

Script generated by TTT

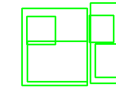
Title: Petter: Compiler Construction (23.04.2020)
- 01: Scanner

Date: Tue Apr 21 14:44:41 CEST 2020

Duration: 14:55 min

Pages: 4

Topic:



Lexical Analysis

The Lexical Analysis - Siever

Classified tokens allow for further pre-processing:

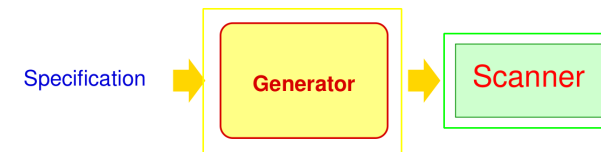
- **Dropping** irrelevant fragments e.g. **Spacing**, **Comments**,...
- **Collecting Pragmas**, i.e. directives for the compiler, often implementation dependent, directed at the code generation process, e.g. **OpenMP**-Statements;
- **Replacing of Tokens** of particular classes with their meaning / internal representation, e.g.
 - **Constants**;
 - **Names**: typically managed centrally in a **Symbol**-table, maybe compared to reserved terms (if not already done by the scanner) and possibly replaced with an index or internal format (⇒ **Name Mangling**).

3/49

The Lexical Analysis

Discussion:

- Scanner and Siever are often combined into a single component, mostly by providing appropriate callback actions in the event that the scanner detects a token.
- Scanners are mostly not written manually, but **generated** from a specification.



4/49

The Lexical Analysis - Generating:

... in our case:

