Wasm on the browser is cool

Wasm beyond the web

- fast, scalable, secure way to run the same code across all machines

Wasm beyond the web

- fast, scalable, secure way to run the **same** code across all machines

Wasm beyond the web

- fast, scalable, secure way to run the **same** code across all machines
- safe and portable

Wasm beyond the web

- fast, scalable, secure way to run the **same** code across all machines
- safe and portable, we want:
  - users and programs can do what they have the right to do

Wasm beyond the web

- fast, scalable, secure way to run the **same** code across all machines
- safe and portable, we want:
  - users and programs can do what they have the right to do
  - do not create problems for other users and programs

Wasm beyond the web

- fast, scalable, secure way to run the **same** code across all machines
- safe and portable, we want:
  - users and programs can do what they have the right to do
  - do not create problems for other users and programs

standard, platform-independent methods

to declare and apply

Wasm beyond the web

- fast, scalable, secure way to run the **same** code across all machines
- safe and portable, we want:
  - users and programs can do what they have the right to do
  - do not create problems for other users and programs
  - standard, platform-independent methods to declare and apply
- Binary compatibility

Wasm beyond the web

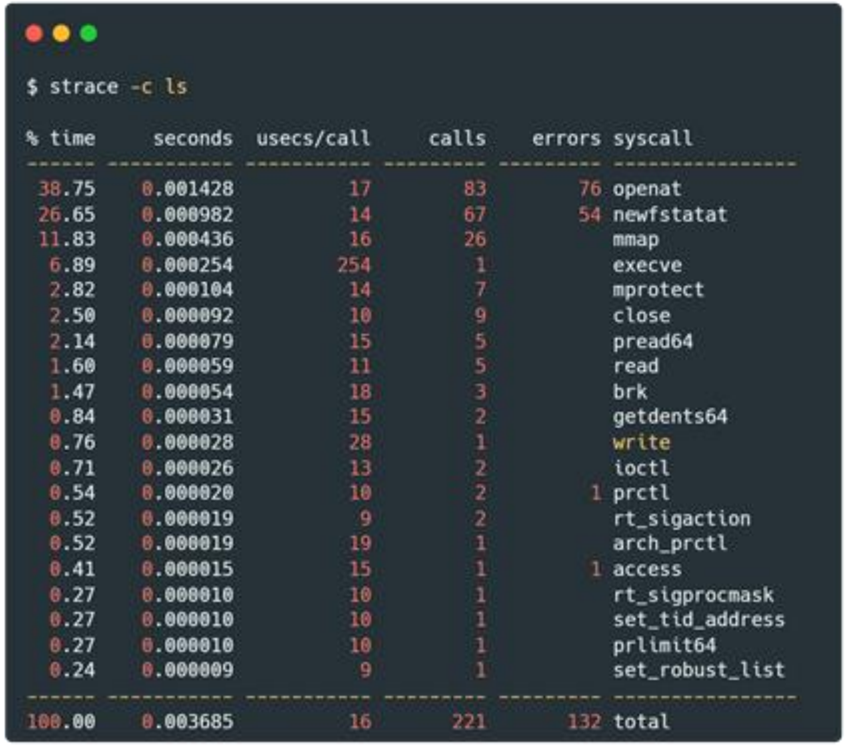
- fast, scalable, secure way to run the **same** code across all machines
- safe and portable, we want:
    - users and programs can do what they have the right to do
    - do not create problems for other users and programs
    - standard, platform-independent methods to declare and apply
- Binary compatibility

Wasm beyond the web

- fast, scalable, secure way to run the **same** code across all machines
- safe and portable, we want:
    - users and programs can do what they have the right to do
    - do not create problems for other users and programs
    - standard, platform-independent methods to declare and apply
- Binary compatibility
- Aside: System Interface
- Aside: Some Nostalgia

System Interface

System Interface



```
$ strace -c ls

% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 38.75    0.001428          17        83        76 openat
 26.65    0.000982          14        67        54 newfstatat
 11.83    0.000436          16        26           mmap
  6.89    0.000254         254         1           execve
  2.82    0.000104          14         7           mprotect
  2.50    0.000092          10         9           close
  2.14    0.000079          15         5           pread64
  1.60    0.000059          11         5           read
  1.47    0.000054          18         3           brk
  0.84    0.000031          15         2           getdents64
  0.76    0.000028          28         1           write
  0.71    0.000026          13         2           ioctl
  0.54    0.000020          10         2         1 prctl
  0.52    0.000019           9         2           rt_sigaction
  0.52    0.000019          19         1           arch_prctl
  0.41    0.000015          15         1         1 access
  0.27    0.000010          10         1           rt_sigprocmask
  0.27    0.000010          10         1           set_tid_address
  0.27    0.000010          10         1           prlimit64
  0.24    0.000009           9         1           set_robust_list
------ ----------- ----------- --------- --------- ----------------
100.00    0.003685          16       221       132 total
```

System Interface

- fast, scalable, secure way to run the **same** code across all machines
- safe and portable, we want:
  - users and programs can do what they have the right to do
  - do not create problems for other users and programs
  - standard, platform-independent methods to declare and apply
- Binary compatibility
- Aside: System Interface
- Aside: Some Nostalgia

Some Nostalgia

- Emscripten was already emulating a system interface, POSIX, on the web
- Implementation of `libc` had to be used with JS glue code which would be called in the browser and would talk to the kernel
- Runtimes re-implement the JS glue code and emulate a browser

Web Assembly Module

Emulating JS Glue and Browser

Kernel