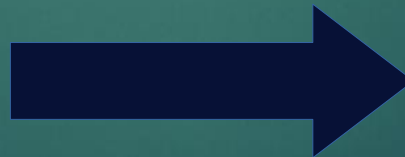


# Go i WASM budućnost weba

► Antonio Nešić

# Historia ipsa repetit

- Čovjek ima potrebu da svaki put iznova “pronalazi kotač”
- Svjedoci smo modernog tehnološkog napretka, koji zapravo najčešće radi inovacije postojećih tehnologija



# Moderna računala i tehnologija

- Višejezgreni procesori
- Puno radne memorije
- Puno različitih operativnih sustava
- Diskrepancija – crossplatform razvoj
- Puno različitih paradigmi i idioma
- Inovacija na starim tehnologijama

# Web – tehnologija koja je promijenila svijet

- Razvio se iz ARPANet-a
- Razvojem WWW – a dobili smo mogućnost bolje razmjene podataka
- Trenutno je najčešći korišten medij u svijetu
- Sve je krenulo s Netscape-om
- Web je postao tehnologija koja je napravila crossplatform development stvarom prošlosti

# Web tehnologije današnjice

- HTML5
- CSS3
- JavaScript + tona frameworka
- Backend sustavi od PHP-a, Jave, JavaScript-a do Go-a
- Standardizirani web – W3C konzorcij

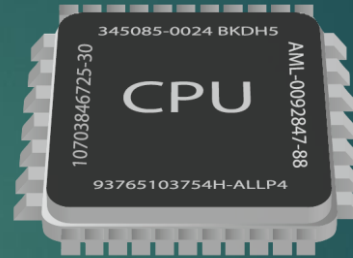
# Java vs JavaScript

- Java – višenamjenski OOP programski jezik
- JavaScript – skriptni jezik rađen za web (lingua franca web-a)
- Java VM -> Appleti (fail)
- Advent JavaScript-a

Performanse

# Kako riješiti problem?

- Vratiti se na osnove
- Assembler
  - Prednosti
    - Mali broj instrukcija
    - Vrlo brz
    - Portabilan
  - Nedostaci
    - Puno instrukcija za osnovne operacije
    - Sintaksa je puno bliža stroju nego čovjeku
- `asm.js`



# WASM – WebAssembly 1.0

▶ <http://www.webassembly.org>



WebAssembly je web standard koji definira binarni format i popratni assembly-like tekst format za izvršni kod na web stranicama.

▶ Drugim riječima:  
Stack bazirana VM za web

Zašto?



# Prednosti WASM-a

- Superbrz
- Napravljen za postojeće low-level jezike C/C++, Rust
- Koristi LLVM toolchain
- Nativni format za browsere
- 4x brži od asm.js-a
- napravljen da kompajlira postojeće jezike u binarni format pogodan za web i daje mogućnost manipulacije kroz JS.

# Za što je pogodan?

- Videoigre
- Decentralizirani serveri
- Visoko performantne aplikacije
- BigData, ML i statistika
- Dovođenje desktop aplikacija na web

# Kako se Go uklapa u ovu priču?

- Go je brz jezik - [link](#)
- Go je moderan jezik sa konkurencijom ugrađenom u sam sustav
- Go se kompajlira u single binary file
- Go je crossplatform jezik
- Kvalitetni libraryji i kod



DEMO

# Kako ovo napraviti kod kuće

- ▶ <https://www.github.com/mutantkeyboard/go>
- ▶ Nakon kloniranja, napraviti checkout **wasm-wip** i otići u src folder
- ▶ **\$cd go/src**
- ▶ **GOOS=js GOARCH=wasm./bootstrap.bash**
- ▶ Kada napravi ovo, dobit ćete bootstrap u direktoriju iznad matičnoga. Primjer **~/Projects/go** je src, a **~/Projects/go-js-wasm** će biti bootstrap folder
- ▶ **\$GOROOT\_BOOTSTRAP=~/Projects/go-js-wasm**
- ▶ **./make.bash**

# Slijedeći koraci

- Kreirajte Go file
- **`./bin/go build -o factorial.wasm factorial.go`**
- **`./misc/wasm/go_js_wasm_exec factorial.wasm`**
- Dobit ćete wasm file, a u misc folderu se nalaze JS i HTML fajlovi koji su boilerplate od kompajlera za pokretanje wasm-a unutar web-a
- <https://github.com/golang/go/issues/18892>
- Gophers slack #wasm



Pitanja?