Betriebssysteme, Übung 9

Prof. Dr. Jan Dünnweber

Verteilte Systeme und Betriebssysteme

1. Folgen Sie den einzelnen Schritten aus dem Tutorial zum Sun RPC
   - Führen Sie das Beispielprogramm zur Fernabfrage der Systemzeit (bzw. zur Umwandlung in einen String) auf je zwei Laborrechnern aus

`http://www.cs.rutgers.edu/~ pxk/rutgers/notes/content/ra-sunrpc.pdf`

2. Programmieren Sie eine neue Version der Matrixmultiplikation mit PThreads

Fassen Sie Elementblöcke zu Untermatrizen zusammen

$$c_{row,col} = \Sigma_{i=1}^{n} a_{row,i} * b_{i,col}$$

   - Verwenden Sie zur Parallelisierung das in der Vorlesung vorgestellte Modell *Workcrew*

3. Wie müssen IDL, Client und Server aussehen, wenn die parallele Matrixmultiplikation mittels Sun RPC im Netzwerk bereit gestellt werden soll?

- Mit folgendem Code können Sie die Laufzeit einer Subroutine in C messen

Mikrosekundengenaue Messung

```
#include <sys/time.h>
struct timeval  tv1, tv2;  /* ... */
gettimeofday(&tv1, NULL);
  /* do some stuff ... */
gettimeofday(&tv2, NULL);
printf ( "Total time = %f seconds\n",
         (double) (tv2.tv_usec - tv1.tv_usec) / 1000000 +
         (double) (tv2.tv_sec - tv1.tv_sec) );  /* ... */
```

- Verwenden Sie das Code-Fragment für Messungen der Laufzeit der parallelen und der verteilten Versionen (mit PThreads & RPC) der Matrixmultiplikation

  ($\rightarrow$ Code im Moodle)

- Folgendes C-Programm enthält eine Race Condition

Data Race

```
#include <pthread.h>
int var = 0;
void* child_fn ( void* arg ) {
  var++;
return NULL;  }
int main ( void ) {
  pthread_t child;
  pthread_create(&child, NULL, child_fn, NULL);
  var++;
pthread_join(child, NULL); return 0; }
```

- Erstellen Sie ein Profil mit `valgrind -tool=helgrind` und analysieren Sie die Ausgabe für dieses Programm und die Bsp. aus der letzten Übung ($\rightarrow$ Blatt 8)