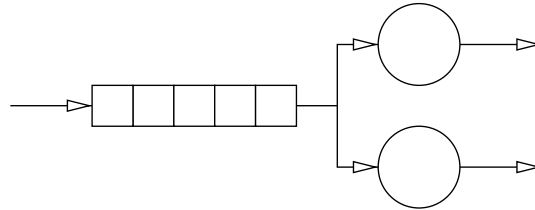


6 Übungsaufgabe #6: Mehrprozessorfähiger Scheduler - Teil 1

In der vorigen Aufgabe wurde ein einfacher, präemptiver 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).



In Aufgabe 6 soll die Synchronisation der globalen Bereitliste durch gegenseitigen Ausschluss (Spinlocks) erfolgen. Dabei sollen zwei verschiedene Varianten eines Spinlocks implementiert werden. In *kconfig* soll dann über eine Konfigurationsoption die konkrete Spinlockvariante ausgewählt werden können.

6.1 Implementierung eines einfachen Spinlocks

Als erste Variante soll ein einfacher Spinlock implementiert werden. Dies lässt sich durch die Nutzung von *Test and Set* sehr einfach realisieren:

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

6.2 Implementierung eines verbesserten Spinlocks

Als zweite Variante soll folgende verbesserte Version des Spinlocks implementiert werden:

```
do {
    while (lock == 1);
} while(TAS(lock));
```

Was macht diese Version effizienter als die einfache Version aus 6.1?

6.3 Synchronisation durch gegenseitigen Ausschluss

Die Bereitliste soll nun konfigurierbar durch eine der beiden obigen Varianten mittels gegenseitigem Ausschluss synchronisiert werden. Hierzu soll die Elementaroperation TAS (Test and Set) der Assemblerebene der jeweiligen Plattform genutzt werden.

6.4 Testprogramm

Zum Testen der Implementierung des mehrprozessorfähigen Schedulers, 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 beider Spinlockvarianten mit *TAS*
- Konkret verwendete Spinlockvariante mittels *kconfig* konfigurierbar machen
- Implementierung einer durch gegenseitigen Ausschluss synchronisierten Bereitliste
- 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.

6.5 Abgabe: am 15.07.2010