Myrddin: Sucklesscon 2016

Ori Bernstein

September 24, 2016

Topics

The Overview

The Language

The Libraries

Building Code

The Future

Are there any questions?

A New Programming Language

- Keep the Spirit of C
- But try to improve
- Simple, understandable, cohesive
- Batteries Included.

Major Features

- Imperative language with a hint of functional.
- Simple, small, statically compiled
- Zero external dependencies
- Imperative Language
- Native Compilation
- Parametric Polymorphism
- Manual memory management
- Unicodely-correct
- An exercise in NIH.

Supported Platforms

- FreeBSD
- OpenBSD
- Linux
- OSX
- 9front

A Taste

```
use std
const main = {
    var lst = ["hello", "world"][:]
    for w in lst
        std.put("{}\n", w)
    ;;
    match std.intparse("123")
    | ‘std.Some 666: std.put("The number of the devil!\n",
    | ‘std.Some n: std.put("integer: {}\n", n)
    | ‘std.None:
    ;;
}
 -- output --
hello
world
integer: 123
```

Code

Fuck it: Live demo.
What could go wrong?

Some Batteries Included

- bio
- date
- fileutil
- regex
- sys
- thread
- cryptohash
- escfmt
- inifile
- std
- testr

libsys

- The platform on which all other libraries are built
- Direct, unabstracted system calls
- One implementation per platform
- Generic syscall stubs in assembly
- System call API exposed via Myrddin
- wrapped in syswrap(-ss)?.myr

libstd

A messy clump of the most common functionality.

- Memory allocation and Slicing
- Error handling
- I/O and File Manipulation
- Portable OS APIs
- Basic Networking
- Common data structures
- Command line parsing
- Formatting
- Varargs
- String Manipulation
- Unicode
- And a bit more misc

libregex

- Full regex implementation
- Supports most useful operators
- Good Unicode support
- Thompson automaton

libdate

- Dates and Times
- Timezones
- Leap years
- Parsing and Formatting

libbio

- Buffered IO
- Line oriented
- Binary Oriented
- No automatic flushing

libescfmt

- Escaping for text
- Virtually no API
- Just casts and `__init__`

lithread

- Basic threading
- Spawn, mutex, condvar
- Currently crap
- Anyone here an expert?

libcryptohash

- md5
- sha1
- sha512
- sha256
- etc

Mbld

- Why not make?
- Ok. So something new.
- In Myrddin, of course.

- Binaries, Libraries, Manpages, Tests, Data
- Can generate source files
- Handles most default rules
- Can select files based on build tags
- Project-aware
- Recursive

bld.proj *orbld.sub*

```
bin foo =
    x.myr
    y.myr
    lib bar
;;

lib bar =
    a.myr
    b.myr
    c.myr
;;
```

The Future

- Self Hosting
- Better Compiler
- Resource Management
- Backends
- Automatic C binding generation
- DWARF support
- Libraries
- Libraries
- Libraries

Self Hosting

- mparse
- qbe
- ?as
- ?lnk

?as, ?lnk

- Consistent behavior
- Static linking
- Farbfeld for objects

Better Compiler

- Currently shit code
- Inlining
- Copy Propagation
- Redundant bound check elimination
- QBE?

Resource Management

- Ownership and moves?
- Linear types?
- This is hard...

Backends

- arm
- mips
- power

Libraries

Libraries Libraries.

- libtls
- libhttp
- lib$windowsystem
- libwidgets
- libfuse
- lib9p
- libmath
- libcompression
- libthread
- libmoreds
- libkvdb
- etc

fin