Progress report in Pen programming language

February 6th, 2022

@raviqqe

Agenda

- Progress report
  - Hash map in standard package
  - Build system improvements
- Next plans

Progress report

Hash map in standard package (#676)

- Hash map is now available in a `Core` standard package!
- Implemented as a persistent data structure of Hash Array Mapped Trie (HAMT.)
  - Some optimizations are omitted for simplicity for now.
- It also provides maps specialized for string/number-only keys.

```
TestMapGet = \() none | error {
  key = "foo"
  value = "bar"

  m = map'Set(map'New(), key, value)

  Assert'True(map'Get(m, key) == value)
}
```

Build system improvements (#633, #702)

- Turtle build system is integrated into Pen!
- This change brings:
  - Completely comprehensive build output
  - Dependency cycle detection in build system
  - No workaround hack for Ninja's dynamic dependency bug

```
> pen build
resolving dependency of bar.pen
resolving dependency of foo.pen
dependency cycle detected: bar.pen -> foo.pen -> bar.pen
error: build failed
```

Even more...

- Reference count validation (#652)
- `Bit` module in a `Core` package (#673)
- Binary/hexadecimal number literals (#698)
- `#![no_std]` in Rust FFI libraries in `Core` and `Prelude` packages
- Several performance improvements

Next plans

- Multiple system packages in one application
  - `std::process::exit` function in Rust leaves async tasks in a `tokio` runtime...
- Join (select) operation for data parallel computation

Summary

- Hash map is added to the standard package.
- Incoming enhancements on practicality:
  - Multiple system packages
  - Join operation