

Übung 2

Kurs: Parallele und verteilte eingebettete Systeme
SS 2017
VAK: 03-ME-712.06
Veranstalter: Dr. Stefan Bosse

Aufgabe 1: Matrixmultiplikation mit OCCAM

Lesen sie die Folien im Abschnitt F9-F15.

1. Implementieren sie den sequenziellen und die verschiedenen parallelen Partitionierungen des Matrixmultiplikationsalgorithmus in OCCAM.

- A.) Sequenzieller Fall

```
VAR A,B,C: ARRAY [1 TO 10,1 TO 10] OF FLOAT;
FOR J = 1 TO 10 DO
  FOR I = 1 TO 10 DO
    FOR K = 1 TO 10 DO
      C[I,J] := A[I,K] * B[K,J];
    DONE
  DONE
DONE
```

- B.) Parallele Partitionierung Fall 1.

```
Object barrier: semaphore;
Process VE[i,j]:
  FOR k := 1 to q DO
    C[i,j] ? C[i,j] + A[i,k] * B[k,j];
  END FOR k;
  barrier.up();

Process MAIN:
  barrier.init(0);
  FOR ALL i,j := 1 to p,1 to r DO START Process VE[i,j];
  FOR N := 1 to (p*r) DO barrier.down();
```

- C.) Parallele Partitionierung Fall 2.

```
Object barrier: semaphore;
Process VE[j]:
  FOR i := 1 to p DO
    FOR k := 1 to q DO
      C[i,j] ? C[i,j] + A[i,k] * B[k,j];
    END FOR k;
  END FOR i;
  barrier.up();

Process MAIN:
  barrier.init(0);
  FOR j := 1 to r DO START Process VE[j];
  FOR N := 1 to r DO barrier.down();
```

● D.) Parallele Partitionierung Fall 3.

```
Object barrier, barrier_t[p,q]: semaphore;
Process VE[i,j,k]:
  t[i,j,k] + A[i,k] * B[k,j];
  barrier_t[i,j].up();
-> Ein VE[i,j,k] ist Master
Process VE[i,j] ? V[i,j,k]:
  barrier[i,j].init(0);
  FOR N = 1 to q DO
    barrier_t[i,j].down();
  END FOR N;
  FOR k = 1 to q DO
    C[i,j] ? C[i,j] + t[i,j,k];
  END FOR k;
  barrier.up();
Process MAIN:
  barrier.init(0);
  FOR ALL i,j,k := 1 to p, 1 to q, 1 to r DO
  START Process VE[i,j,k];
  FOR N := 1 to (p*q) DO barrier.down();
```

2. Testen Sie die verschiedenen Implementierung auf Korrektheit indem sie geeignete Testfälle erzeugen.
3. Führen Sie die Programme mit dem jserver aus und versuchen Sie die Laufzeit für die verschiedene Fälle zu messen. Dabei sollten entsprechend große Matrizen und ein wiederholte Ausführung der Berechnung verwendet werden um geeignete Rechenzeiten > 100ms zu erzielen. Protokollieren Sie die Laufzeiten. Gibt es Unterschiede? Worin würde eine erhöhte Laufzeit bei der parallelen Berechnung begründet sein?