# VIREX

**CPU**

**STACK MEMORY**

**RAM**

**VMCALL**

USER

execute orin with appropriate commandline arguments

execute sasm with appropriate commandline arguments

Compile ORIN to x86

Compile ORIN to SASM

Disassemble SASM Code

Assemble SASM Code

Execute SM Code

Exit

TUI

Validate executable Format

instruction remaining?

Refresh Windows

Execution instruction

is step execution?

Refresh Windows

## INDEX

Components

Execution Options

Execution Pipeline

Algorithmic Steps

## SASM

**COMPILE TIME FUNCTIONS**

**OPCODE LUT**

**MEMORY ALLOCATOR**

Load executable in

dump instruction details

USER

Disassemble SASM Code

Assemble SASM Code

CLI

**EXPRESSION PARSING**

**STATEMENT PARSING**

**LINE PARSING**

**FILE PARSING**

**ASSEMBLY**

---

### ASSEMBLY

- create and push a "global" scope
- "translate" the input file
- pop scope
- resolve all unresolved operands
- resolve entry point address
- create executable

(a) Assembly Pipeline

---

### FILE PARSING

- Create a Lexer class object
- load input file into lexer
- convert these lines into blocks of statements
- Translate the Code block
- generate an AST for current file

(b) File Parsing

---

### LINE PARSING

- Line remaining?
- Discard comments
- Identify line type
- Parse statement from line
- Add Statement to Code Block
- move to next line

(c) Line Parsing

---

### EXPRESSION PARSING

- Load the string into tokenizer
- tokens remaining?
- Fetch next token
- convert text into valid expressions

(e) Expression Parsing

---

### STATEMENT PARSING

- is directive?
- is Label?
- is Instruction?
- entry
- bind
- include
- Scope
- convert the label name from string to expression format
- process the instruction body into expressions

  NOTE: the structure for an instruction line consists of 'instruction name' and 'body'

- process the expression from the passed directive body
- process the binding name and value expressions from the passed directive body
- process the expression from the passed directive body
- convert lines into CodeBlock until end directive is found

  NOTE: The structure of a directive statement consists of a head ptr to a linked list of statements, and as such, the entire block will be treated as a single statement of type 'local scope'

(d) Statement Parsing

---

## 📦 Project Structure

```
/docs/                    # Reference documentation
/examples/                # Sample programs
/include/                 # Public headers for VM, SASM, OCC
/src/                     # Core implementation (VM, assembler,
compiler)
/tests/                   # Simple Test programs written in SASM
/tools/themes/vs_code/    # VS Code syntax highlighter
/install.sh               # Install script for linux
```

## Tech Stack

- **Programming Language:** C
- **Version Control:** Git
- **Build System:** GNU Make
- **AST VISUALIZER:** Graphviz