Gehen Sie dazu wie folgt vor.
   (a) Erstellen Sie ein Datei `queue.h` mit den folgenden Inhalten.
      - Ein Verbund `struct q_node` mit den folgenden Membern: `struct q_node *next` und `struct process *p`.
      - Ein Verbund `struct queue` mit den folgenden Membern: `struct q_node *start` und `struct q_node *end`.
      - Den Funktionsprototyp `void q_add(struct queue *q, struct process *p)`. Bei dieser Funktion soll der Prozess `p` der Warteschlange `q` hinzugefügt werden. Der Prozess wird damit das letzte Element in der Warteschlange.
      - Funktionsprototypen `struct process *q_remove(struct queue *q)`. Die Implementierung soll das erste Element der Warteschlange `q` entfernen und zurückgeben.

- Den Funktionsprototypen `void q_print(struct queue *q)`. Die Implementierung soll die übergebene Warteschlange `q` in menschenlesbarer Form ausgeben.

(b) Erstellen Sie ein Datei `queue.c` welche die Funktionsprototypen aus `queue.h` implementiert.

(c) Schreiben Sie ein Programm `queuedemo.c` welche Ihre Implementation testet und ein Makefile welches Ihr Testprogramm baut.

4. Implementieren sie das 2-Zustands-Prozessmodell Simulator indem Sie wie folgt vorgehen.

   (a) Erstellen Sie eine Headerdatei `processmodel.h` mit den folgenden Komponenten

   - Ein Verbund `struct pctx` mit den folgenden Membern: `struct queue *qready` und `struct process *running` verfügt.
   - Eine Funktionsprototyp `void print(struct pctx *ctx)`. Die Implementierung soll den Kontext `ctx` in menschenlesbarer Form ausgeben.
   - Eine Funktionsprototyp `void step(struct pctx *ctx)`. Die Implementierung soll den momentan laufende Prozess `running` in die Warteschlange `qready` hinzugefügt und der erste Prozess in `qready` wird zum neuen laufenden Prozess.

   (b) Erstellen Sie eine Demoanwendung welches die Zustandsübergänge des 2-Zustands-Prozessmodells mit 10 Prozessen simuliert. Nach der Initialisierung soll der Prozess-Kontext durch eine neue Zustandsänderung modifiziert und ausgegeben werden.