

Übungen zur Vorlesung
"Paralleles Höchstleistungsrechnen"

Prof. Dr. P. Bastian, Ch. Engwer

Abgabe bis 24. 11. 2008 an christian.engwer@iwr.uni-heidelberg.de

ÜBUNG 1 PETERSON LOCK

Implementieren Sie folgendes parallele Programm mit Hilfe der P-Threads:

```
parallel ThreadCounter
{
  int counter;
  int iterations=1e7;
  process  $\Pi$  [int  $p \in \{0, \dots, P-1\}$ ] {
    for (int i=0; i<iterations; i++) {
      // lock holen
      ...
      // kritischer Abschnitt
      counter++;
      // lock freigeben
      ...
    }
  }
  print(count);
  print(P*iterations);
}
```

Verwenden Sie zur Absicherung des kritischen Abschnitts das Peterson Algorithmus für P Prozesse (Skript 4.1.4).

Experimentieren Sie mit verschiedenen Anzahlen an Threads (z.B. 2...8) und verschiedenen Optimierungseinstellungen des Compilers (z.B. unoptimiert, -O3, ...). Wenn möglich testen Sie auch auf etwas "exotischeren" Plattformen als i386.

Welches Verhalten erwarten Sie von dem Programm, diskutieren Sie das beobachtete Verhalten.

Hinweis: Zur Implementierung können Sie entweder direkt P-Threads verwenden, oder die in der Vorlesung eingeführten ThreadTools.

ÜBUNG 2 SEMPHORE

In der Vorlesung wurde das Konzept der Semaphore besprochen (Skript 4.3). Implementieren Sie eine Semaphore aufbauend auf dem Konzept des Mutex. Verwenden Sie zur Implementierung entweder direkt die P-Threads oder die ThreadTools.