CSN-352 Compiler: Assumptions and Specifications

Group 18

April 6, 2025

Introduction

This document outlines the key assumptions and specifications implemented in our custom compiler. These design choices reflect the current limitations and supported features of our compiler's architecture.

Assumptions and Specifications

1. Arrays inside `struct` types must be accessed using standard pointer semantics.
2. Multidimensional arrays allocated on the stack are treated as pointers to their base type (`type*`).
3. Array sizes must be known at compile time. Dynamic array sizing is not supported.
4. `pointer` sizes are by default taken for 64-bit architecture.
5. Any `type*` (pointer to a type) can be assigned to stack-allocated arrays of that type.
6. Nested `struct`s are not supported at this stage.
7. Function calls support array parameters only in the following form: `int fun(int arr[], ...)`. Passing multidimensional arrays to functions is not supported.
8. Due to single pass nature of the compiler, named types (`typedef`) don't work immediately after their declaration.

Notes

- These specifications are subject to change in future versions as the compiler matures.
- Limitations mentioned above are intentional design decisions to simplify parsing, semantic analysis, and code generation in the current stage.