Porting Mono to WebAssembly

Experience Report

What’s mono?

- dotnet runtime, FOSS effort started in 2001
- Highly portable
- Used on mobile and by gaming engines
- Interpreter, JIT, AOT, LLVM, plus weird mixes

What are we trying to do?

- Run existing dotnet code in browsers
- Package C# for easy consumption
- 1.0 based on WASM MVP
- High quality language bindings

Current Users

- Community
  - Ooui
  - Xamarin.Forms
- Microsoft
  - Blazor

Interpreter

- Using Emscripten toolchain
    - toolchain stability was an issue early on (fall 2017)
- Most of the work in porting the class libraries
    - Single threaded environment / No IO
- Release runtime -Oz --llvm-lto 1 is 1.7Mb + 166kb JS glue.
- Debug runtime with `-g4 -Os` is 4.8Mb + 875kb JS glue.

AOT Compiler

- Started with our x86 LLVM backend
  - Custom libc/glue due to immaturity
- Now moving to clang with wasm backend
  - lld support major enabler
- Hello world around 10Mb with no C# tree shaking

Debugging

- Chrome only. Implemented as a proxy for v8 wire protocol
- Would LOVE a standardized way to do it
- Source maps NOT the answer, can’t deal with language/ runtime specific features.

Garbage Collection

- Mono requires conservative stack scanning
    - Schedule the GC around frames to avoid that problem
- GC takes about 350kb out of the 1.7mb runtime
- GC spec has major problems and missing features
    - Hard to represent our object/dispatch model
    - No inner pointers hard
    - Host API discussion missing. How to Interpreter / reflection

Exceptions

- We use setjmp/longjmp emcc emulation.
- It works, perf is meh.
- Not high on our list of concerns.

Interop

- Current code requires explicit resource management on both ends.
- Rather wait for GC spec than unleash another GC interop monster

Threading

- C# Libraries not ready for it
- Creative fixing on a case-by-case basis.
- Most user code gets away by using `async` with everything.

Browsers

- Compilation times fluctuates a lot across releases and vendors. It takes 300ms on browser A and 18s on browser B.

- Mobile Browser C all the sudden decided to give up on wasm.

- Corporate firewalls hate us. $10 that half of them simply grep for `This program cannot be run in DOS mode` in binary payloads.