`eval`-based interpreter in Stak Scheme

@raviqqe

July 21, 2024

Contents

- Stak Scheme
- Progress
- The `eval`-based interpreter
- Future work

Stak Scheme

- A bytecode compiler and virtual machine (VM) for Scheme
  - The compiler is written in Scheme.
  - The VM is written in Rust.
- It aims to support R7RS-small.

Progress

- An `eval`-based interpreter
  - Faster startup time
- The `process-context` library

The `eval`-based interpreter

The `stak` command

- A Stak Scheme interpreter
- `cargo install stak`

Previous architecture

1. AOT compile of source codes into bytecodes
2. Run the bytecodes on a VM

Current architecture

1. Run bytecodes of `eval`-based interpreter written in Scheme

Interpreter in Scheme

```
(import
  (scheme base)
  ; ...
  (scheme eval))

(define (main)
  (define program (open-input-file (list-ref (command-line) 1)))

  (do ()
    ((eof-object? (peek-char program))
      #f)
    (if (char-whitespace? (peek-char program))
      (read-char program)
      (eval (read program) (interaction-environment)))))

(main)
```

Demo

Future work

- Faster startup time
- Faster `cargo install` time
  - Build scripts?
- More compatibility
  - `(scheme time)` library

Summary

- Building an interpreter!