Systemnahe Informatik

Übungsgruppe Xeon Phi

Dominik Walter

Sommersemester 2018

Mutex

Beispiel

```
// eventuell parallel
pthread_mutex_lock(&mutex);
// garantiert sequentiel
pthread_mutex_unlock(&mutex);
// eventuell parallel
```

Semantik

Warte bis kritischer Bereich betreten werden kann.

Verwendung

Schutz einer gemeinsamen Variablen / Datenstruktur

Condition-Variable: Teil 1

Beispiel

```
// eventuell parallel
pthread_mutex_lock(&mutex);
while (condition) {
        pthread_cond_wait(&cond, &mutex);
}
// garantiert sequentiel und !condition
pthread_mutex_unlock(&mutex);
// eventuell parallel
```

Semantik

Warte bis spezifische Bedingung erfüllt wurde.

Verwendung

Vermeidung von spinlocks (z.B. für queue::pop())

Condition-Variable: Teil 2

Beispiel

```
// eventuell parallel
pthread_mutex_lock(&mutex);
// garantiert sequentiel
condition = false;
pthread_mutex_unlock(&mutex);
// eventuell parallel
pthread_cond_signal(&cond);
```

Semantik

Bedingung wurde geändert und muss von allen Threads erneut überprüft werden.

Semaphore: Teil 1

Beispiel

```
// eventuell parallel
sem_wait(&sem);
// garantiert maximal n threads
// (-> sem_init(&sem, 0, n))
sem_post(&sem);
// eventuell parallel
```

Semantik

Warte bis "kritischer" Bereich (n threads gleichzeitig) betreten werden kann.

Verwendung

Beschränkung der Parallelität

Semaphore: Teil 2

Beispiel: Consumer-Thread

```
sem_wait(&sem);
//Ressource ist vorhanden
pthread_mutex_lock(&m);
y = queue.pop();
pthread_mutex_unlock(&m);
```

Beispiel: Producer-Thread

```
pthread_mutex_lock(&m);
queue.push(x);
pthread_mutex_unlock(&m);
sem_post(&sem);
//Ressource ist vorhanden
```

Semantik

Warte bis eine abstrakte Ressource verfügbar ist. Der eigentliche Zugriff muss allerdings separat geschützt werden!

Verwendung

Producer-Consumer Modell

Barriere

Beispiel

```
// t_id: thread-ID
// Phase 1: Globales Lesen
local = buffer[t_id + 1];
pthread_barrier_wait(&barrier);
// Phase 2: Lokales Schreiben
buffer[t_id] += local;
pthread_barrier_wait(&barrier);
```

Semantik

Warte bis n threads die Barriere erreicht haben.

Verwendung

Unterteilung in verschiedene Phasen (z.B. Bulk-Synchronisation)