Progress report in Pen programming language

February 19th, 2022

@raviqqe

Agenda

- Progress report
  - Multiple system packages
- Next plans

Progress report

Multiple system packages (#744)

- Multiple system packages can be used in one application package now!
- Previously, we needed to choose only a system package per application package.
  - e.g. the standard `Os` or `OsSync` packages
  - All other system-dependent codes were built on top of them.
- This enables developers to build *plugin* system packages.
  - e.g. An `Http` package works along with the `Os` package.
- Linking applications is still done by one of system packages in an application package.
  - Multiple link scripts are simply forbidden.

Multiple system packages (examples)

```
import Http'Client
import Os'File

fetch = \(ctx context) none | error {
  s = Client'Get(ctx.Http, "https://pen-lang.org")?
  File'Write(ctx.Os, File'StdOut(), s)?

  none
}
```

Even more...

- Thunk lock handling (#774)
- Algorithm examples (#747)
- Several bug fixes and performance improvements

Next plans

- Proper tail calls in Rust FFI (#757)
  - Without this, some programs in Pen can *leak* stack memory.
- Built-in map type (#634)
  - Some language design work is still required.

Summary

- Progress
  - Multiple system packages can be used in one application package now!
- Future plans
  - More work to stabilize Rust FFI...
  - Built-in map type