Web Assembly

Sam Sartor

February 8, 2018

Mines Linux Users Group

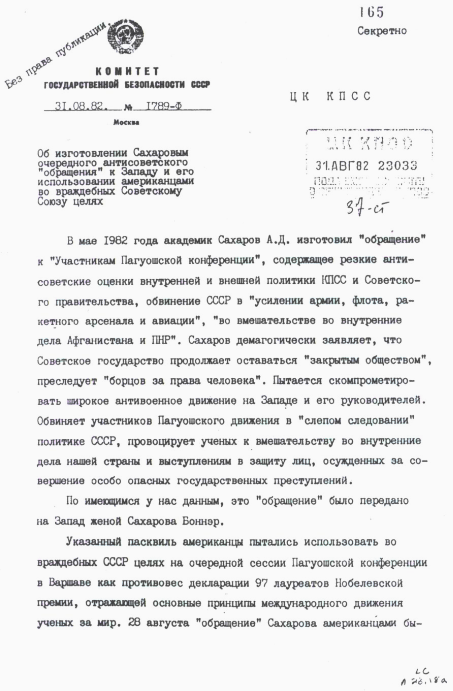
Client Code

What is a website?

A website is really just a fancy document, temporarily downloaded from some remote computer system.

Indeed, Sir Tim Berners-Lee created the world-wide-web as a way of sharing research papers.

These documents contain text, images, and styling information.



165

Секретно

Без права публикации

КОМИТЕТ
ГОСУДАРСТВЕННОЙ БЕЗОПАСНОСТИ СССР

31.08.82. № 1789-Ф

Москва

ЦК КПСС

ЦК КПСС
31АВГ82 23033

Об изготовлении Сахаровым очередного антисоветского "обращения" к Западу и его использовании американцами во враждебных Советскому Союзу целях

В мае 1982 года академик Сахаров А.Д. изготовил "обращение" к "Участникам Пагуошской конференции", содержащее резкие антисоветские оценки внутренней и внешней политики КПСС и Советского правительства, обвинение СССР в "усилении армии, флота, ракетного арсенала и авиации", "во вмешательстве во внутренние дела Афганистана и ПНР". Сахаров демагогически заявляет, что Советское государство продолжает оставаться "закрытым обществом", преследует "борцов за права человека". Пытается скомпрометировать широкое антивоенное движение на Западе и его руководителей. Обвиняет участников Пагуошского движения в "слепом следовании" политике СССР, провоцирует ученых к вмешательству во внутренние дела нашей страны и выступлениям в защиту лиц, осужденных за совершение особо опасных государственных преступлений.

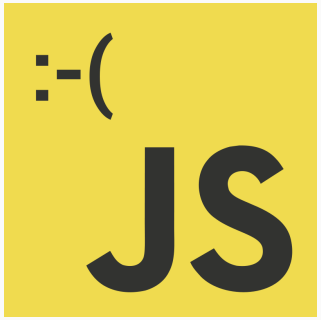
По имеющимся у нас данным, это "обращение" было передано на Запад женой Сахарова Боннэр.

Указанный пасквиль американцы пытались использовать во враждебных СССР целях на очередной сессии Пагуошской конференции в Варшаве как противовес декларации 97 лауреатов Нобелевской премии, отражающей основные принципы международного движения ученых за мир. 28 августа "обращение" Сахарова американцами бы-

Interactive Websites

Is it possible to make websites interactive?

We would have to include some instructions along with the document. We need some kind of scripting language for webpages.



JavaScript

On a dark and stormy night . . .

The year is 1995. Netscape Communications Corporation is dying. In a frantic attempt to one-up Microsoft, the company decides to embed a scripting language into the Netscape browser.

They give **Brendan Eich** 10 days to make a prototype.

Eich dreams of Scheme. He wants his new language to be elegant, fast, and pure. **But it is too late.** The lawyers at Netscape have made a deal with Sun Microsystems.

It will be known as **JavaScript**.

The Result


Activate Your Web Pages
JavaScript
The Definitive Guide
O'REILLY
David Flanagan
Unearthing the Excellence in JavaScript
JavaScript:
The Good Parts
O'REILLY
YAHOO! PRESS
Douglas Crockford

This page uses 800 thousand lines of JavaScript

Multiproject Workspace

label:cart | DASHBOARD | HELP & UPDATES NEW | LISADEMO

4

+ Add Story

Provide Feedback On The Beta

My Work 7

MY SAMPLE PROJECT 10
- Current/Backlog
- Icebox
- Epics
- Other Panels

RELEASES & MILESTONES 10
- Current/Backlog
- Icebox
- Epics
- Other Panels

SHOP MOBILE 10
- Current/Backlog
- Icebox
- Epics
- Other Panels

SHOP WEB 10
- Current/Backlog
- Icebox
- Epics
- Other Panels

SHOP WEB
Current/Backlog

56 18 Aug - Current — Pts: 0 of 20

- Admin should be able to import multiple new products from CSV file (LISA C, CE) — Accept / Reject
  admin
- Admin should be able to create new product (LD, CE) — Deliver
  admin
- Admin should be able to login (LISA C) — Finish
  admin
- Shopper should see list of products, with primary photo as thumbnail — Finish
  shopping
- Shopper should be able to add product to shopping cart (LISA C) — Finish
  cart, shopping
- Shopper should be able to click on a product, and see all product details, including photos — Finish
  shopping
- Admin should be able to upload multiple product photos and mark one as the primary (LISA C) — Finish
  admin, blocked
- Admin should be able to upload product photo (CE, LISA C) — Finish
  admin
- **Basic Admin Features** — Finish
- another bug created by groovy (CE) — Accept / Reject
- Shopper should be able to compare products (CE, LISA C) — Accept / Reject
  shopping

57 25 Aug — Pts: 10

- Inventory status endpoint — Start
  api, inventory
- Product browsing should be paginated, with 10 products per page (LISA C) — Start
  needs discussion, shopping
- Shopper should be able to view contents of shopping cart — Start
  cart, shopping
- Shopper should be able to remove product from shopping cart — Start
  cart, shopping
- Cart manipulation should be AJAXy — Start
  cart, shopping
- Some product photos not scaled properly when browsing products — Start
  shopping
- **Initial demo to investors** — Finish

58 1 Sep — Pts: 4

- Shopper should be able to recommend a product to a friend — Start
  shopping
- Shopper should be able to search for product — Start
  search, shopping
- *Shopper should be able to review products*
  needs design, shopping
- *New!*
  design

MY SAMPLE PROJECT
Current/Backlog

10 18 Aug - Current — Pts: 0 of 10

- Shopper should see a sample shopping cart (LISA C, CE, LD) — Finish
  cart
- Some product photos not scaled properly when browsing products (LISA C) — Finish
  shopping
- Shopper should be able to recommend a product to a friend — Start
  shopping
- configure solr for full text searching — Start
  search
- Shopper should be able to search for product — Start
  search, shopping
- **Initial demo to investors** — Finish
- Shopper should be able to enter credit card information and shipping address — Start
  checkout, shopping
- Integrate with payment gateway — Start
  checkout, shopping
- When shopper submits order, authorize total product amount from payment gateway — Start
  checkout, needs discussion, shopping

11 25 Aug — Pts: 10

- If system fails to authorize payment amount, display error message to shopper — Start
  checkout, shopping
- If authorization is successful, show order number and confirmation message to shopper — Start
  checkout, shopping
- Send notification email of order placement to admin — Start
  admin, checkout, shopping
- Shopper should be able to check status of order by entering name and order number — Start
  orders
- Shopper should be able to ask question about order — Start
  orders
- Admin can review all order questions and send responses to shoppers — Start
  admin, orders
- Set up Engine Yard production environment — Start
  deployment
- **Beta launch** — Finish
- Shopper should be able to remove product from shopping cart (LISA C) — Start
  shopping
- Shopper should be able to sign up for an account with email address — Start
  signup / signin

12 1 Sep — Pts: 9

- Shopper should be able to reset forgotten password — Start
  signup / signin

label:cart

show 1 done story

- **My Sample Project** — Finish
  Shopper should see a sample shopping cart (LISA C, CE, LD)
  cart
- **Shop Mobile** — Deliver
  Shopper should be able to select item and quantity (LISA C)
  cart, requires web story
- **Shop Mobile** — Finish
  Shopper should be able to view items and quantity
  cart, shopping
- **Shop Mobile** — Start
  Shopper should be able to delete previously selected items
  cart, shopping
- **Shop Web** — Finish
  Shopper should be able to add product to shopping cart (LISA C)
  cart, shopping
- **Shop Web** — Start
  Shopper should be able to view contents of shopping cart
  cart, shopping
- **Shop Web** — Start
  Shopper should be able to remove product from shopping cart
  cart, shopping
- **Shop Web** — Start
  Cart manipulation should be AJAXy
  cart, shopping

Emscripten

We need C

- People start writing games in JavaScript
- Why can't we use the Unity game engine?
- It's written in C
- Browsers don't run C
- Browsers only run JavaScript
- What if we compiled C to JavaScript?

Emscripten

Introducing. . .

It compiles things to JavaScript!

asm.js Example

C

```c
int factorial(int n) {
  if (n == 0)
    return 1;
  else
    return n *
   ↪  factorial(n-1);
}
```

asm.js

```
function __Z9factoriali($0) {
 $0 = $0|0;
 var $1 = 0, $10 = 0, $2 = 0, $3 = 0, $4 = 0, $5 = 0, $6
 ↪  = 0, $7 = 0, $8 = 0, $9 = 0, label = 0, sp = 0;
 sp = STACKTOP;
 STACKTOP = STACKTOP + 16|0; if ((STACKTOP|0) >=
 ↪  (STACK_MAX|0)) abortStackOverflow(16|0);
 $2 = $0;
 $3 = $2;
 $4 = ($3|0)==(0);
 if ($4) {
  $1 = 1;
 } else {
  $5 = $2;
  $6 = $2;
  $7 = (($6) - 1)|0;
  $8 = (__Z9factoriali($7)|0);
  $9 = Math_imul($5, $8)|0;
  $1 = $9;
 }
 $10 = $1;
 STACKTOP = sp;return ($10|0);
}
```

That was a Bad Idea

JavaScript was designed for...

- ✓ Crazy people
- ● Humans
- × Computers

Why compile low-level → high-level?

Why don't we have *machine code for the web?*

```
80483b4: 55      push
80483b5: 89 e5   mov
80483b7: 83 e4   and
80483ba: 83 ec   sub
80483bd: c7 44   movl
80483c4: 00
80483c5: eb 11   jmp
80483c7: c7 04   movl
80483ce: e8 1d   call
80483d3: 83 44   addl
80483d8: 83 7c   cmpl
80483dd: 7e e8   jle
80483df: b8 00   mov
80483e4: c9      leave
80483e5: c3      ret
80483e6: 90      nop
80483e7: 90      nop
80483e8: 90      nop
80483e9: 90      nop
80483ea: 90      nop
```

Web Asssembly

Which Direction?

emscripten can compile your code to *Web Assembly*!

WASM Example

C

```c
int factorial(int n) {
  if (n == 0)
    return 1;
  else
    return n *
   ↪  factorial(n-1);
}
```

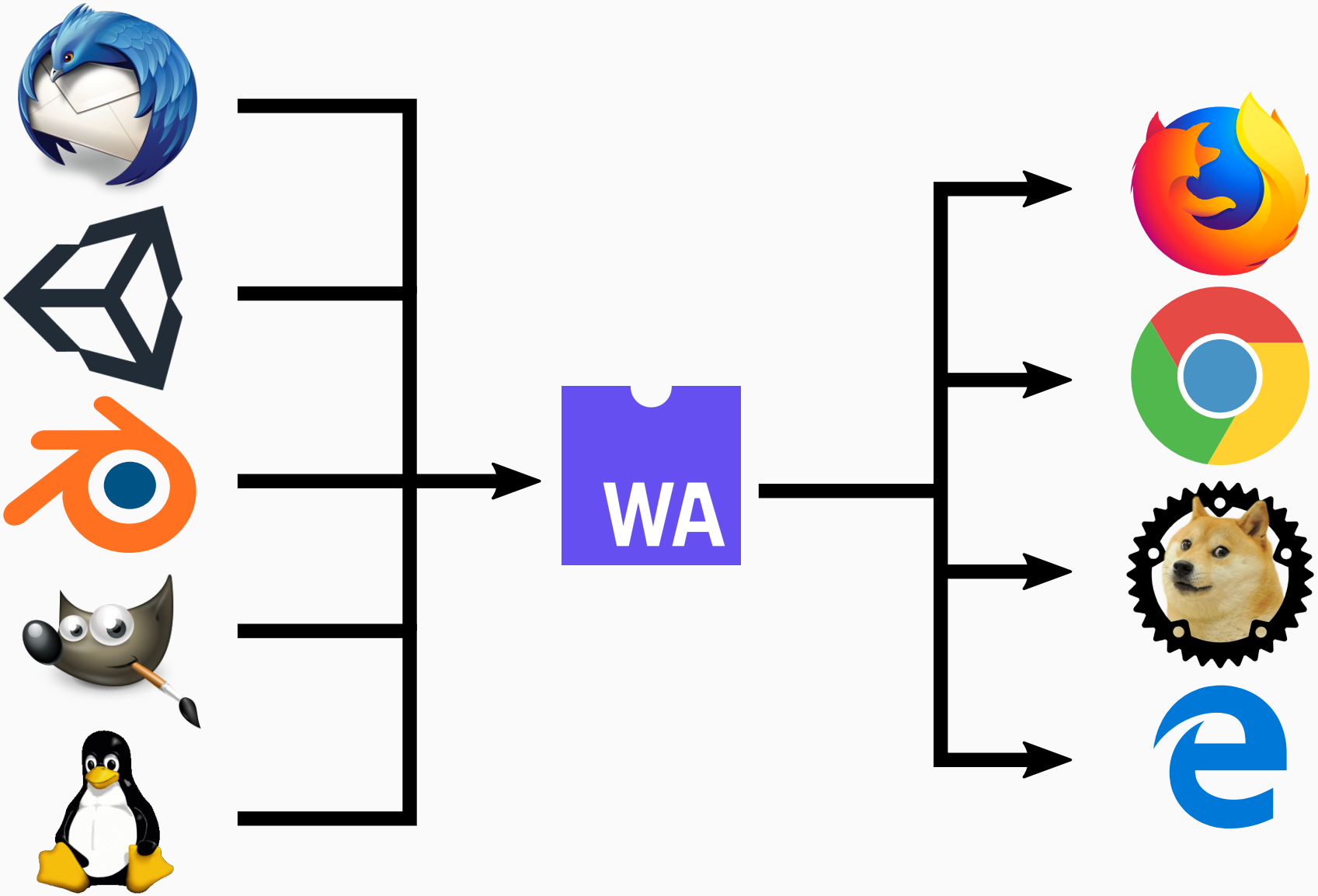
WASM Text

```
get_local 0
i64.eqz
if i64
    i64.const 1
else
    get_local 0
    get_local 0
    i64.const 1
    i64.sub
    call 0
    i64.mul
end
```

WASM Binary

```
20 00
50
04 7E
42 01
05
20 00
20 00
42 01
7D
10 00
7E
0B
```

Software in the Browser



Practical WASM

C/C++

emsdk

We can use the emscripten sdk to compile C/C++ to JavaScript and WASM.

Emscripten can build a test HTML webpage, JavaScript file, or a pure wasm library.

```
$ emcc demo.c -s WASM=1 -o demo.html
$ emcc demo.c -s WASM=1 -o demo.js
$ emcc demo.c -s WASM=1 -s SIDE_MODULE=1 -o demo.wasm
```

Rust

The Rust Language

Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety.

```rust
fn main() {
  let number = 13;
  let fact = match number {
    1 => "is one",
    2 | 3 | 5 | 7 => "is prime",
    13...19 => "is a teen",
    _ => "ain't special",
  };

  println!("{} {}!", number, fact);
}
```

Rust Toolchains

Rust can be easily compiled to JavaScript and WebAssembly!

You can install a WASM toolchain by running

```
$ rustup toolchain add wasm32-*-*
```

The available toolchains are:

`wasm32-unknown-unknown`	compile to pure WASM
`wasm32-unknown-emscripten`	compile to a WASM executable
`asmjs-unknown-emscripten`	compile to JavaScript

Cargo

Rust comes with a build system called **Cargo**.

```
$ cargo run
   Finished helloworld [unoptimized + debuginfo] target(s) in 0.0 secs
    Running `target/debug/helloworld`
Hello, world!
```

Cargo can target WASM!

```
$ cargo build --target wasm32-unknown-emscripten
   Compiling helloworld v0.1.0 (file:///home/sam/Code/helloworld)
    Finished dev [unoptimized + debuginfo] target(s) in 1.26 sec
$ cd target/wasm32-unknown-emscripten/debug/
$ node helloworld.js
Hello, world!
```

stdweb

Interact

The whole point is to make interactive websites. How can we edit HTML through Rust? How do we make Rust web development feasible?

Introducing. . .

parcel

Bundeling JS and Rust

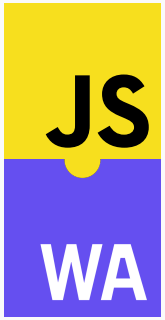
We can use Parcel to bundle JS and WASM into the same project.

In `main.js`:

```
import {add} from './add.rs';
console.log(add(2, 3));
```

In `add.rs`:

```
#[no_mangle]
pub fn add(a: i32, b: i32) -> i32 {
    a + b
}
```



Putting it all Together

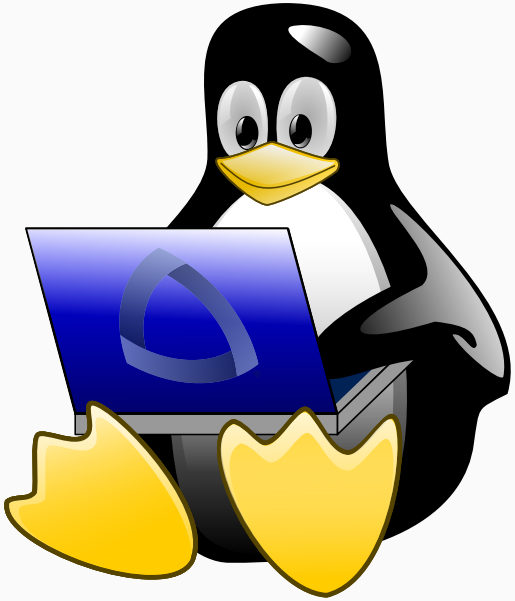
Conclusion

- Faster websites
- Written in any language
- With desktop libraries
- With native graphics
- With modern tools

?

Questions?

Copyright Notice



Colorado School of Mines
Linux Users Group