Algorithms and Programming IV
Recap: Concepts of Programming

Summer Term 2024 | 15.04.2024
Claudia Müller-Birn, Barry Linnert
Our Approach for an Efficient Solution for Problems

- **Problem**: Essential part of the solution of a problem. Very creative phase.
- **Algorithm**: Programming or coding the algorithm in a specific programming language. Easier phase.
- **Software in a Programming Language**
Need for Higher Programming Languages

a = b + c

```
mov -0x8(%rbp),%eax
mov -0x4(%rbp),%edx
add %edx,%eax
mov %eax,-0xc(%rbp)
```

```
ldr r2, [fp, #-8]
ldr r3, [fp, #-12]
add r3, r2, r3
str r3, [fp, #-16]
```
From Machine Languages to Higher Programming Languages

Machine Languages

Assembler Languages

Higher Programming Languages

Declarative Languages

Imperative Languages
Classification According to Programming Paradigm

Higher Programming Languages

- Declarative Languages
  - Functional Languages
  - Logical Languages
  in ALP 1

- Imperative Languages
  - Object-oriented Languages
  in ALP 2

What?

How?
Classification According to Programming Paradigm

Higher Programming Languages

We translated an algorithm in a sequential order.

Individual operations are executed one after the other in the sequence of instructions clearly defined in the source code. You defined thread.

- Declarative Languages
  - Functional Languages
  - Logical Languages

- Imperative Languages
  - Object-oriented Languages

in ALP 1

in ALP 2
What is a different Approach for Implementing an Algorithm?

• We can implement an algorithm non-sequentially.

• An algorithm is called non-sequential when the linear order of its operations is replaced by a non-linear order. We create multiple threads at the same time.

• However, these threads can have
  - A shared memory using one CPU -> concurrent algorithms
  - Shared or distributed memory using more than one CPU -> parallel algorithms
  - Neither shared memory nor CPU -> distributed algorithms
Our Course Perspective

• The programming model is derived from the machine model. We explain existing mechanisms and algorithms depending on the programming model. For these, we discuss existing problems and present possible solutions.

• The course is divided into three major areas of non-sequential programming:
  − Concurrent programming: Machine with one CPU, but with a common memory
  − Parallel programming: Machine with several CPUs with or without shared memory machine, which requires message exchange
  − Distributed programming: Different machines