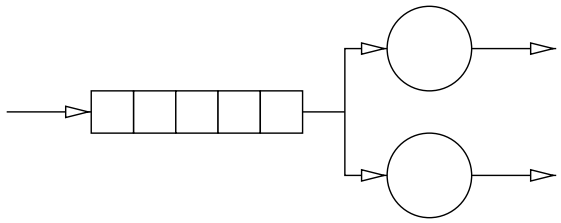
6 Übungsaufgabe #6: Mehrprozessorfähiger Scheduler

In der vorigen Aufgabe wurde ein einfacher, preemptiver Scheduler implementiert, welcher keine Fadenmigration zwischen verschiedenen Prozessoren erlaubt. Dieser Scheduler soll nun so erweitert werden, dass eine globale Bereitliste für alle CPUs genutzt wird und so keine Bindung zwischen Faden und Prozessor mehr besteht. Diese Liste soll als FIFO-Liste ausgelegt werden und konkurrierende Listenoperationen erlauben (Siehe Abbildung).



Die Synchronisierung der Liste soll einmal durch gegenseitigen Ausschluss (Spinlocks) und durch den Einsatz einer nichtblockierend synchronisierten Liste erfolgen.

6.1 Synchronisierung durch gegenseitigen Ausschluss

Die erste Variante der Bereitliste soll durch gegenseitigen Ausschluss synchronisiert werden. Hierzu soll die Elementaroperation TAS (Test and Set) der Assemblerebene der jeweiligen Plattform genutzt werden.
Durch die Nutzung von Test and Set lassen sich Spinlocks sehr einfach implementieren:

```
while(TAS(lock));
```

6.2 Nichtblockierende Synchronisierung

In der Vorlesung am 29.6.2009 wurde eine nichtblockierend synchronisierte FIFO-Liste besprochen. Diese soll als zweite Variante der Bereitliste genutzt werden. Diese Liste benötigt eine architekturspezifische CAS (Compare and Swap) Operation.

6.3 ABA-Problem

Die in der Vorlesung vorgestellte FIFO-Liste ist für das ABA-Problem anfällig. Sollte dies im Scheduler auftreten, soll die Bereitliste durch den Einsatz eines Generationszählers (wie bei dem in der Vorlesung vorgestellten Wartestapel) verbessert werden.

6.4 Testprogramm

Um die Implementierung des mehrprozessorfähigen Schedulers zu testen, sollen $2 * CPU + 1$ Fäden ohne Angabe einer CPU gestartet werden. Diese sollen an einer für jeden Faden eindeutigen Bildschirmposition folgende Daten ausgeben:

- Thread-ID
- aktuelle CPU-ID
- aktuelle Länge der Bereitliste

Das Scheduling soll wie in Aufgabe 5 präemptiv erfolgen.

Aufgaben:

→ Implementierung von Spinlocks mit *TAS*

→ Implementierung einer durch gegenseitigen Ausschluss synchronisierten Bereitliste

→ Implementierung der in der Vorlesung vorgestellten NBS FIFO-Liste als Bereitliste

→ Auftreten des ABA-Problems überprüfen und evtl. durch Einführung eines Generationszählers (*wheel*) vermeiden

→ Test der Implementierungen durch ein einfaches Testprogramm

Hinweise:

- Auf Intel bietet sich die Instruktion `BTS` (Bit Test and Set) mit `LOCK`-Prefix zur Implementierung des Spinlocks an.
- Die Instruktion `CMPXCHG` (Compare and Exchange) mit `LOCK`-Prefix soll auf Intel als *CAS* genutzt werden.

6.5 Abgabe: am 09.07.2009