Sprint 3 Reflection: WASM as OS

Team: CTRL 5 **Members Present (Sprint 3 Meeting):** Ololade Awoyemi, Shivam Sakthivel Pandi, Biraj Sharma, Benjamin Wilson, Sritan Reddy Gangidi **Date:** December 4, 2025

Accomplishments (Tasks Completed and Team Contribution)

Rust Runtime and Parser: Benjamin Wilson
Parser Tests: Benjamin Wilson
Design Document Design Rationale and Wasm Engine Class Diagram: Benjamin Wilson

Key Milestones Achieved:

- Rust WASM Parser Created
- Rust Wasm Runtime VM Created
- Integrated Fronted and Backend
- Updates for Design Doc

What Went Well:

- Frontend integration performed at Sprint meeting
- Wasm Parser parses valid wasm files
- Wasm Runtime supports i32 opcodes
- Tests that were present during meeting worked
- All members attended meeting

What Could be Improved:

- Frontend Backend integration has some bugs
- Wasm Parser needs improved error handling
- Wasm Runtime needs to support more opcodes
- Member tardiness to meeting
- Team members need to speak a little more confidently
- Design document needs to be updated in a more timely manner
- Client insight into project
- Requirements doc needs updating
- Not all Tests passed

Plans for Sprint 3

- Frontend and Backend Integration with no bugs

- JSON logging for Backend to Frontend
- Support for all WASM Opcodes in VM Runtime
- Add limited support for import functions through an import table
- Maybe Basic Scheduling.