Übungsblatt 1- Lösungen

Multikern-Praktikum Wintersemester 06-07
Aufg. 1: Process vs. Thread

• Processes:
  • also called **heavy-weight** process
  • each process has its own environment
    ➔ change of process results in change of environment
  • creation and changes to other processes are expensive
    ➔ performance loss
  • costs of inter-process communication are higher than for communication between threads

• Threads:
  • also called light-weight process
  • threads share a common environment
    ➔ have access to the same address space
    ➔ change of thread does not require change of environment
  • creation is less expensive than for processes
Aufg. 1: Process vs. Thread

Abbildung 1: Workstation PC

Abbildung 2: Shared memory machine
Aufg. 2: Data Parallelism

a) Problem: loop carried values (d)
   for (i=0; i < n; i++){
      d=i*(i+1)/2;
      a[i] = b[i] + c[d];
   }
Aufg. 2: Data Parallelism

a) Problem: recurrences (i, i+m)
   • For \( m \geq n \), there are no data dependencies within this loop. A parallelization of the loop is possible.
   • For \( m = 1 \) or \( m = -1 \), due to the data dependencies between consecutive iterations, no parallelization is possible.
   • For the rest of the cases, only parts of this loop can be parallelized.

```c
for (i=1; i<n; i++){
    a[i] = a[i+m] + b[i];
}
```
Aufg. 2: Data Parallelism

a) Problem: indirect addressing (d[i])
   • As long as there are no duplicates among the values from the d array, a parallelization of the loop is possible.
   • Otherwise, extra measurements have to be taken, depending on the concrete situation.

```c
for (i = 0; i<n; i++){
    a[d[i]] = a[d[i]] + b[i];
}
```