

Lösungswege zum Aufgabenpaket 7

Aufgabe 7.1

Das C-Programm zur Berechnung zweier Matrizen könnte wie folgt aussehen:

```

/* matrixmultiplikation */
/* Directive SPP -> Shared Memory */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <sys/time.h>
#include <sys/times.h>
#define SIZE 700

float a[SIZE] [SIZE];
float b[SIZE] [SIZE];
float c[SIZE] [SIZE];
int nprocs;

void init_matrixx()

{
  int i, j;
  for (i = 0; i < SIZE; i++) {
    for (j = 0; j < SIZE; j++) {
      a[i][j] = (float)i + j;
      b[i][j] = (float)i - j;
    }
  }
}

void matmul()

{
  int i, j, k;

#ifdef SPP
#pragma _CNX loop_parallel(threads,ivar=i)
#endif

  for (i = 0; i < SIZE; i++) {
    for (j = 0; j < SIZE; j++) {
      for (k = 0; k < SIZE; k++)
        { c[i] [j] += a[i] [k] * b[k] [j]; }
    }
  }
}

main ()
{

```

```

int          nprocs;                /* zahl der parall. prozesse */
int          i;
time_t      runt1,runt2;
struct tms  tm;
double      cputime, runtime;

nprocs = 1;
time(&runt1);
runtime = (double) 0.;
cputime = (double) 0.;

init_matrixx();

#ifdef SPP                          /* beginn der parallelisierung für den spp -
rechner */
#pragma _CNX no_parallel          /* dieser teil wird nicht parallelisiert */
#endif                             /* ende der parallelisierung */

for (i=0;i<1;i++)
    { matmul(); }

time(&runt2); runtime += difftime(runt2,runt1);
times(&tm); cputime = (tm.tms_utime+tm.tms_stime)/CLK_TCK;
printf("\nRun Time = %15.2lf [s]\n",runtime);          /* ausgabe der run time */
printf("\nCPU Time = %15.2lf [s]\n\n",cputime);        /* ausgabe der cpu time */
}

```

Anmerkung:

Um das Programm nun skalar zu übersetzen, gibt man folgende Zeile ein:
 cc -O -Ae <Name des C-Files> -o <Name des ausführbaren Files>

Damit das Programm mit mehreren Prozessoren ausgeführt werden kann, benutzt man folgende Zeile:
 Cc -Ae +O3 -DSPP +Oparallel +Onodonsel +Oreport +Onoautopar +pa <Name des C-Files> -o <Name des ausführbaren Files> -lcps

Mit Hilfe folgender Zeile bestimmt man nun vor dem Programmaufruf die Anzahl der verwendeten Prozessoren:

```

mpa -m -min <minimal verwendete Prozessoren> -max <maximale Anzahl
verwendeter Prozessoren> <Name des ausführbaren Files>

```

Aufgaben 7.2 - 7.3

Anzahl der verwendeten Prozessoren	Run-time In s	CPU-time in s
1	81	81
2	38	75
4	21	80
6	14	76
8	12	75
Skalar	66	64