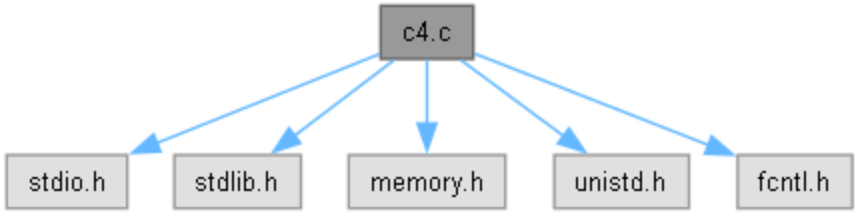
C4 Compiler Architecture

1. Overall Architecture



Shows the main components and their relationships

2. Data Flow and Components

Main Components:

- **Lexer (next function):** Converts source code to tokens
- **Parser (expr/stmt):** Builds abstract syntax tree
- **Virtual Machine:** Executes instructions
- **Symbol Table:** Manages identifiers and types
- **Stack:** Handles function calls and local variables

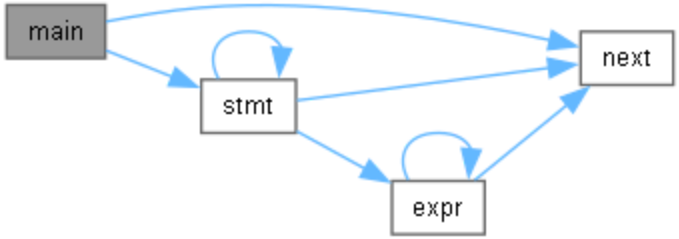
Data Flow:

1. Source Code → Tokens (via lexer)
2. Tokens → Abstract Syntax Tree (via parser)
3. AST → Instructions (via code generation)
4. Instructions → VM Execution

Key Data Structures:

- **Tokens:** tk (token type), ival (token value)
- **Instructions:** e array (emitted code)
- **Stack:** sp (stack pointer), bp (base pointer)
- **Symbol Table:** sym array (identifiers and types)
- **Data Segment:** data array (global variables)

3. Component Interactions



Shows how components interact in the main function (e.g. lexer, parser, VM)

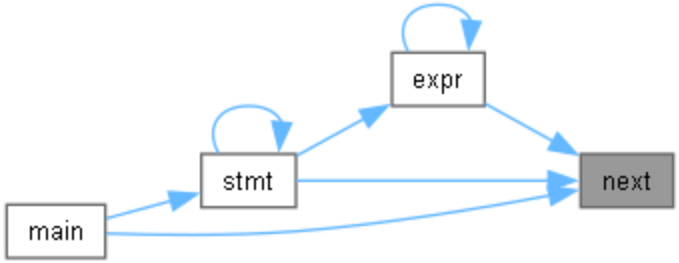
4. Lexical Analysis Flow

Token generation and symbol table interaction (e.g. identifiers, types)

5. Parsing Process

*Expression parsing and instruction generation (e.g. +, -, *, /)*

6. Statement Processing



Control flow and statement handling (e.g. if, while, return)