Improving IFD

- Discussion pad: https://pad.lix.systems/Xd5Xyt5cToyYxIv-lNdehA
- Further reading: https://jade.fyi/blog/the-postmodern-build-system/

pennae's idea:

```
magic_ifd_primop = \f i -> import f >>= i
# nix:
importAndThen (res: <stuff>) some-ifd.outPath
```

Observation: a magic IFD primop that immediately does something to the result of an import can be "thunked" (thus processed in the background) and offer an "interesting" fix to IFDs. We cannot just make `import` be lazy because of the evaluator internals.

Subsequent observations:

- Explicit monads in Nix may be interesting
- Explicit monads are not necessary if everything "monadic" is related to continuations
- Thunked continuation "entities" / "objects" are sufficient

Why is this interesting?

The capability of forking / parallelizing the work.

Questions:

- jade: how do you generalize this over all the ways that one can consume things via IFD (readFile, etc)?

Revision #4
Created 2025-03-24 03:36:00 UTC by kfearsoff
Updated 2026-02-01 13:14:00 UTC by piegames