## **Progress report in Pen programming language**

October 16, 2022

@raviqqe

## Agenda

- Progress report
  - Reflect package
  - Test package improvements
  - $\circ\,$  C calling convention
  - $\circ~$  Compiler optimization
- Next plans

#### **Progress report**

## **Reflect** package

- Reflect package is added for reflection.
- By using it, programs can access type information of values at runtime.
- Currently, it has two functions.
  - Debug
    - Type: \(any) string
    - Pretty-prints a value.
  - Equal
    - Type: \(any, any) boolean | none
    - Compares values for partial equality.

## Test package

• Assertions functions in the Test package can now pretty-print values on failures.

Test code:

```
Check = \() none | error {
   Assert'Equal(42, none)
}
```

Output:

foo.test	t.pen FAIL	Check
summary		lessage: values are not equal (found: 42) (expected: none)
	FAIL	) passed, 1 failed

# **C** calling convention

- The C calling convention is implemented in the Pen compiler.
  - I couldn't wait for C wrapper emission in MLIR...
- It's based on the System V x86-64 ABI.
  - Many other platforms also have adopted it.
- Many hacks to circumvent a fake C calling convention is removed from the compiler.
  - Variant types are passed on stack instead of heap if necessary.

# **Compiler optimization**

Miscellaneous optimization in the Pen compiler.

- Functional algorithm to imperative for CPS transformation
- Unique names for local variables in F--
- Type compilation cache
- 3x speedup for compiling certain modules

#### **Others**

- Everything runs on macOS with M1 chip now.
- Content hash-based rebuilds
  - Implemented in the Turtle build system

### Next plan

• More lower-hanging fruits for compiler optimization

### Summary

- Progress
  - Reflect package
  - Test package improvements
  - $\circ\,$  C calling convention
  - $\circ$  Compiler optimization
- Next plans