

# Web-basierte Systeme – Übung

## X2: WebAssembly outside the browser

---

Wintersemester 2023

Arne Vogel



Lehrstuhl für Verteilte Systeme  
und Betriebssysteme



Friedrich-Alexander-Universität  
Technische Fakultät

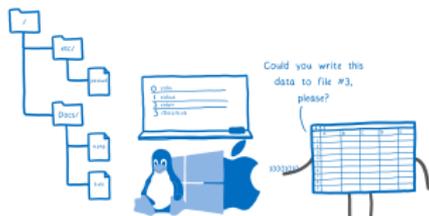
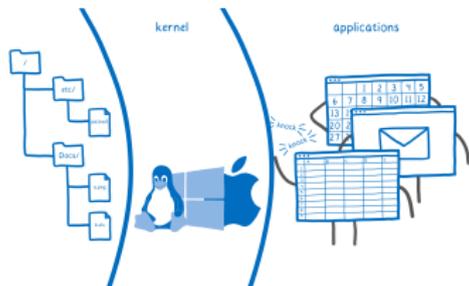
# Security & Portability

- Implemented with security in mind!
- Use WebAssembly outside the browser?

crashes & data leaks

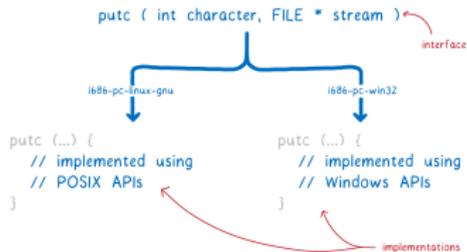


- Protective barriers to system's resources
- Kernel mediates system resource access
- Each system has own system calls



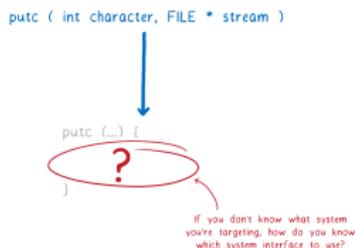
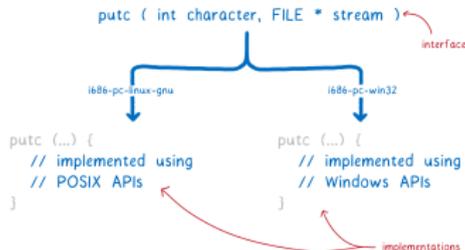
# Abstraction

- Interface to system resources
- At compile time different implementations are used
- Problem solved?



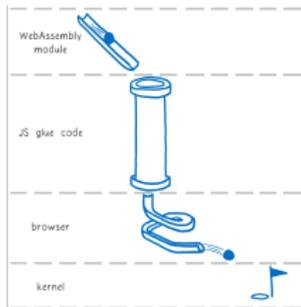
# Abstraction

- Interface to system resources
- At compile time different implementations are used
- WebAssembly bytecode is platform independent
- Where does the interface implementation come from?

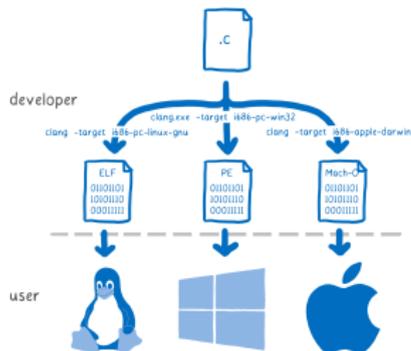


# In the Browser

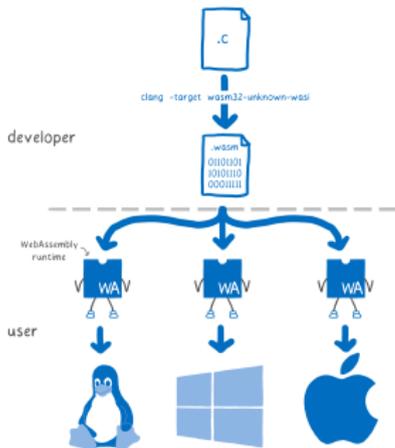
- Emscripten emulates POSIX on the web
- JS glue code uses browser APIs
- Browser talks to kernel



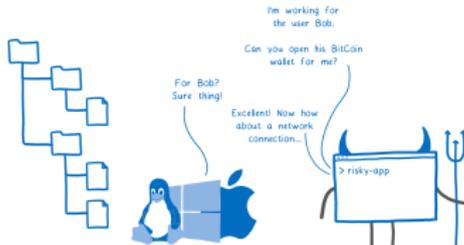
- Compile for every system you target
- More work for every systems
- Do we know all systems?
- Systems that do not exist yet?



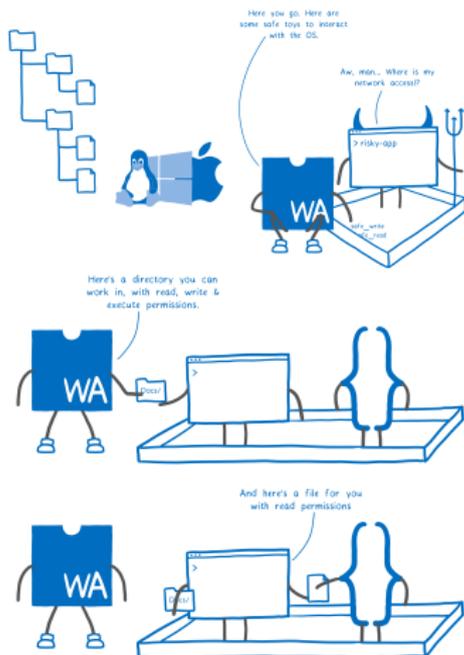
- Only one compilation target
- Compile against the WebAssembly System Interface (WASI)
  - Functions are left to be imported
- Runtime provides system specific implementations
  - Imports are provided at instantiation



- Program has access to all system calls
- No limitation on arguments



- Limit system calls
  - e.g., only read and write files
- Limit arguments
  - e.g., only read files in directory `/config/wbs`
  - e.g., only write files in directory `/tmp`
- Runtime itself can limit capabilities to functions



- Two broad categories
- Standalone applications
  - e.g., compile *grep* for universal deployment
- Libraries
  - Execute WebAssembly inside of programs through embedded runtime



Solomon Hykes / @shykes@hachyderm.io

@solomonstre

...

If WASM+WASI existed in 2008, we wouldn't have needed to create Docker. That's how important it is. Webassembly on the server is the future of computing. A standardized system interface was the missing link. Let's hope WASI is up to the task!



Lin Clark @linclark · Mar 27, 2019

WebAssembly running outside the web has a huge future. And that future gets one giant leap closer today with...

🔥 Announcing WASI: A system interface for running WebAssembly outside the web (and inside it too)

[hacks.mozilla.org/2019/03/standa...](https://hacks.mozilla.org/2019/03/standa...)

[Show this thread](#)

9:39 PM · Mar 27, 2019