wasi-backtrace

Andrew Brown

November 2023

Motivation

- **Troubleshooting**: being able to dump a stack trace is quite useful (e.g., diagnosed a multi-threading deadlock)
  - Similar to `debugger` in a browser context
  - Downside: requires a recompile
- **Reflection**: some interest for exception stack traces in Java (?)

wasi-backtrace

```
capture-backtrace: func() -> list<frame>;
record frame {
  codeoffset: u32,
  funcidx: u32,
  funcname: option<string>,
  moduleidx: u32,
  modulename: option<string>,
}
```

Optionally:

```
  locals: list<value>;
  stack: list<value>;
```

Problems

- Can’t be easily implemented in Wasmtime:
  - a WebAssembly backtrace is only accessible via a `Store`
  - Wasmtime’s WIT tooling prevents this—no access to `Caller` ⟶ `Store`
  - Implementing lifting and lowering for `Frame` is complex, fragile

Next steps

- Some options:
  - Discuss further?
  - Poll for phase 1?