

WebAssembly (Wasm): Binární kód pro web

WebAssembly je nízkoúrovňový binární formát pro spustitelný kód v prohlížeči. Není náhradou za JavaScript, ale jeho partnerem. Umožňuje webovým aplikacím provádět výpočetně náročné úlohy (stříh videa, 3D hry, šifrování), které byly dříve pro JavaScript příliš pomalé.

1. Proč potřebujeme WebAssembly?

JavaScript byl původně navržen pro jednoduché interakce na stránce. S rozvojem komplexních aplikací (Google Earth, Figma, Photoshop v prohlížeči) narazil na své limity:

- **Parsování:** JS kód musí prohlížeč nejprve stáhnout jako text a zparsovat, což trvá dlouho.
- **Typování:** Protože je JS dynamicky typovaný, engine prohlížeče musí za běhu neustále kontrolovat typy proměnných.
- **Garbage Collection:** Automatické čištění paměti v JS může způsobit krátké záseky v aplikaci.

2. Klíčové vlastnosti Wasm

- **Rychlost:** Wasm je binární, takže se stahuje rychleji a prohlížeč ho může okamžitě kompilovat do strojového kódu.
- **Bezpečnost:** Běží v bezpečném izolovaném prostředí (sandboxu) prohlížeče, stejně jako JavaScript. Nemá přímý přístup k hardwaru bez svolení uživatele.
- **Portabilita:** Je navržen jako standard (W3C), takže funguje ve všech moderních prohlížečích a na všech operačních systémech.
- **Podpora jazyků:** Umožňuje vzít existující kód v C++ nebo Rustu a s minimálními změnami ho „překlopit“ na web.

3. Jak WebAssembly funguje?

Vývojář nepíše kód přímo ve WebAssembly (to by bylo jako psát v Assembleru). Proces vypadá následovně:

1. Vývojář napíše kód v ****vysokoúrovňovém jazyce**** (např. `[[it:sw:rust_programming|Rust]]`).
2. Pomocí kompilátoru (např. Emscripten nebo wasm-pack) se kód přeloží do souboru ****wasm****.
3. JavaScript v prohlížeči tento soubor načte, zkompiluje a spustí jeho funkce.

4. Praktické využití

Kde už dnes WebAssembly potkáváte, aniž byste o tom věděli?

- **Figma:** Populární nástroj pro designéry běží díky Wasm v prohlížeči stejně plynule jako nativní aplikace.
- **Google Meet:** Rozmazání pozadí u videohovorů v reálném čase je výpočetně náročné a zpracovává ho právě Wasm.
- **Herní engine:** Unity a Unreal Engine využívají Wasm pro export her přímo do webového prohlížeče.
- **Blockchain:** Mnoho moderních sítí (např. Polkadot nebo Near) používá Wasm jako formát pro své [chytré smlouvy](#).

—

5. Budoucnost: Wasm mimo prohlížeč (WASI)

WebAssembly se začíná šířit i mimo webové prohlížeče díky standardu **WASI (WebAssembly System Interface)**. To umožňuje spouštět Wasm moduly na serverech, v cloudu nebo na IoT zařízeních jako bezpečnější a lehčí alternativu k Docker kontejnerům.

Související články:

- [Rust: Nejlepší jazyk pro kompilaci do Wasm](#)
- [JavaScript: Spolupráce s WebAssembly](#)
- [Smart Contracts: Wasm jako engine pro blockchain](#)

Tagy: it wasm webassembly frontend rust performance web-development

From:
<https://www.serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:
<https://www.serviceit.cz/doku.php?id=it:sw:webassembly>

Last update: **2026/01/02 20:23**

