Utilizzare WebAssembly con .NET, ovunque

Cristian Civera
CTO @ Easydom
cristian@cristiancivera.com | @cristiancivera

.NET Conference
Italia 2023

WebAssembly: perché

- JavaScript non ci basta
- Maggiori prestazioni
- «Offuscare» il codice
- Riutilizzo di librerie
- Linguaggi diversi
- Esempi
  - Google, Unity, Figma, Amazon

WebAssembly: basi

- Esecuzione in sandbox
  - Tutti i moderni browser
- Stack machine
- Formato binario portatile
  - .wasm
  - Versione testuale: .wat
- Componenti
  - Module: estensione .wasm
  - Memory: allocazione di memoria
  - Table: array di referenze
  - Instance: esecuzioni dei moduli

WebAssembly

- Type system
  - I32, i64, f32, f64
  - funcref - referenza a funzione WebAssembly
  - externref - referenza ad un oggetto esterno
- No runtime, garbage, file system, thread, reflection
  - Interazione con l'hosting: JavaScript o hosting esterni
- https://webassembly.org/roadmap/

.NET e WASM

- Runtime di Mono
  - .NET 8.0: SIMD e exception
- Supporto browser
  - dotnet.js e WebCIL
  - Attributi JSImport e JSExport
- Strumenti
  - dotnet workload install wasm-tools
  - dotnet workload install wasm-experimental
- Limitazioni
  - Thread, Networking, I/O

Demo

.NET nel browser

.NET Conference
Italia 2023

WASI

- WebAssembly System Interface
  - https://bytecodealliance.org
- API POSIX
  - Thread, socket, filesystem, random, crypto
- SDK
  - Compilazione
- Runtime
  - Wastime
  - Capability-based security
  - Hosting per Rust, Go, .NET, Python, C/C++, Bash

.NET e WASI

- Compilazione e trimming su WASM
  - Single file
- Allineato al runtime Wasmtime
  - NuGet per hosting
- Import e export funzioni
  - Al momento necessario codice in C
- Supporto non completo
  - WASI non ancora completo

Demo

.NET e WASI

.NET Conference
Italia 2023

Pro e contro

- Sandbox
  - Secure by default
- Platform independent
- Constraint CPU e memory
- Single file deployment
- Fast startup
  - Near native speed
- Multi linaguato
  - Interoperabilità

- Specifiche non complete
- ABI e component model da definire
- Limitazioni nelle API

.NET e Docker

- Supporto beta a WASM
  - https://docs.docker.com/desktop/wasm/
- Multi runtime
  - Wasmtime, Wasmedge, Wasmer, Lunatic, Slight, Spin, Wws
- Stessi constraint e configurazioni
- Rapida creazione OCL
  - Immagine universale

```
FROM scratch
COPY --from=build /build/hello_world.wasm /hello_world.wasm
ENTRYPOINT [ "/hello_world.wasm" ]
```

Demo

.NET e Docker

.NET Conference
Italia 2023

Link utili

- https://github.com/Ricciolo/DotNetWasm
- https://wasmtime.dev
- https://github.com/SteveSandersonMS/dotnet-wasi-sdk
- https://docs.docker.com/desktop/wasm/

@cristiancivera
cristian@cristiancivera.com

aspitalia.com

UGIdotNET

Slide e materiale su
https://www.dotnetconference.it/

.NET Conference
Italia 2023

.NET