Transpiling WebAssembly into .NET Assemblies

Eric Sink
Microsoft MVP
Twitter: @eric_sink
GitHub: ericsink
Blog: https://ericsink.com/

Overview

- Intro to Wasm and WASI
- Exploration of wasm2cil
- Demos

What is WebAssembly (Wasm)

- Portable assembly language
- Virtual machine
- Safety
- Browser
- Open standard, W3C, Mozilla, et al

What is WASI ?

- WebAssembly System Interface
- Wasm outside the browser
- System calls
- Safety, trust

Work in progress

- Specs for Wasm and WASI are in active development
- Wasm in the browser? Ready today.
- WASI? Not yet mature.

How to get compiled Wasm?

- Almost everywhere (LLVM)
- Rust

How to run Wasm + WASI?

- Host environment
- Interpreter (wasm3)
- JIT-style runtime (wasmtime, wasmer)

Demo

- rust-raytracer
  - forked and simplified a bit (threading)
  - https://github.com/ericsink/rust-raytracer
- Rust
  - cargo wasi
- wasmtime
  - https://wasmtime.dev
- wasmer
  - https://wasmer.io

wasm2cil

- https://github.com/ericsink/wasm2cil
- Transpile Wasm to .NET CIL
  - Inspiration: RyanLamansky/dotnet-webassembly
- Provide an implementation of WASI for .NET

Same demo with wasm2cil

- raytracer.wasm ----> raytracer.dll
- wasi.dll

Wasm vs CIL

- wasm2wat
- ildasm

Instruction Set

Wasm	CIL
i32.shl	shl
i64.add	add
local.get	ldloc
drop	pop

Tricky instructions

- clz
- ctz
- popcnt

Memory model

```
// in wasm:

i32.const 100
i32.const 8675309
i32.store

// in cil:

ldc.i4 100
ldsfld native int [mem]sg_wasm::__mem
add
ldc.i4 8675309
stind.i4
```

Structured control flow blocks

```
block  ;; label = @12
    loop  ;; label = @13
        local.get 0
        i32.load
        local.get 6
        i32.eq
        br_if 1 (;@12;)
        local.get 0
        i32.load offset=8
        local.tee 0
        br_if 0 (;@13;)
        br 2 (;@11;)
    end
end
```

WASI features

- File/Directory I/O
- Paths
- Environment variables
- Clocks
- Random numbers
- Command arguments

WASI file I/O

- Strict permissions from host environment
- POSIX style file descriptors
- stdout, stdin, ...

WASI on .NET

```
(import "wasi_snapshot_preview1" "proc_exit" ...)
(import "wasi_snapshot_preview1" "random_get" ...)
(import "wasi_snapshot_preview1" "clock_time_get" ...)
(import "wasi_snapshot_preview1" "fd_write" ...)
(import "wasi_snapshot_preview1" "fd_prestat_get" ...)
(import "wasi_snapshot_preview1" "fd_prestat_dir_name" ...)
(import "wasi_snapshot_preview1" "environ_sizes_get" ...)
(import "wasi_snapshot_preview1" "environ_get" ...)
```

random_get

```
static Random rnd = new Random();
public static int random_get(
    int addr_buf,
    int buf_len
    )
{
    var ba = new byte[buf_len];
    rnd.NextBytes(ba);
    Marshal.Copy(ba, 0, sg_wasm.__mem + addr_buf, ba.Length);
    return __WASI_ESUCCESS;
}
```

fd_write

```
public static int fd_write(int fd, int addr_iovecs, ...)
{
    ...
    var strm = get_stream_for_fd(fd);
    ...
    for (int i=0; i<iovecs_len; i++) {
        Span<byte> src = (from iovec[i]);
        strm.Write(src);
    }
    ...
    return __WASI_ESUCCESS;
}
```

Performance

- Not enough data to draw precise conclusions
- Ballpark? Maybe faster?
- Unfair comparison? (memory checks)
- .NET Core very mature

Future of Wasm + WASI

- Universal assembly language
- Running untrusted code safely
- Expect large ecosystem of Wasm + WASI packages

Why wasm2cil ?

- Access to Wasm + WASI packages
- Alternative to P/Invoke
- Compile any language for .NET

Demo: cowsay

- wapm

Eric Sink
Twitter: @eric_sink
GitHub: ericsink
Blog: https://ericsink.com/