ba·bash·ka

A native Clojure interpreter for scripting

Michiel Borkent
@borkdude
2021-04-14
• CLI tools with instant startup! (<10ms)

• **clj-kondo**: a linter for Clojure that sparks joy

• **jet**: convert between JSON, EDN and Transit

• No eval: dynamic classloader not supported!
$ time bb -e '(+ 1 2 3)'
6
0.00s  user 0.00s  system 67%  cpu 0.013 total

- Native Clojure scripting tool, single binary, no JVM
- Can be used to replace “the grey areas” of bash
- Installable via script, brew (macOS, linux), aur (linux), scoop (Windows)
# CLJ scripting

<table>
<thead>
<tr>
<th>Runtime</th>
<th>Impl</th>
<th>Startup*</th>
<th>Interop</th>
<th>Windows</th>
<th>Execution</th>
<th>Threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>clojure</td>
<td>JVM</td>
<td>~1048ms</td>
<td>+</td>
<td>+</td>
<td>Compiled</td>
<td>+</td>
</tr>
<tr>
<td>planck</td>
<td>JSCore</td>
<td>~728ms</td>
<td>+</td>
<td>-</td>
<td>Compiled</td>
<td>-</td>
</tr>
<tr>
<td>joker</td>
<td>Native</td>
<td>~7ms</td>
<td>-</td>
<td>+</td>
<td>Interpreted</td>
<td>-</td>
</tr>
<tr>
<td>babashka</td>
<td>Native</td>
<td>~10ms</td>
<td>+</td>
<td>+</td>
<td>Interpreted</td>
<td>+</td>
</tr>
</tbody>
</table>

*) measured with multitime -n10 <cmd> -e '(+ 1 2 3)' on Ubuntu Bionic with Intel i7-3770K CPU @ 3.50GHz
#!/usr/bin/env bb
(def now (java.time.ZonedDateTime/now))
(def LA-timezone (java.time.ZoneId/of "America/Los_Angeles"))
(def LA-time (.withZoneSameInstant now LA-timezone))
(def pattern (java.time.format.DateTimeFormatter/ofPattern "HH:mm"))
(println (.format LA-time pattern))
Cross platform! 😎

Command Prompt

C:\Users\borkdude\Downloads>type pst.clj
(def now (java.time.ZonedDateTime/now))
(def LA-timezone (java.time.ZoneId/of "America/Los_Angeles"))
(def LA-time (.withZoneSameInstant now LA-timezone))
(def pattern (java.time.format.DateTimeFormatter/ofPattern "HH:mm"))
(println (.format LA-time pattern))

C:\Users\borkdude\Downloads>bb pst.clj
04:50

C:\Users\borkdude\Downloads>
• Built-in libraries

• Examples:
  • python -m http.server replacement

• Pods

• External libraries / projects

• Docs: book.babashka.org
Companies using babashka / sci

Add your company to the list:

Github #254
Los Angeles Clojure Users Group

Wednesday April 8th, 2020

Writing Clojure on the Command Line with Babashka
Small Clojure Interpreter

- Clojure interpreter
- Works on JVM / GraalVM / JS
- Sandboxing
- Works in CLJS advanced compiled apps

(def f (sci/eval-string "#(+ 1 2 %)"))
(f 1)
;;=> 4
SCI eval

- Why does sci work with GraalVM and clojure.core/eval doesn't?

- core eval = compile = generating classes

- A native image cannot load new classes at runtime

- Sci eval does not generate any new classes. Compilation -> interpretation.
Projects using sci

Sci is used in:

- **Babashka.** A Clojure scripting tool that plays well with Bash.
- **Bootleg.** An HTML templating CLI.
- **Bytefield-svg.** NodeJS library to generate byte field diagrams.
- **Chlorine.** Socket-REPL and nREPL package for Atom editor.
- **Cj-kondo.** A Clojure linter that sparks joy.
- **Closh.** Bash-like shell based on Clojure. GraalVM port is work in progress.
- **Dad.** A configuration management tool.
- **Datalevin.** Durable Datalog database.
- **Firn.** Org-mode static site generator.
- **Jet.** CLI to convert between JSON, EDN and Transit.
- **Logseq.** A local-only outline notebook which supports both Markdown and Org mode.
- **Mall.** Plain data Schemas for Clojure/Script.
- **PCP.** Clojure Processor (PHP replacement).
- **PGMig.** Fast Standalone PostgreSQL Migration Runner.
- **Prose.** Alternate syntax for Clojure, similar to what Pollen brings to Racket.
- **SiCMutils.** Computer Algebra System in Clojure, tailored for math and physics investigations.
- **Spire.** Pragmatic provisioning using Clojure.
- **Zprint.** Tool to beautifully format Clojure(script) code and data.
S-expressions to "evaluator" forms

Resolving symbols to fns, locals and classes

Optimizations

Push as much work as possible to analyzer instead of evaluator

The actual function calls to native and interpreted fns

Text to s-expressions
Interop support

(def classes
  {:all [java.io.BufferedReader
    java.io.BufferedWriter
    java.io.ByteArrayInputStream
    java.io.ByteArrayOutputStream
    java.io.File
    java.io.FileInputStream
    java.io.IOException
    java.io.OutputStream
    java.io.Reader
    java.io.StringReader
    java.io.StringWriter])

(Reflector/invokeInstanceMethod obj method (object-array args))
https://gist.github.com/borkdude/9390e0d450ca61d42a037c969d646d9b
Thank you!

ba•bash•ka
A native Clojure interpreter for scripting

https://github.com/borkdude/babashka
https://github.com/borkdude/sci