mio Documentation

Release 0.0.1

James Mills

September 20, 2014

Contents

Release 0.0.1

Date September 20, 2014

CHAPTER 1

About

CHAPTER 2

Minimal I/O Interpreter

This is a minimal I/O Interpreter. This is a rewrite of:

- https://bitbucket.org/prologic/mio-lang

> **Warning:** mio is a new programming language in early **Development**.
> DO NOT USE IN PRODUCTION!
> USE AT YOUR OWN RISK!

2.1 Prerequisites

It is recommended that you do all development using a Python Virtual Environment using virtualenv and/or using the nice virtualenvwrapper.

```
$ mkvirtualenv mio
```

You will need the RPython toolchain to build the interpreter. The easiest way to do this is to My Fork of PyPy which includes a convenient `setup-rpython.py` to make working with the RPython toolchain a bit easier.

```
$ hg clone https://bitbucket.org/prologic/pypy
$ cd pypy
$ python setup-pypy develop
$ python setup-rpython.py develop
```

2.2 Installation

Grab the source from https://bitbucket.org/miolang/mio and either run `python setup.py develop` or `pip install -r requirements.txt`

```
$ hg clone https://bitbucket.org/miolang/mio
$ cd mio
$ pip install -r requirements.txt
```

2.3 Building

To build the interpreter simply run `mio/main.py` against the RPython Compiler. There is a `Makefile` that has a default target for building and translating the interpreter.

```
$ make
```

As of 690c894 you can now build mio using Docker and fig.

```
$ fig build
```

2.4 Usage

You can either run the interpreter using Python itself or by running the compiled interpreter `mio` in `./bin/mio`.

```
$ ./bin/mio examples/hello.mio
```

Untranslated running on top of Python (*CPython*):

```
$ miopy examples/hello.mio
```

As of 690c894 you can now run mio using Docker and fig.

```
$ fig run mio hello.mio
```

2.5 Grammar

The grammar of mio is currently as follows:

```
program = expressions

expressions = { expression }

expression = message | terminator

message = symbol | arguments | symbol arguments

arguments = T_LPAREN arguments_list T_RPAREN |
            T_LBRACE arguments_list T_RBRACE |
            T_LBRACKET arguments_list T_RBRACKET

arguments_list = expressions | expressions T_COMMA arguments_list

symbol = T_IDENTIFIER | T_OPERATOR | T_NUMBER | T_STRING

terminator = T_TERMINATOR
```

CHAPTER 3

Documentation

3.1 mio package

3.1.1 Subpackages

mio.objects package

Submodules

mio.objects.builtins module

mio.objects.cfunction module

mio.objects.message module

mio.objects.number module

mio.objects.object module

mio.objects.string module

Module contents

3.1.2 Submodules

mio.ast module

mio.bytecode module

mio.compiler module

mio.interpreter module

mio.lexer module

mio.main module

mio.objspace module

mio.parser module

mio.registry module

class mio.registry.**Registry**
Bases: object

populate(*obj*, *space*)

register(*alias=None*)

mio.tokens module

mio.utils module

mio.utils.**unquote_string**(*s*)

3.1.3 Module contents

3.2 Change Log

- : ...

3.3 Road Map

Here's a list of upcoming releases of mio in order of "next release first".

Each bullet point states a high level goal we're trying to achieve for the release whilst the "Issues List" (*linked to our Issue Tracker*) lists specific issues we've tagged with the respective milestone.

Note: At this stage we don't have any good estimates for our milestones but we hope we can improve this with future releases and start adding estimates here.

3.3.1 mio 0.1

- Methods, Exceptions
- Number, String, Object
- Documentation
- Test Suite

See also:

mio 0.1 milestone

3.4 TODO

Manually maintained ToDo List.

- Fix handling of constant/literal values being parsed, their bytecode and the way they are handled during runtime interpreter evaluation.
- Message tree rewriting.
  - `a = 1` -> `setattr("x", 1)`
  - `del a` -> `delattr("x")`
- Message tree operator shuffling
  - `1 + 2 * 3` -> `(1 + (2 * 3))`
- Implement all bytecode instructions.
- Implement the runtime environment and core objects.
- Test Suite
- Documentation
- Examples
- Debugger
- Code Formatter
- Package Manager

CHAPTER 4

Indices and tables

- *genindex*
- *modindex*
- *search*

m