

**VIREX** (VIRtual EXecuter) is a platform-independent virtual machine designed around a flexible intermediate language called **SASM** (Simulated Assembly). It's inspired by the **Java Virtual Machine (JVM)**, but unlike JVM bytecode, SASM is **open, readable, and writable** — you can program directly in it.



## What is SASM?

Just like Java compiles to bytecode for the JVM, any language can be compiled into SASM for VIREX. The difference is:

- SASM is **assembly-like**, human-readable, and editable.
- SASM is **open**, letting anyone build tools and languages around it.

You can even create your own programming language that compiles into SASM and runs anywhere VIREX runs — making your language instantly portable.



## Why SASM?

- Learn how **assembly-level code** works through a clean and simplified syntax.
- Build a **compiler** without worrying about machine-level code generation.
- Make your own language **platform-independent** by targeting SASM.



---

## 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
```



---

## Getting Started (LINUX)

### 1. Clone this repo:

```
git clone https://github.com/Soham-Metha/virex.git
cd virex/
```

### 2. Build the project (requires **sudo**):

```
./install.sh
```

### 3. Run an example program:

```
cd ./examples/SASM/  
virex
```

If the **TUI doesn't render properly**, try adjusting your **terminal font size**.


If that doesn't help, you can tweak layout values in **src/VM/vm\_tui.c::CreateWindows()**.  
The constants used are defined as **percentages** of the screen dimensions.

P.S. **kitty terminal** config, and font used, are available in **/tools**

### 4. Inside VIREX, do the following:

- Select **"Run SASM/ORIN command with custom flags"**
- Enter the following command:


```
-i helloWorld.sasm -I ./ -o tmp.sm
```

 use **Arrow keys** for navigation in menu.

- Select **"SASM build and exec"** by pressing **'a'**
- Enter the output filename (**tmp.sm**)

### 5. Activate the syntax highlighter in VS Code

- Open VS Code
- Press **Ctrl + Shift + P**
- Type: **Preferences: Color Theme**
- Select: **Palenight+sasm**

 Open any **.sasm** file in vs code to see the syntax highlighter at work!



## Want to Contribute?

### We're actively building:

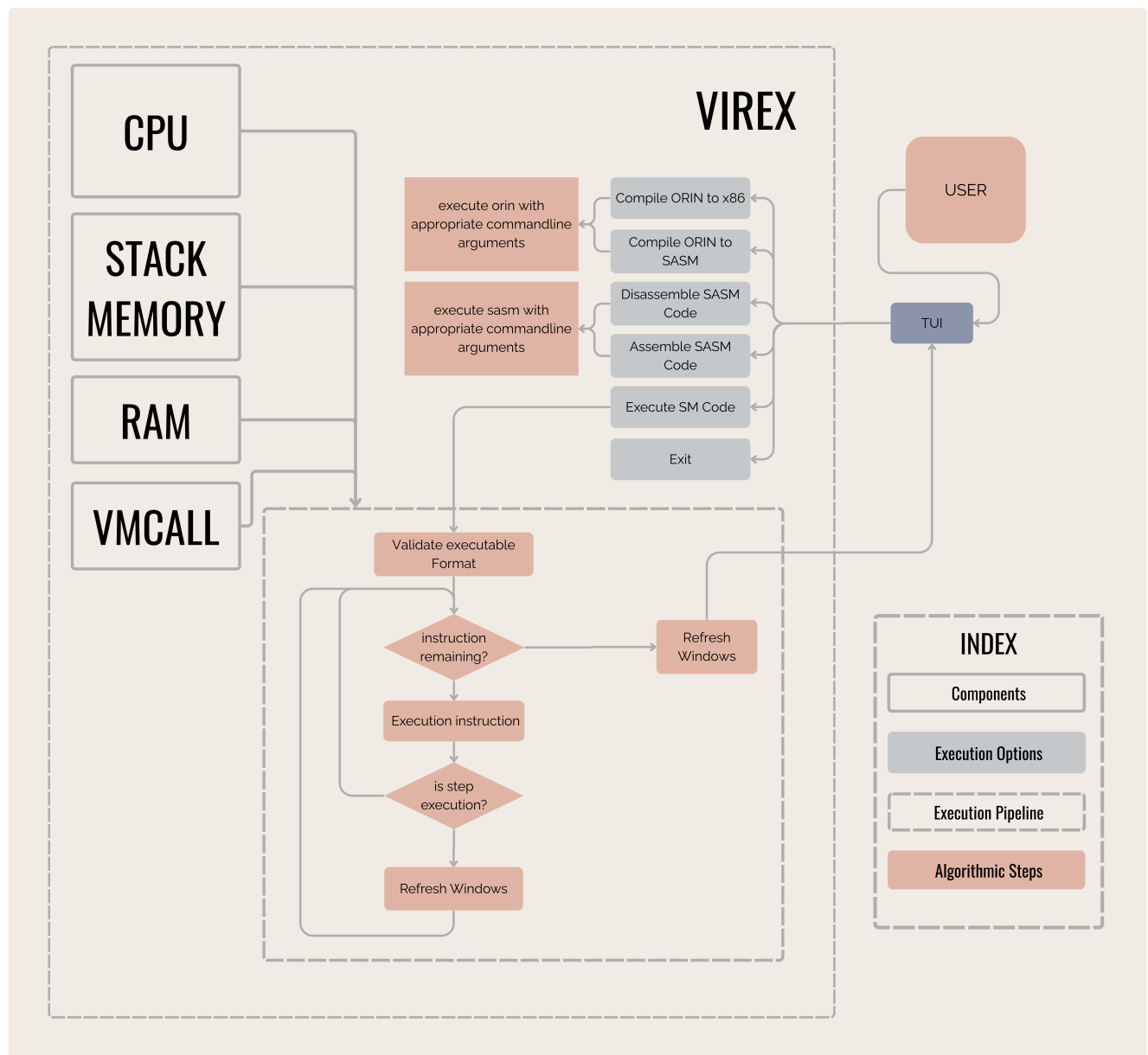
1. The **ORIN programming language**
2. Improved **SASM tooling** (UI, debuggers, optimizers, etc.)
3. Expanded **Documentation** and **tutorials**

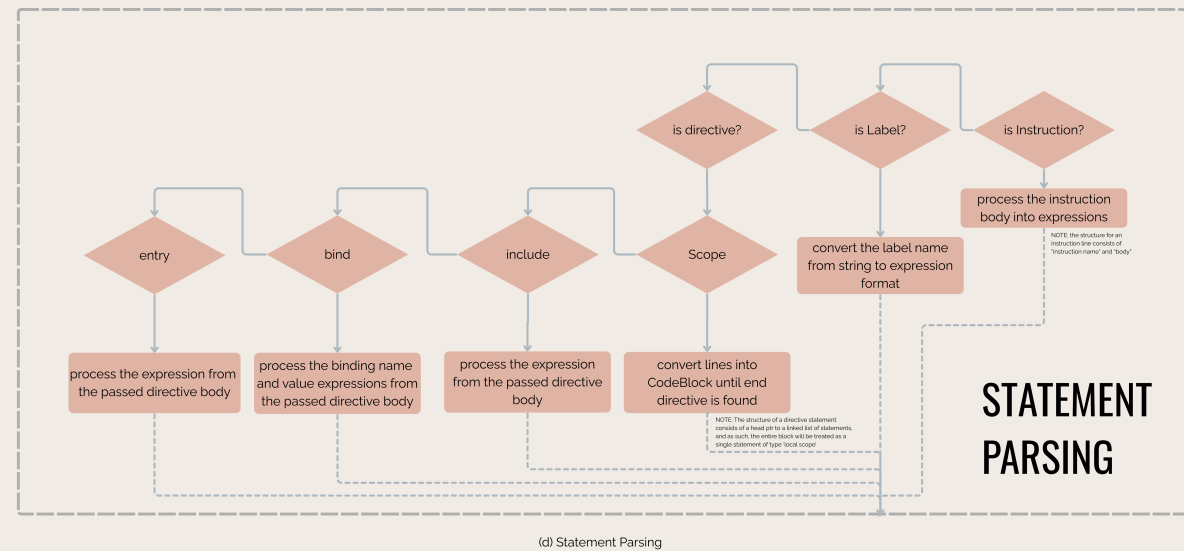
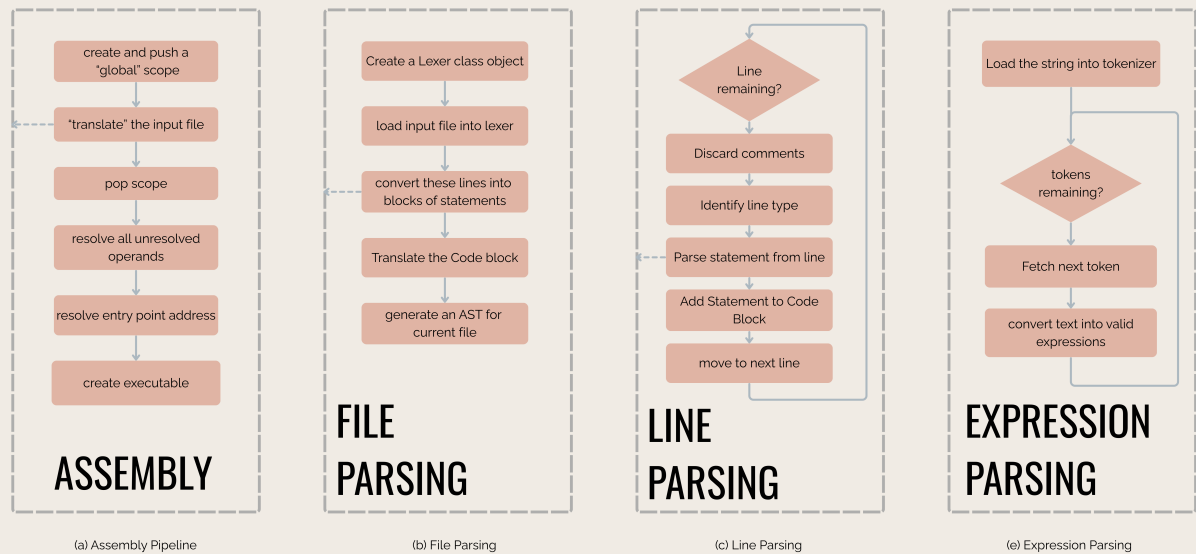
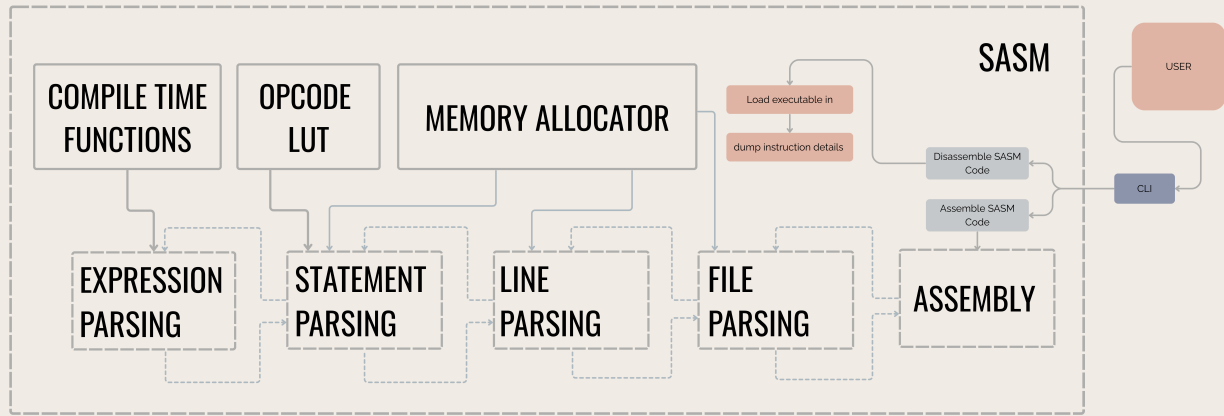
!!! info inline end ""



The image displays the 'Virtual Executor' application window, which has a dark gray background and a light gray title bar. The window is divided into several functional panels. At the top, there are three main sections: 'REGISTERS', 'OUTPUT', and 'MEMORY'. The 'REGISTERS' panel on the left lists various CPU registers (R0-R15, PC, SP) and their current values. The 'OUTPUT' panel in the center shows the execution progress, including the binary search for the number 5 and the current instruction (33: RET). The 'MEMORY' panel on the right displays a memory dump in hexadecimal and ASCII. Below these are three smaller panels: 'PROGRAM' showing the current instruction (CALL 8), 'VIREX' with a large 'VIRTUAL EXECUTOR' logo, 'INPUT' for entering the SM file path, and 'CREDITS' listing the developers and contributors. The interface is clean and modern, with a focus on readability and ease of use.

# System Design and Architecture





## Tech Stack

- **Programming Language: C**

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

## Maintainers

Tool	Maintainer
VIREX, SASM	Soham Metha
AST visualizer	Soham Metha
Syntax Highlighter	Soham Metha
ORIN Compiler	Omkar Jagtap
Core lib(Hashtable)	Omkar Jagtap
Core libs(other)	Soham Metha

---

## References

- [Tsoding](#)
  - [Dr Birch](#)
  - [Low Byte Productions](#)
  - [Cobb Coding](#)
-