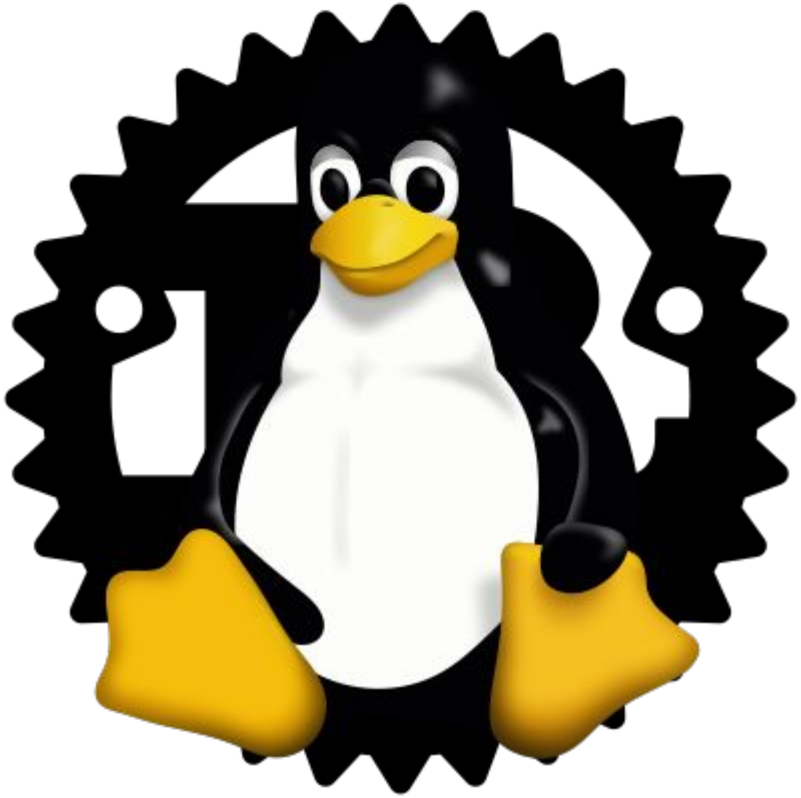


Rust in the Linux ecosystem

Miguel Ojeda

ojeda@kernel.org

Credits & Acknowledgments

Rust

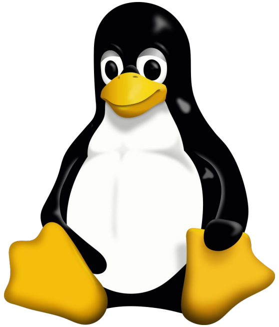
...for being a breath of fresh air

Kernel maintainers

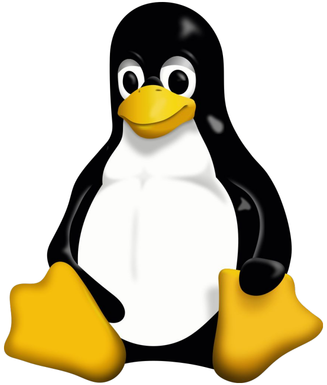
...for being open-minded

Everyone that has helped Rust for Linux

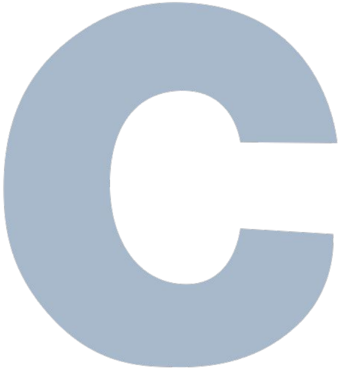
(see credits in the RFC & patch series)



History

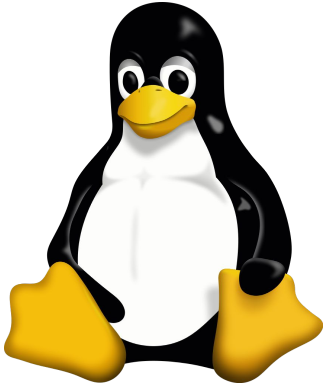


30 years of Linux

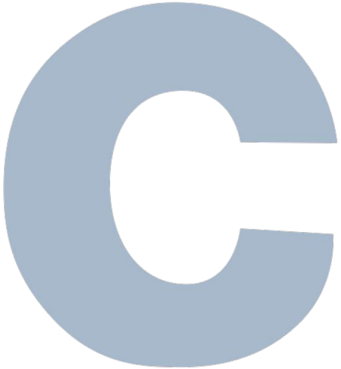


30 years of ISO C

Love story*



❤️*



30 years of Linux

30 years of ISO C

** Terms and Conditions Apply.*

Why is C a good system programming language?

Why is C a good system programming language?

> “Do you see any language except C which is suitable for development of operating systems?”

Why is C a good system programming language?

> “Do you see any language except C which is suitable for development of operating systems?”

> “I like interacting with hardware from a software perspective. And I have yet to see a language that comes even close to C.”
>
> — Linus Torvalds 2012

Why is C a good system programming language?

“You can use C to generate good code for hardware.”

Fast

“When I read C, I know what the assembly language will look like.”

Low-level

“The people that designed C ... designed it at a time when compilers had to be simple.”

Simple

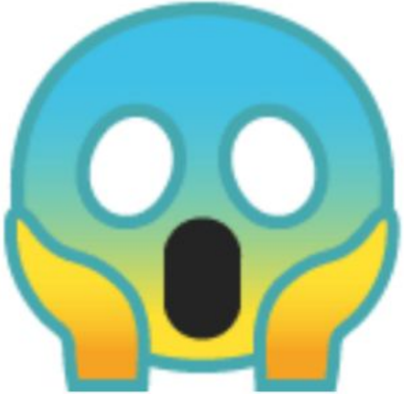
“If you think like a computer, writing C actually makes sense.”

Fits the domain

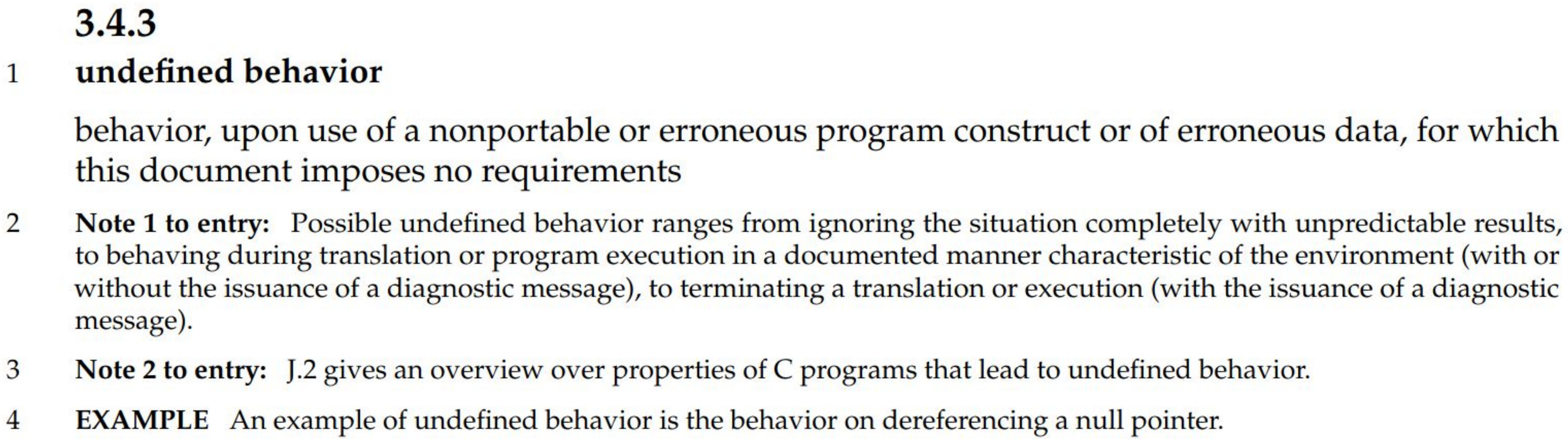
But...

But...

UB



Undefined Behavior



3.4.3

1 **undefined behavior**

behavior, upon use of a nonportable or erroneous program construct or of erroneous data, for which this document imposes no requirements

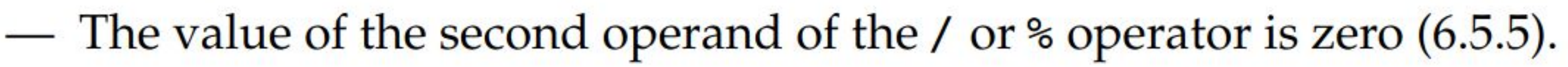
2 **Note 1 to entry:** Possible undefined behavior ranges from ignoring the situation completely with unpredictable results, to behaving during translation or program execution in a documented manner characteristic of the environment (with or without the issuance of a diagnostic message), to terminating a translation or execution (with the issuance of a diagnostic message).

3 **Note 2 to entry:** J.2 gives an overview over properties of C programs that lead to undefined behavior.

4 **EXAMPLE** An example of undefined behavior is the behavior on dereferencing a null pointer.

— N2596 C2x Working Draft

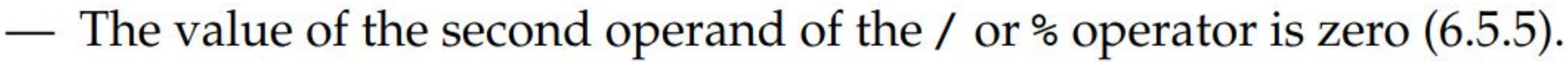
Example of UB



— The value of the second operand of the / or % operator is zero (6.5.5).

```
int f(int a, int b) {
    return a / b;
}
```

Example of UB



— The value of the second operand of the / or % operator is zero (6.5.5).

```
int f(int a, int b) {
    return a / b;
}
```

UB ∀x f(x, 0);

Example of UB

Any other inputs that trigger UB?

```
int f(int a, int b) {
    return a / b;
}
```

Example of UB

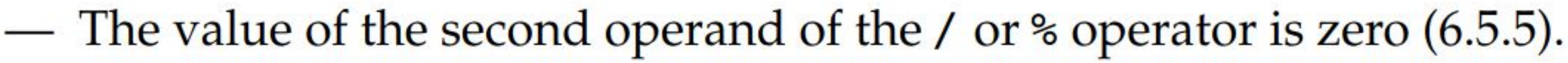
Any other inputs that trigger UB?

```
int f(int a, int b) {
    return a / b;
}
```

```
UB f(INT_MIN, -1);
```

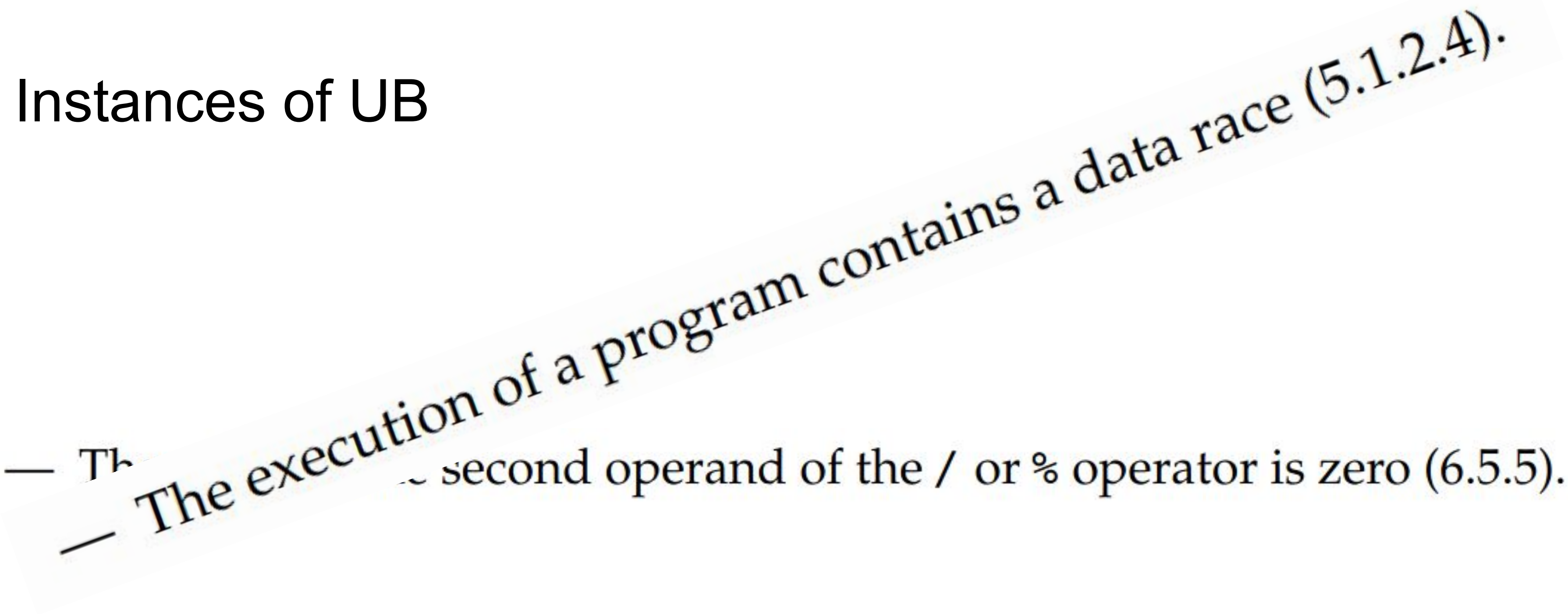
Instances of UB

Instances of UB



— The value of the second operand of the `/` or `%` operator is zero (6.5.5).

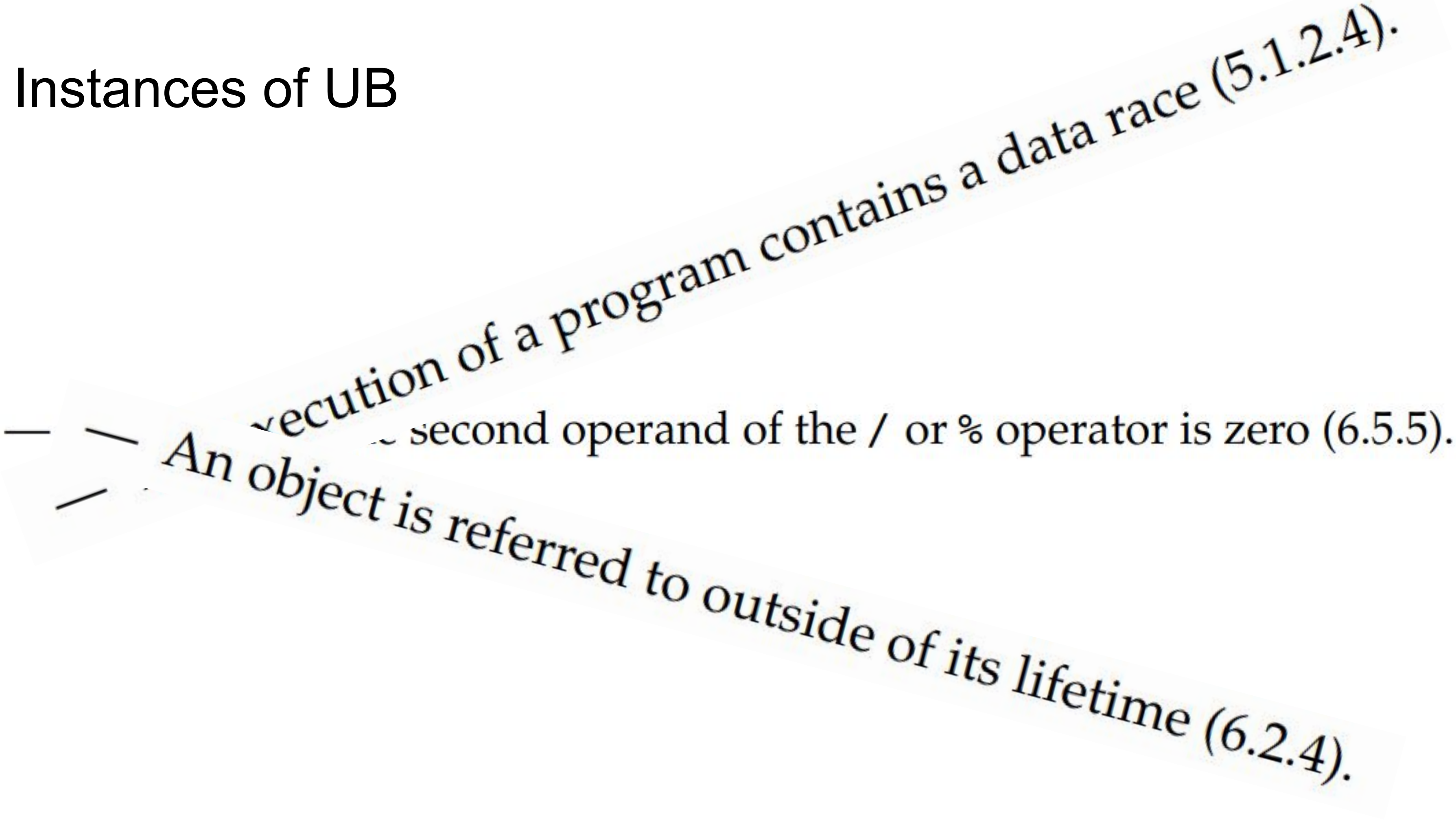
Instances of UB



— Th second operand of the / or % operator is zero (6.5.5).

— The execution of a program contains a data race (5.1.2.4).

Instances of UB

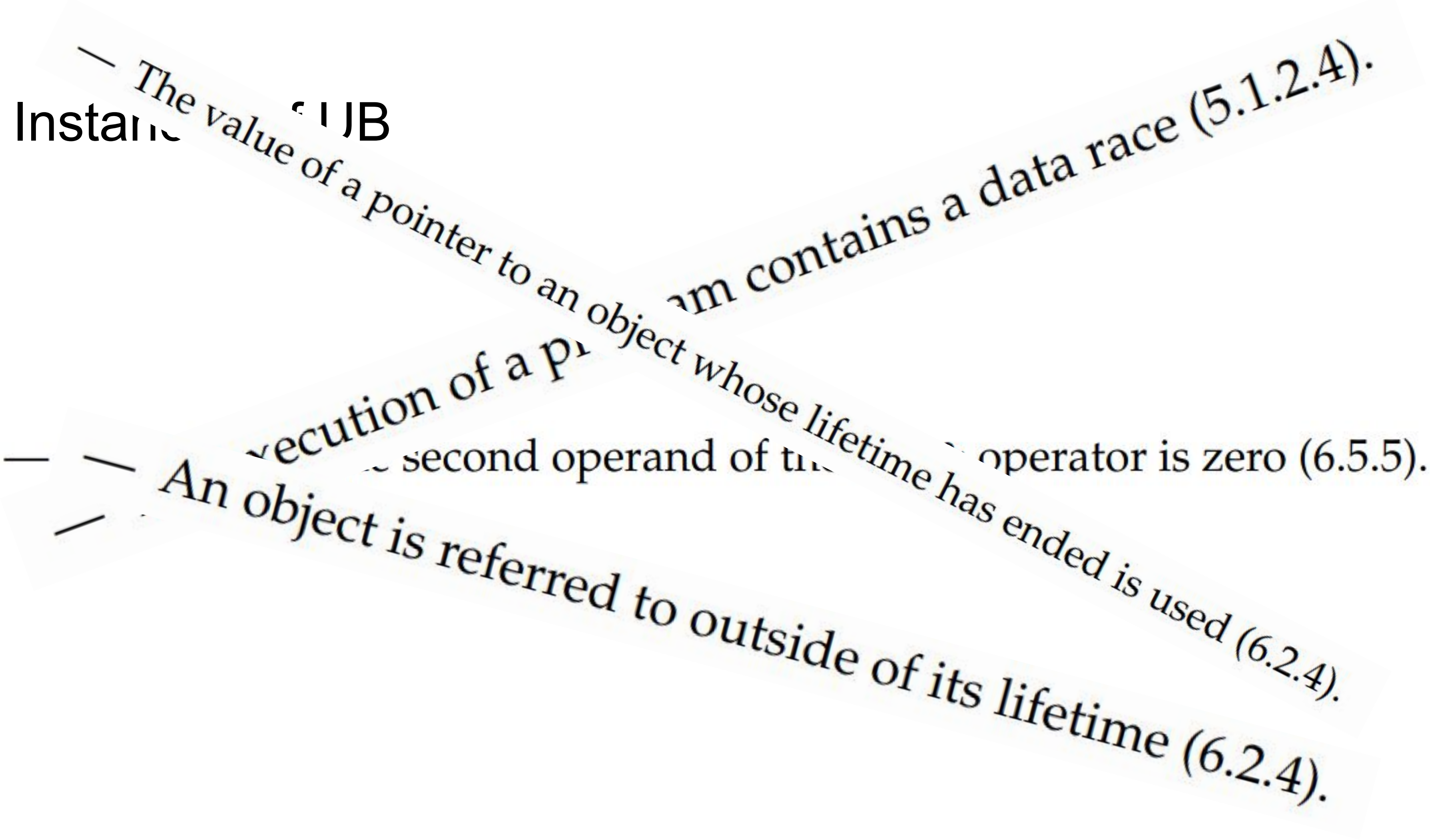


ecution of a program contains a data race (5.1.2.4).

— second operand of the / or % operator is zero (6.5.5).

— An object is referred to outside of its lifetime (6.2.4).

Instan f UB



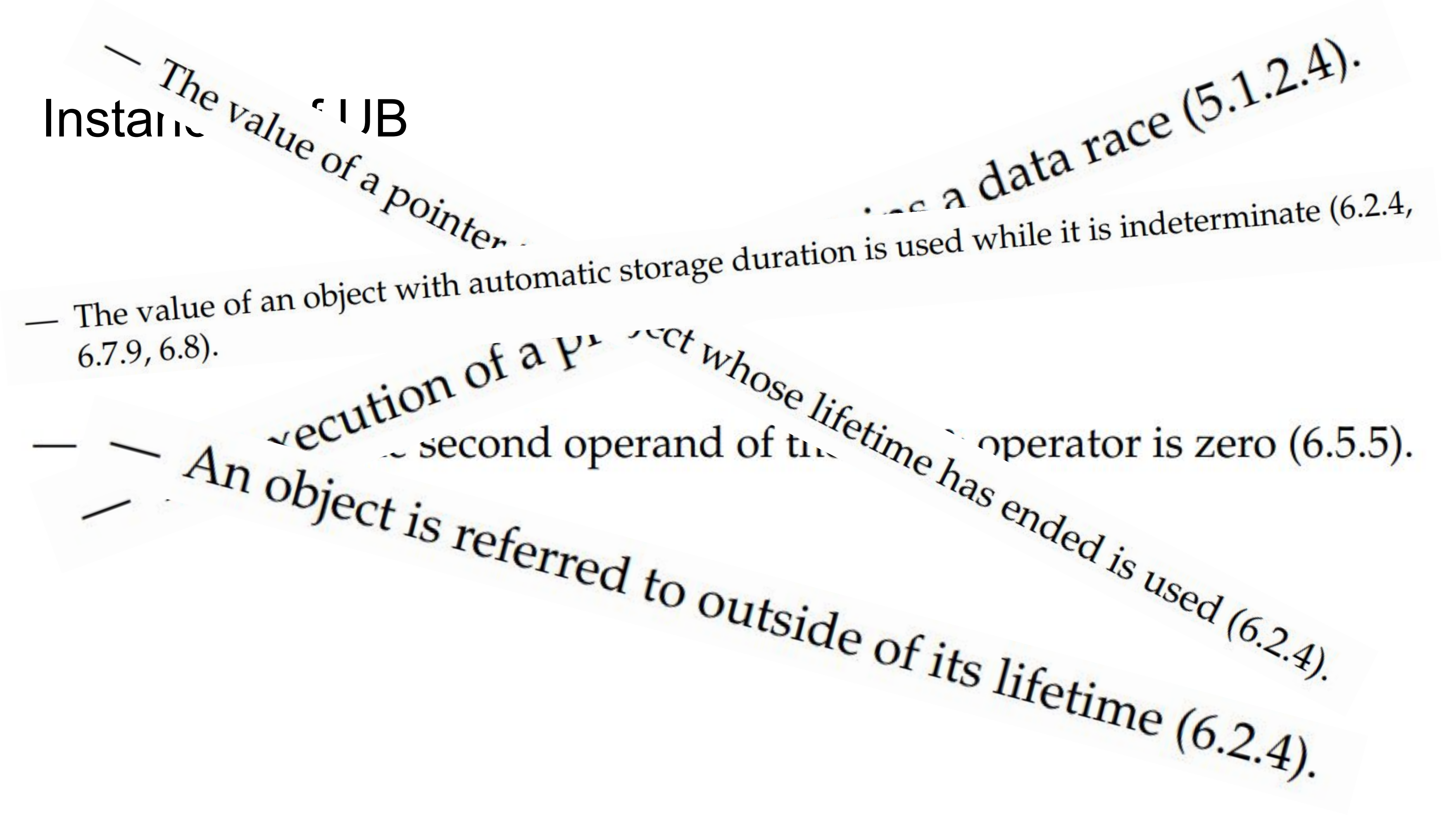
— The value of a pointer to an object whose lifetime has ended is used (6.2.4).

xecution of a pr am contains a data race (5.1.2.4).

— second operand of th operator is zero (6.5.5).

— An object is referred to outside of its lifetime (6.2.4).

Instan UB



— The value of a pointer

a data race (5.1.2.4).

— The value of an object with automatic storage duration is used while it is indeterminate (6.2.4, 6.7.9, 6.8).

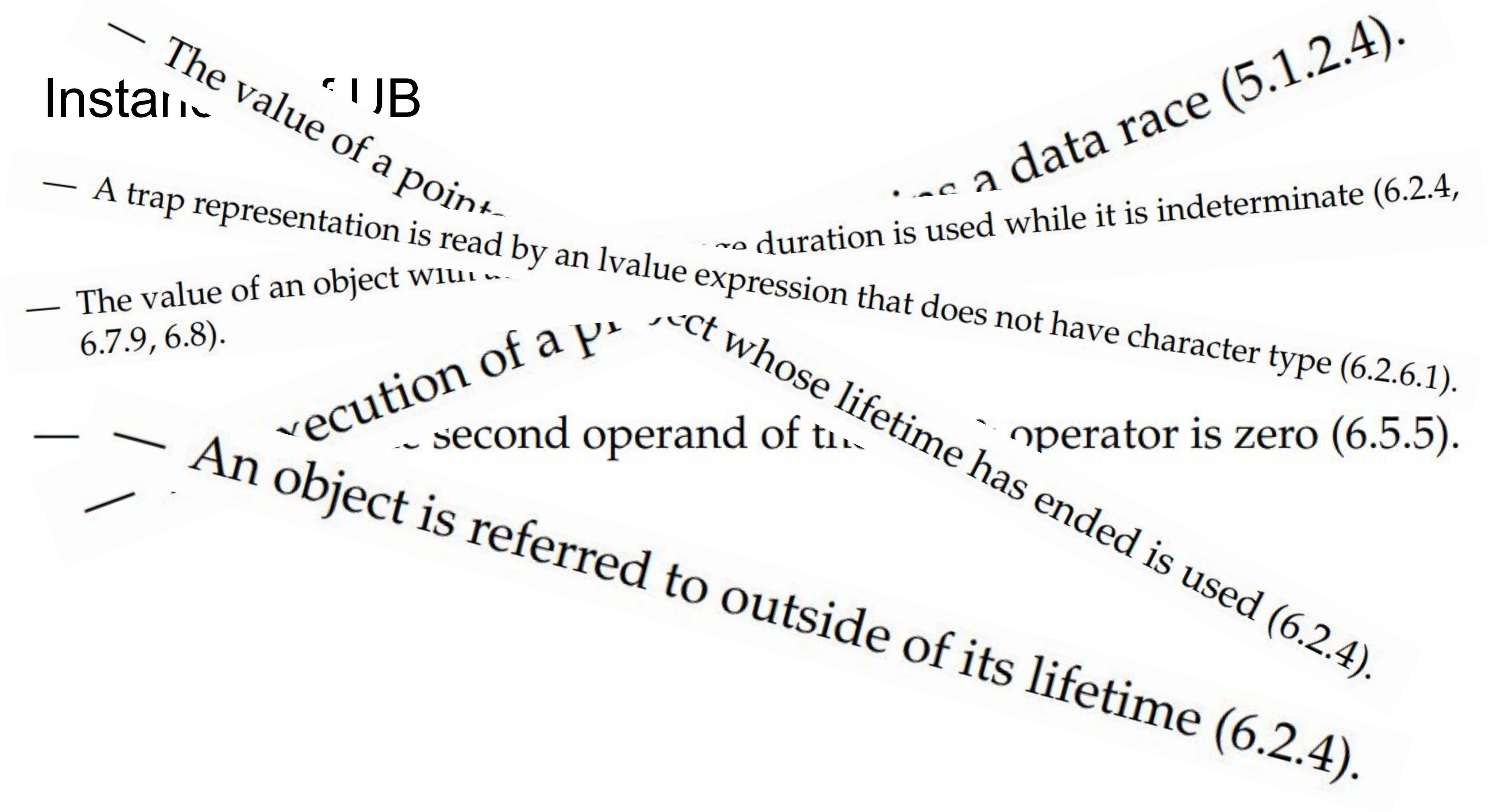
ecution of a p

ect whose lifetime has ended is used (6.2.4).

— — second operand of th operator is zero (6.5.5).

— — An object is referred to outside of its lifetime (6.2.4).

Instan UB



— The value of a poin ect whose lifetime has ended is used (6.2.4).

— ecution of a pr es a data race (5.1.2.4).

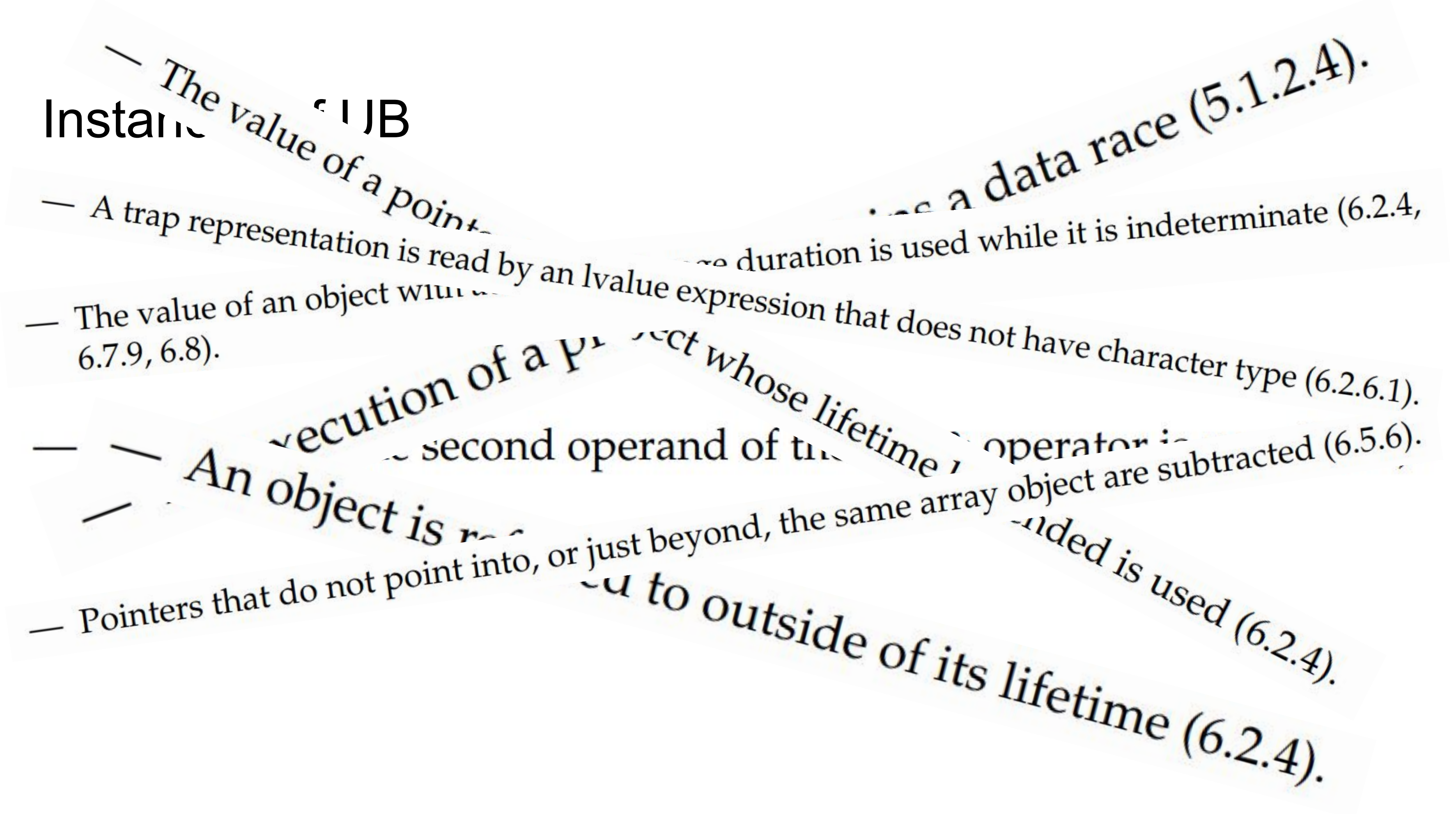
— A trap representation is read by an lvalue expression that does not have character type (6.2.6.1).

— The value of an object wit e duration is used while it is indeterminate (6.2.4, 6.7.9, 6.8).

— second operand of th operator is zero (6.5.5).

— An object is referred to outside of its lifetime (6.2.4).

Instances of UB



— The value of a point ect whose lifetime nded is used (6.2.4).

— A trap representation is read by an lvalue expression that does not have character type (6.2.6.1).

— The value of an object wit e duration is used while it is indeterminate (6.2.4, 6.7.9, 6.8).

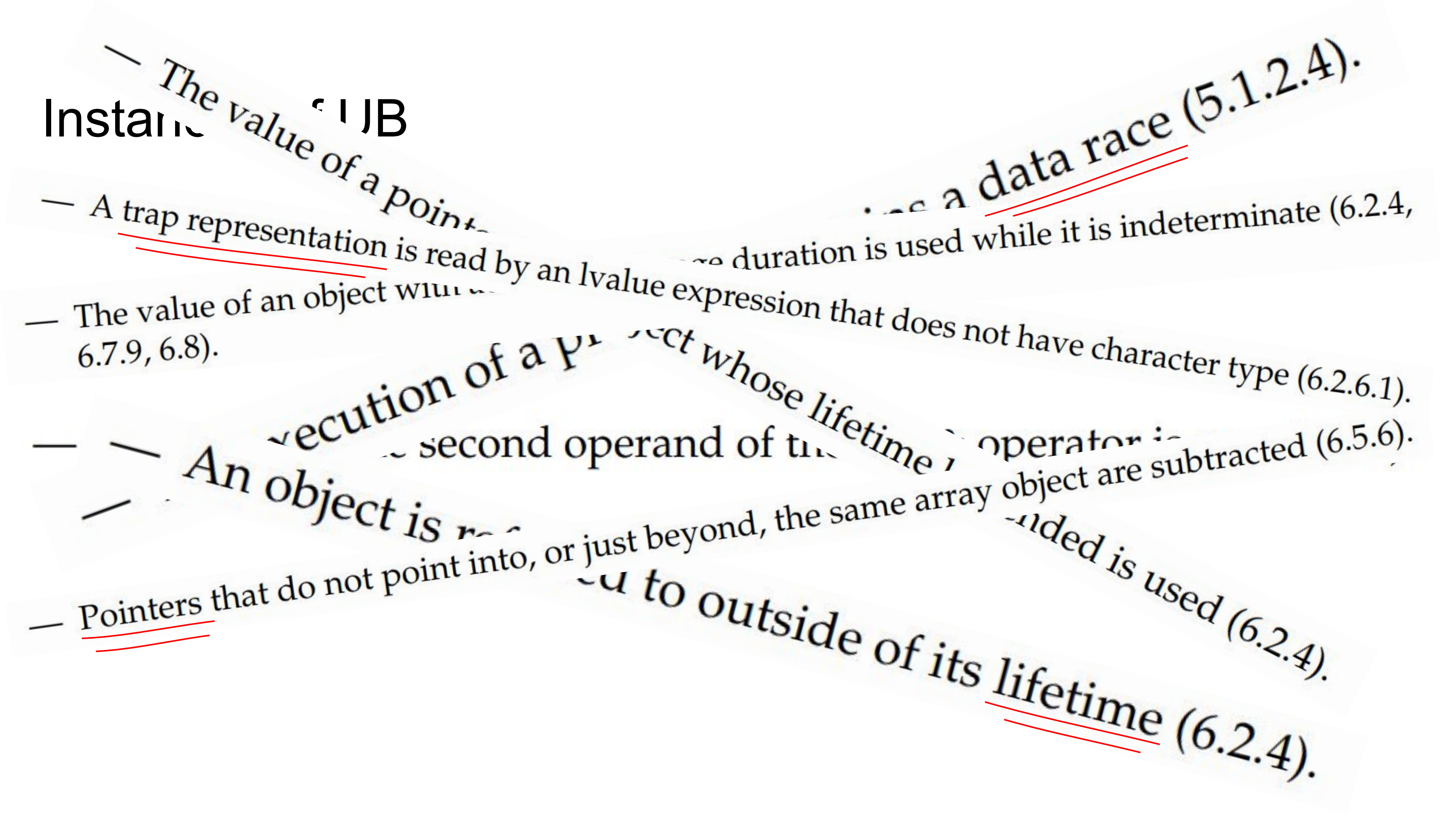
— ecution of a pr es a data race (5.1.2.4).

— second operand of th operator i

— An object is re d to outside of its lifetime (6.2.4).

— Pointers that do not point into, or just beyond, the same array object are subtracted (6.5.6).

Instan UB



— The value of a poin ect whose lifetime operator i nded is used (6.2.4).

— A trap representation is read by an lvalue expression that does not have character type (6.2.6.1).

— The value of an object with ge duration is used while it is indeterminate (6.2.4, 6.7.9, 6.8).

— xecution of a pr es a data race (5.1.2.4).

— e second operand of th

— An object is re d to outside of its lifetime (6.2.4).

— Pointers that do not point into, or just beyond, the same array object are subtracted (6.5.6).

So, what does Rust offer?

So, what does Rust offer?

~~UB~~

Safety

Safety in Rust

=

No undefined behavior

Safety

Safety in Rust

≠

Safety in “safety-critical”

as in functional safety (DO-178B/C, ISO 26262, EN 50128…)

✈ 🚕 🚆

Safety examples

⇒

`abort()`s in C

are

Rust-safe

Safety examples

⇒

`abort()`s in C

are

Rust-safe

Even if your company goes bankrupt.

Safety examples

⇒ `abort()`s in C

are

Rust-safe

Even if your company goes bankrupt.

Even if somebody is injured.

Avoiding UB

```
int f(int a, int b) {
    if (b == 0)
        abort();

    if (a == INT_MIN && b == -1)
        abort();

    return a / b;
}
```

Avoiding UB

```
int f(int a, int b) {
    if (b == 0)
        abort();

    if (a == INT_MIN && b == -1)
        abort();

    return a / b;
}
```

f is a safe function

Safety examples

⇒

Rust panics

are

Rust-safe

Safety examples

⇒

Kernel panics

are

Rust-safe

Safety examples

⇒

Uses after free, null derefs, double frees,

OOB accesses, uninitialized memory reads,

invalid inhabitants, data races...

are **not**

Rust-safe

Safety examples

⇒

Uses after free, null derefs, double frees,

OOB accesses, uninitialized memory reads,

invalid inhabitants, data races...

are **not**

Rust-safe

Even if your system still works.

Safety examples

⇒

Race conditions

are

Rust-safe

Safety examples

⇒

Memory leaks

are

Rust-safe

Safety examples

⇒

Deadlocks

are

Rust-safe

Safety examples

⇒

Integer overflows

are

Rust-safe

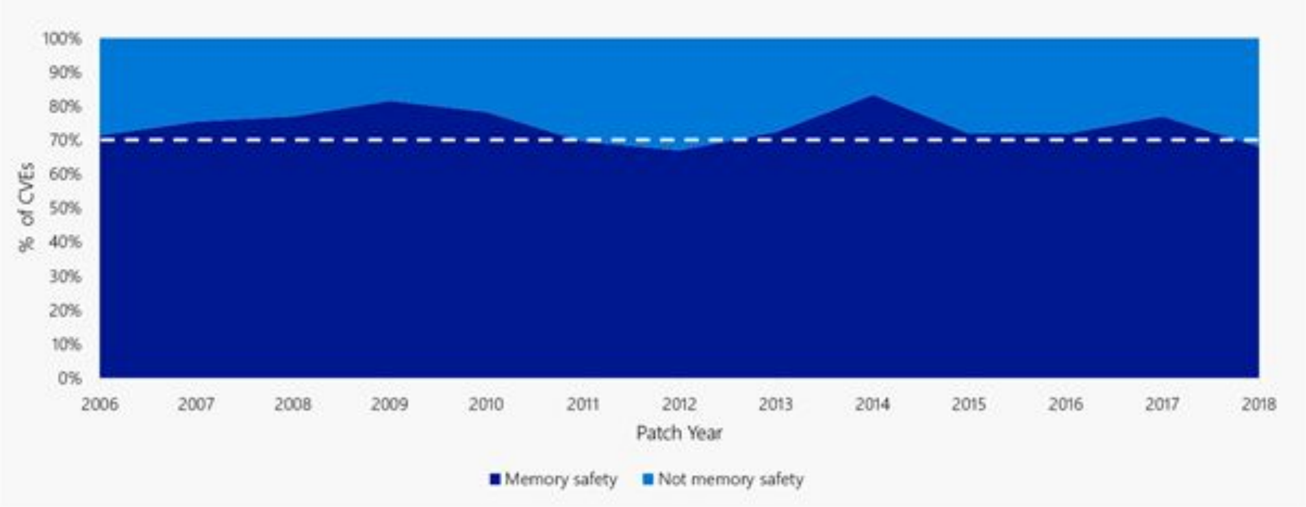
Is avoiding UB that important?

Is avoiding UB that important?

~70%

of vulnerabilities in C/C++ projects come from UB

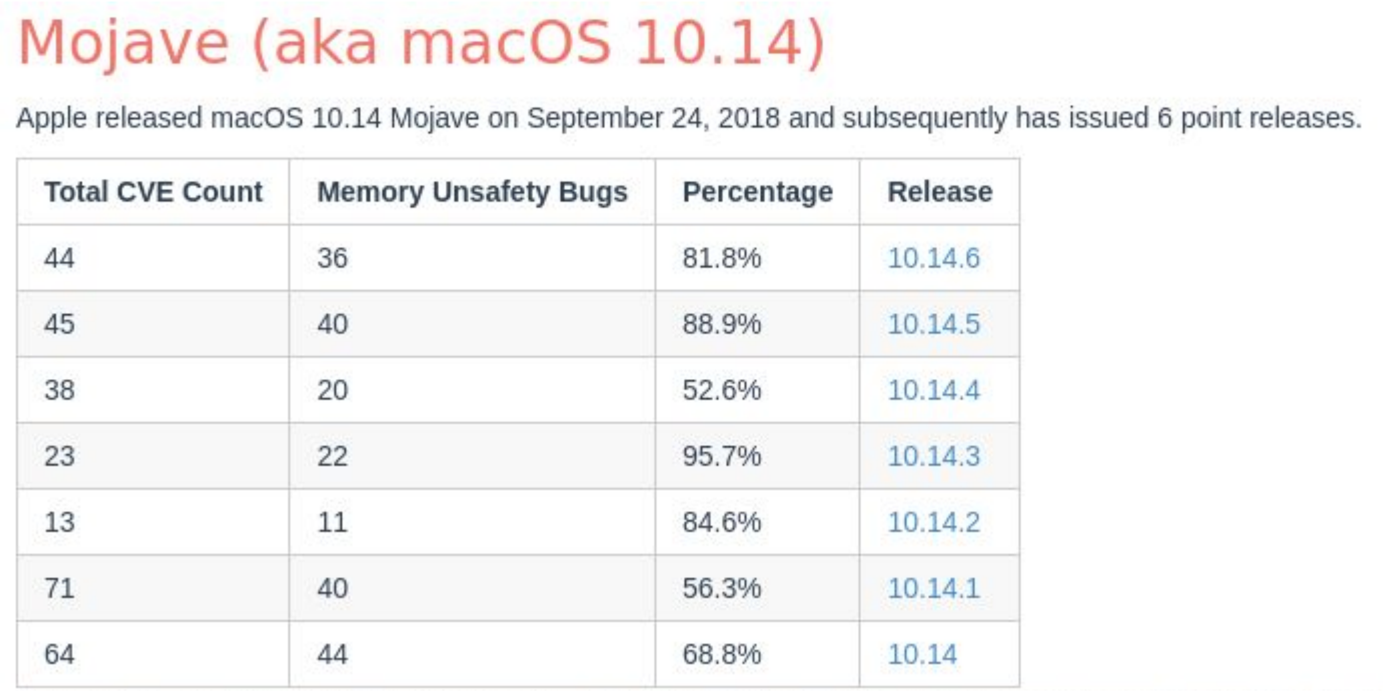
Is avoiding UB that important?



~70% of the vulnerabilities Microsoft assigns a CVE each year continue to be memory safety issues

— https://msrc-blog.microsoft.com/2019/07/18/we-need-a-safer-systems-programming-language/

Is avoiding UB that important?



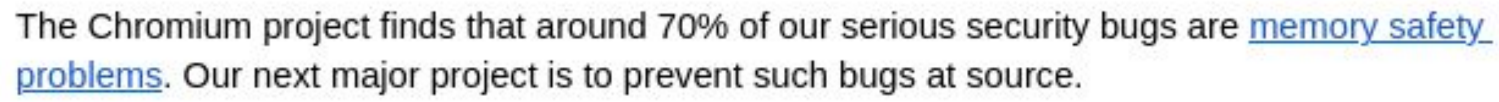
Mojave (aka macOS 10.14)

Apple released macOS 10.14 Mojave on September 24, 2018 and subsequently has issued 6 point releases.

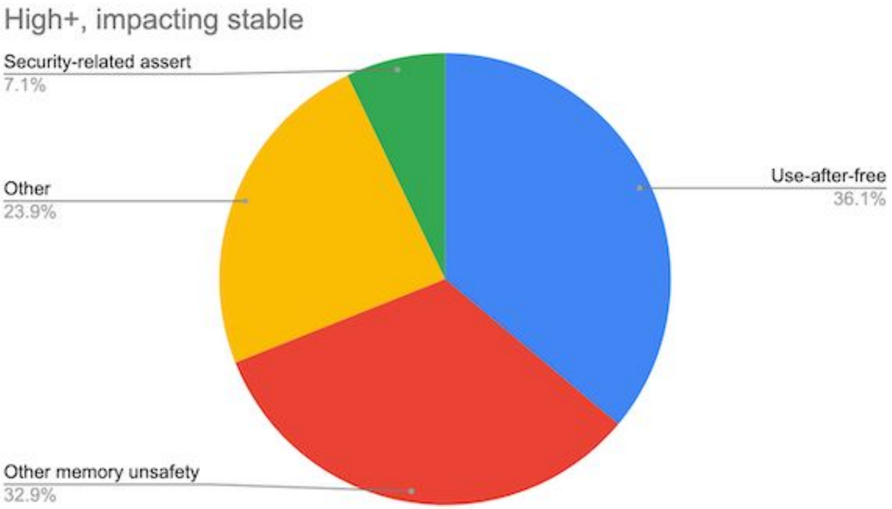
Total CVE Count	Memory Unsafety Bugs	Percentage	Release
44	36	81.8%	10.14.6
45	40	88.9%	10.14.5
38	20	52.6%	10.14.4
23	22	95.7%	10.14.3
13	11	84.6%	10.14.2
71	40	56.3%	10.14.1
64	44	68.8%	10.14

— https://langui.sh/2019/07/23/apple-memory-safety/

Is avoiding UB that important?

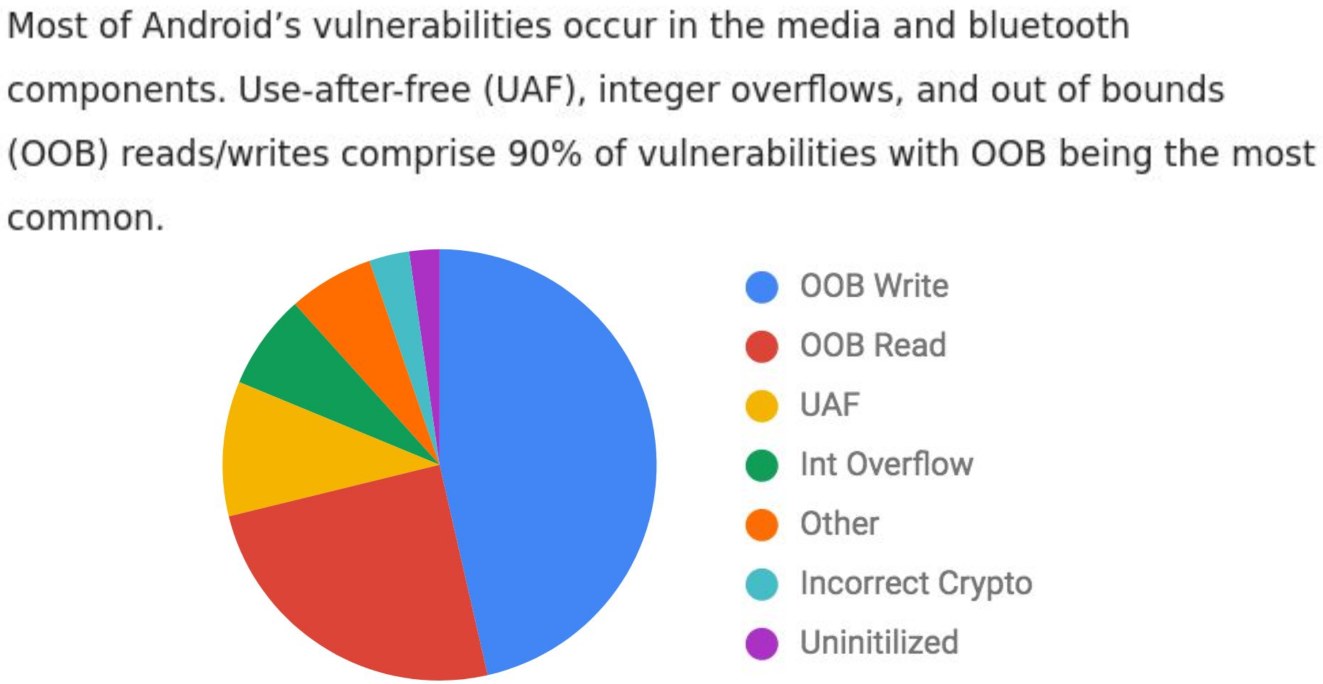


The Chromium project finds that around 70% of our serious security bugs are memory safety problems. Our next major project is to prevent such bugs at source.



— https://www.chromium.org/Home/chromium-security/memory-safety

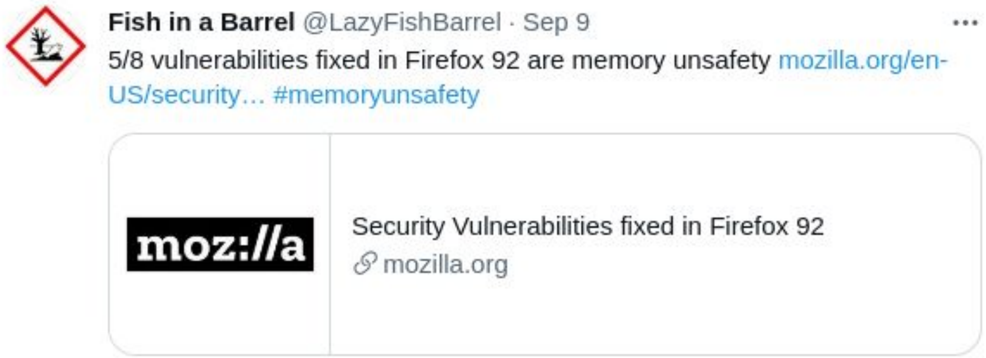
Is avoiding UB that important?



Most of Android's vulnerabilities occur in the media and bluetooth components. Use-after-free (UAF), integer overflows, and out of bounds (OOB) reads/writes comprise 90% of vulnerabilities with OOB being the most common.

— https://security.googleblog.com/2019/05/queue-hardening-enhancements.html

Is avoiding UB that important?

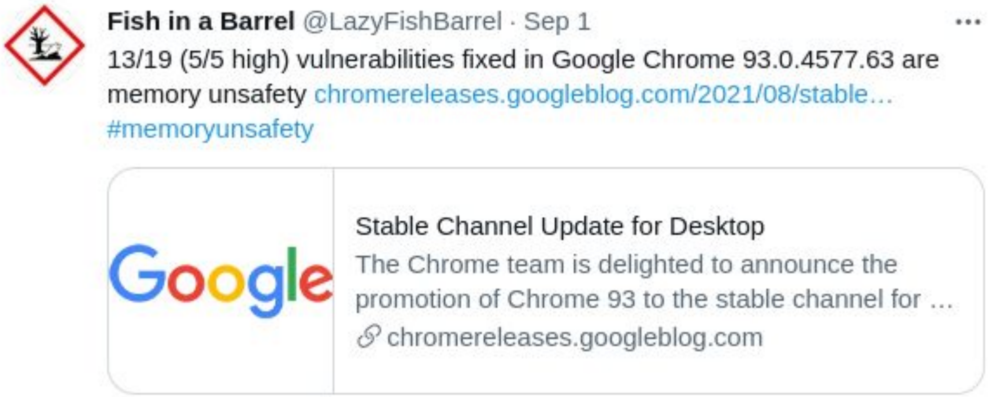


Fish in a Barrel @LazyFishBarrel · Sep 9

5/8 vulnerabilities fixed in Firefox 92 are memory unsafety mozilla.org/en-US/security… #memoryunsafety

Security Vulnerabilities fixed in Firefox 92

mozilla.org



Fish in a Barrel @LazyFishBarrel · Sep 1

13/19 (5/5 high) vulnerabilities fixed in Google Chrome 93.0.4577.63 are memory unsafety chromereleases.googleblog.com/2021/08/stable… #memoryunsafety

Stable Channel Update for Desktop

The Chrome team is delighted to announce the promotion of Chrome 93 to the stable channel for …

chromereleases.googleblog.com

Sure, UB is an issue and safe Rust does not have it…

Sure, UB is an issue and safe Rust does not have it…

...but does Rust really help, though?

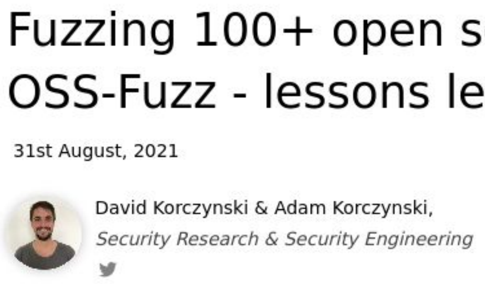
Does Rust help?

Does Rust help?

I took a look at this spreadsheet published three weeks ago...

Does Rust help?

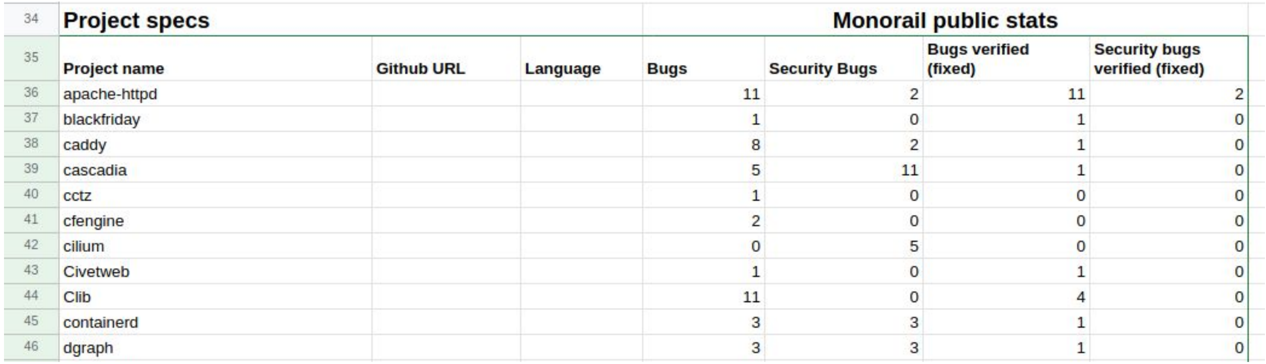
I took a look at this spreadsheet published three weeks ago...



Fuzzing 100+ open source projects with OSS-Fuzz - lessons learned.

31st August, 2021

David Korczynski & Adam Korczynski,
Security Research & Security Engineering



34	Project specs			Monorail public stats			
35	Project name	Github URL	Language	Bugs	Security Bugs	Bugs verified (fixed)	Security bugs verified (fixed)
36	apache-httpd			11	2	11	2
37	blackfriday			1	0	1	0
38	caddy			8	2	1	0
39	cascadia			5	11	1	0
40	cctz			1	0	0	0
41	cfengine			2	0	0	0
42	cilium			0	5	0	0
43	Civetweb			1	0	1	0
44	Clib			11	0	4	0
45	containerd			3	3	1	0
46	dgraph			3	3	1	0

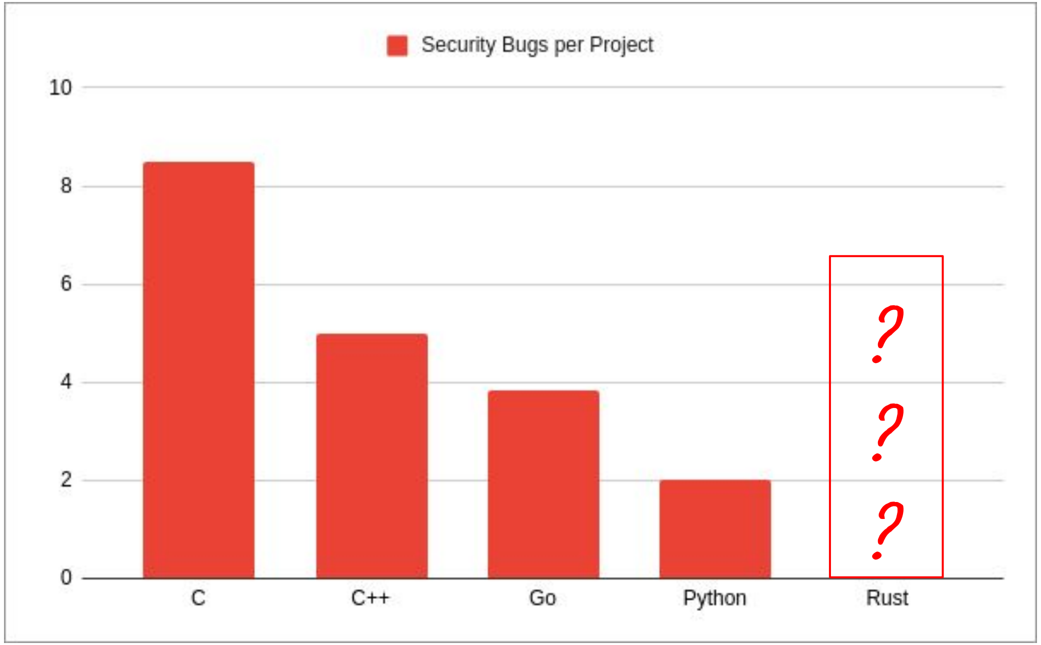
— https://adalogics.com/blog/fuzzing-100-open-source-projects-with-oss-fuzz

Does Rust help?

I filled the language column and plotted...

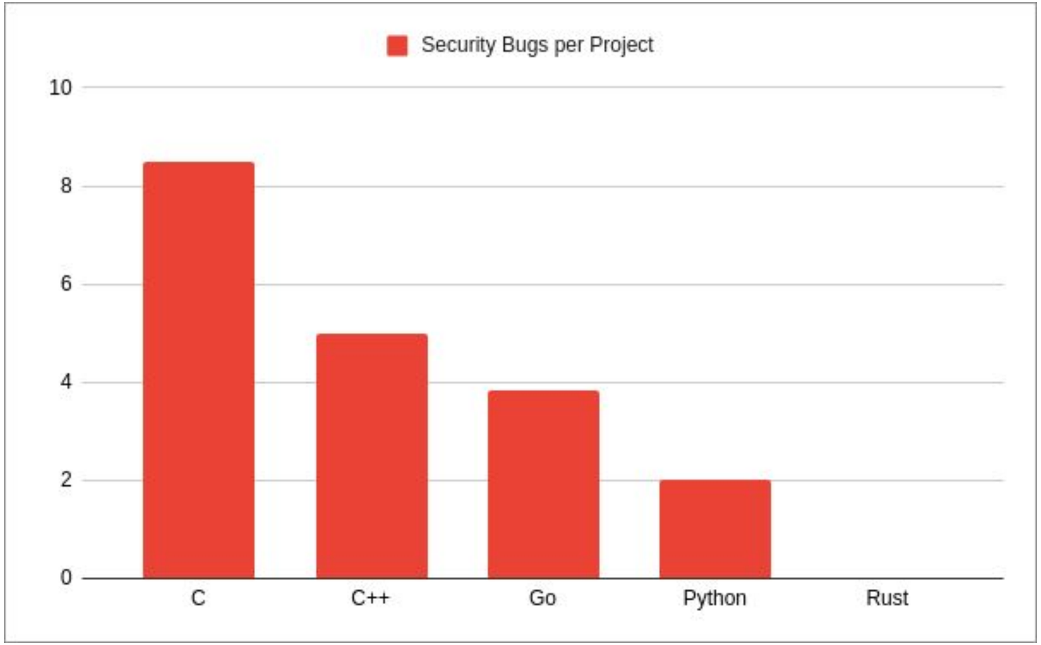
Does Rust help?

I filled the language column and plotted...



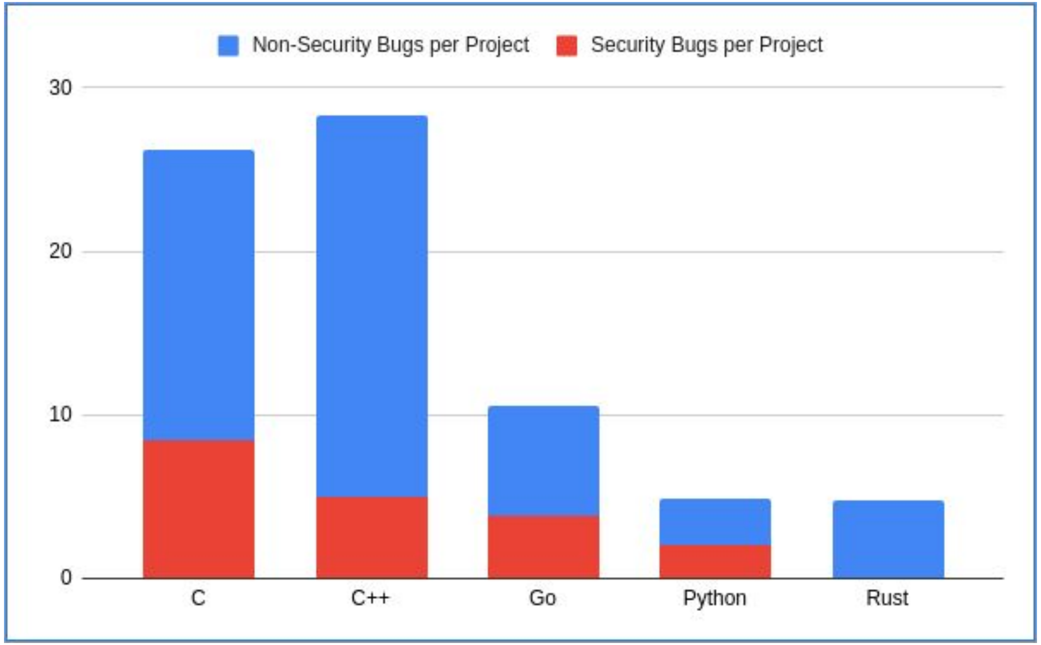
Does Rust help?

I filled the language column and plotted...



Does Rust help?

I filled the language column and plotted...



What else does Rust offer?

Language

What else does Rust offer?

Shared & exclusive references

Modules & visibility

Generics

Lifetimes

Stricter type system

Language

Pattern matching

Safe/unsafe split

RAII

Sum types

Powerful hygienic and procedural macros

What else does Rust offer?

Standard library

What else does Rust offer?

Pinning

Vocabulary types like `Result` and `Option`

Formatting

Networking

Standard library

Collections

Iterators

Processes & Threads

Checked, saturating & wrapping integer arithmetic primitives

Paths & Filesystem

What else does Rust offer?

Tooling

What else does Rust offer?

Documentation generator

Unit & integration tests

Static analyzer

Build system

C ↔ Rust bindings generators

Linter

Tooling

Macro debugging

Formatter

IDE tooling

Great compiler error messages

UBSAN-like interpreter

What else does Rust offer?

Documentation generator

Unit & integration tests

Static analyzer

Build system

C ↔ Rust bindings generators

Linter

Tooling

Macro debugging

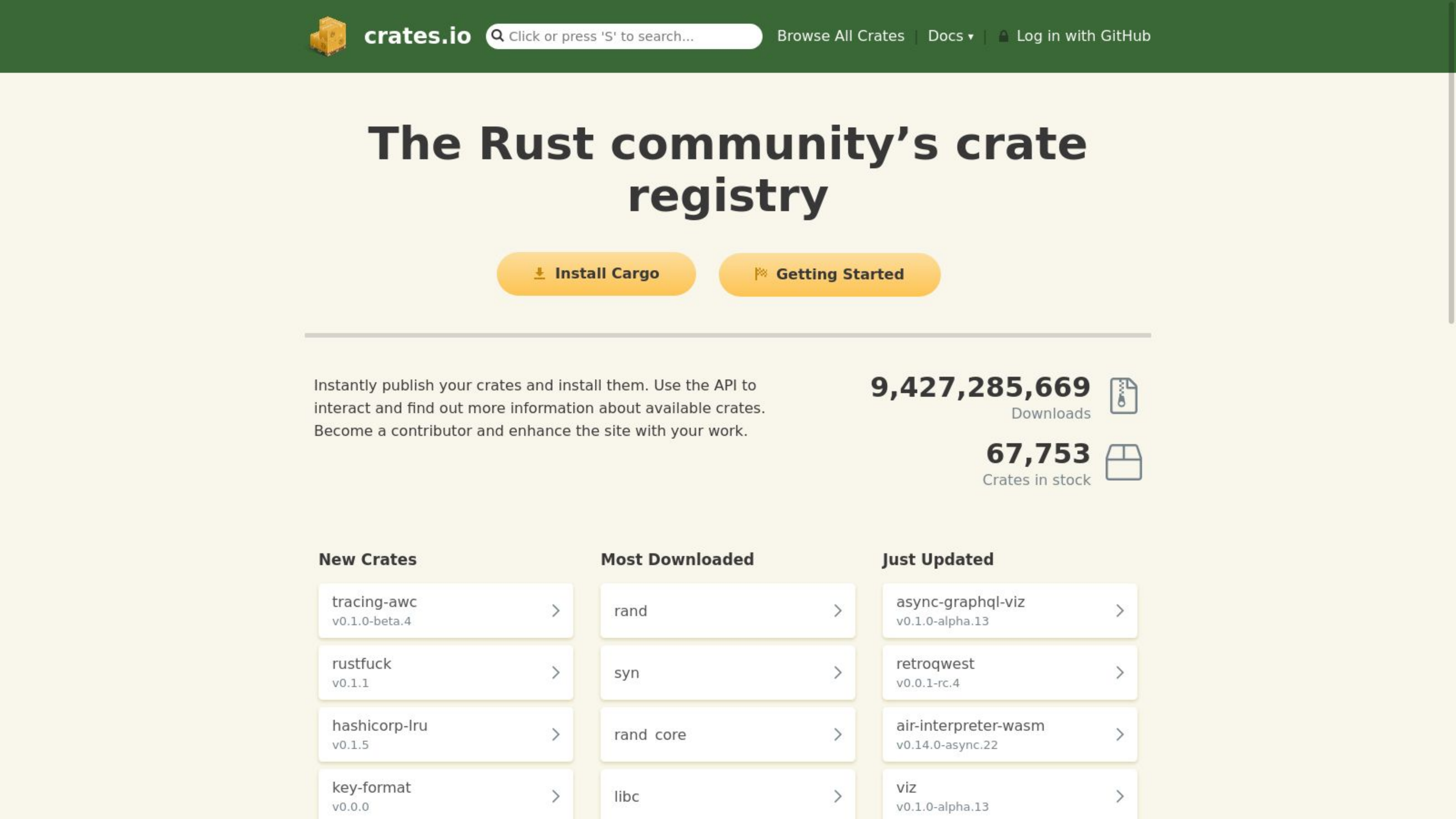
Formatter

IDE tooling

Great compiler error messages

UBSAN-like interpreter

plus the usual friends: `gdb`, `lldb`, `perf`, `valgrind`*...*



crates.io

Click or press 'S' to search...

Browse All Crates | Docs ▾ | Log in with GitHub

The Rust community’s crate registry

Install Cargo

Getting Started

Instantly publish your crates and install them. Use the API to interact and find out more information about available crates. Become a contributor and enhance the site with your work.

9,427,285,669
Downloads

67,753
Crates in stock

New Crates

- tracing-awc
  v0.1.0-beta.4
- rustfuck
  v0.1.1
- hashicorp-lru
  v0.1.5
- key-format
  v0.0.0

Most Downloaded

- rand
- syn
- rand core
- libc

Just Updated

- async-graphql-viz
  v0.1.0-alpha.13
- retroqwest
  v0.0.1-rc.4
- air-interpreter-wasm
  v0.14.0-async.22
- viz
  v0.1.0-alpha.13

What is the catch?

What is the catch?

Cannot model everything ⇒ Unsafe code required

What is the catch?

Cannot model everything ⇒ Unsafe code required

More information to provide ⇒ More complex language

What is the catch?

Cannot model everything ⇒ Unsafe code required

More information to provide ⇒ More complex language

Extra runtime checks ⇒ Potentially expensive

What is the catch?

Cannot model everything	⇒	Unsafe code required
More information to provide	⇒	More complex language
Extra runtime checks	⇒	Potentially expensive
An extra language to learn	⇒	Logistics & maintenance burden

Why is C a good system programming language?

“You can use C to generate good code for hardware.”

Fast

“When I read C, I know what the assembly language will look like.”

Low-level

“The people that designed C ... designed it at a time when compilers had to be simple.”

Simple

“If you think like a computer, writing C actually makes sense.”

Fits the domain

Rust

Why is C a good system programming language?

“You can use C to generate good code for hardware.”

Fast Yes

“When I read C, I know what the assembly language will look like.”

Low-level Sometimes

“The people that designed C ... designed it at a time when compilers had to be simple.”

Simple Not really

“If you think like a computer, writing C actually makes sense.”

Fits the domain ...

Who is using Rust?

Production users

HealPay

We have been pushing Rust into every project we can. Currently we have several backend services built in rust.

Metaswitch

Rust is the primary programming language for all communications solutions architected using microservices methodologies.

360dialog

Most of our service consumers are written with Rust.

moz://a

Mozilla

Building the Servo browser engine, integrating into Firefox, other projects.

ATLASSIAN

Atlassian

We use Rust in a service for analyzing petabytes of source code.

system76

System76

As a Linux-based computer-manufacture, much of our infrastructure and desktop Linux projects are written in Rust. From hardware certification, flashing, and imaging; to system services and GTK3 desktop applications.

FIRE EMERGENCY

Fire and Emergency NZ

The New Zealand Fire Service is using a custom geolocation search engine, built in rust, that runs on embedded hardware within a fire truck to stream hazard information to a fire crew at an incident.

Dropbox

Dropbox

Optimizing cloud file-storage.

CLOUDFLARE

Cloudflare

We are using Rust as a replacement for memory-unsafe languages (particularly C) and are using it in our core edge logic.

1Password

1Password

We use Rust to power the entire backend (encryption, networking, database, and business logic) of all our client apps.

deliveroo

Deliveroo

We are using Rust to quickly make assignment decisions in our food delivery network.

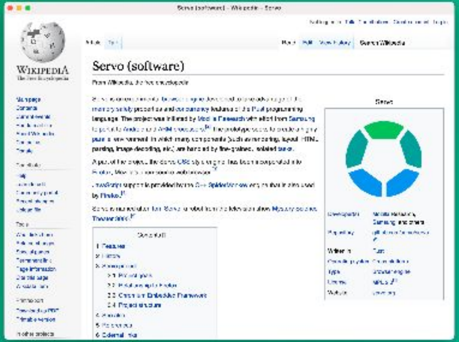
CANONICAL

Canonical

Everything from server monitoring to middleware!

Projects written in Rust

servo How to start Contributing Blog Governance GitHub Twitter


Servo (software)

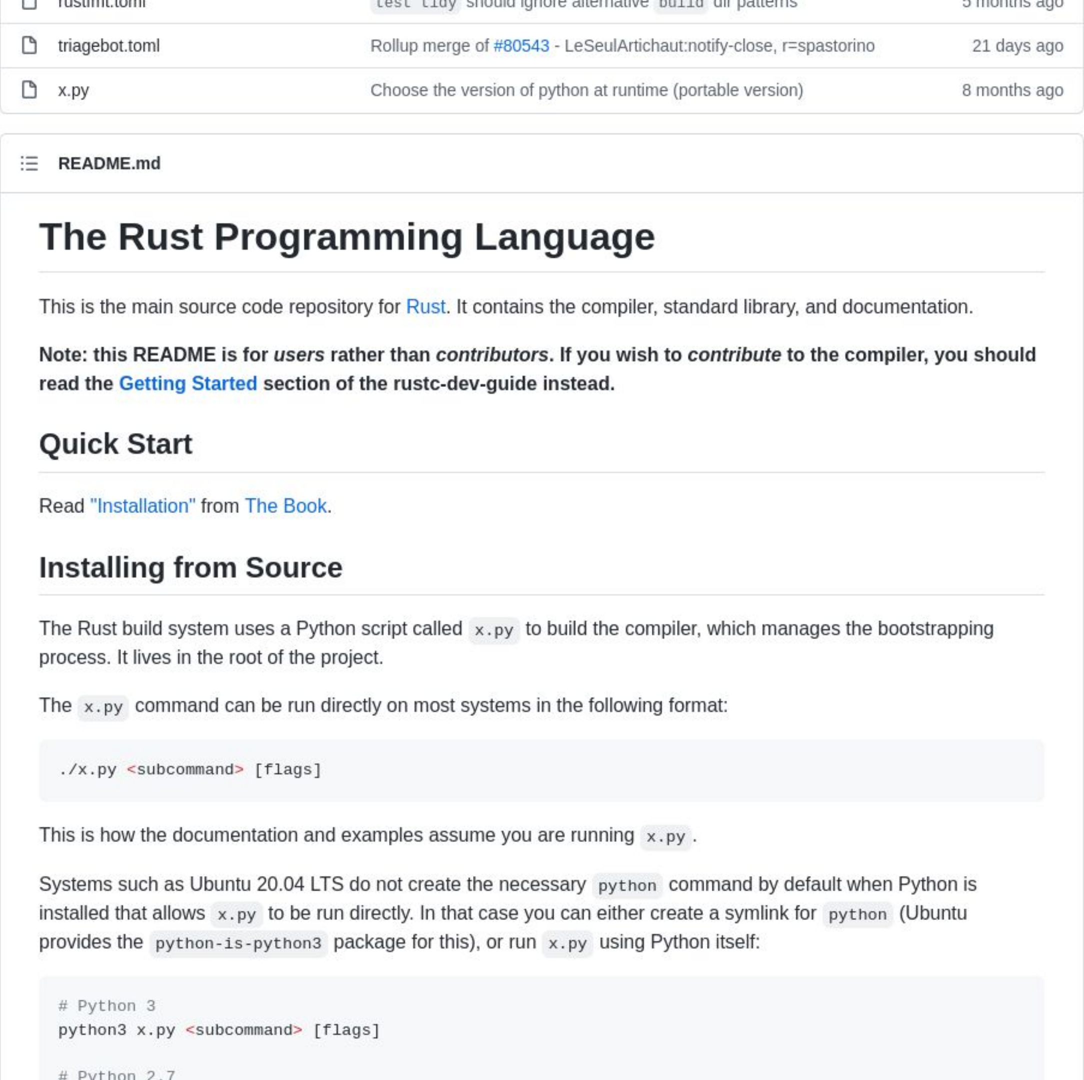
From Wikipedia, the free encyclopedia

Servo’s mission is to provide an independent, modular, embeddable web engine, which allows developers to deliver content and applications using web standards.

Servo is written in Rust, and shares code with Mozilla Firefox and the wider Rust ecosystem. Since its creation in 2012, Servo has contributed to W3C/WHATWG web standards by reporting specification issues and submitting new cross-browser automated tests, and core team members have co-edited new standards that have been adopted by other browsers. As a result, the Servo project helps drive the entire web platform forward while building on a platform of reusable, modular technologies that implement web standards.

Support and donations

Interested in helping out the Servo Project? Please do! You could write code or documentation, test nightlies and file issues, or donate to help cover the project’s CI and hosting costs. If you know a company that would like to support the Servo Project, please have a look at our Participation Agreement get in touch: info@servo.org.



rustfmt.toml	`test tidy` should ignore alternative `build` dir patterns	5 months ago
triagebot.toml	Rollup merge of #80543 - LeSeulArtichaut:notify-close, r=spastorino	21 days ago
x.py	Choose the version of python at runtime (portable version)	8 months ago

README.md

The Rust Programming Language

This is the main source code repository for Rust. It contains the compiler, standard library, and documentation.

Note: this README is for *users* rather than *contributors*. If you wish to *contribute* to the compiler, you should read the Getting Started section of the rustc-dev-guide instead.

Quick Start

Read "Installation" from The Book.

Installing from Source

The Rust build system uses a Python script called `x.py` to build the compiler, which manages the bootstrapping process. It lives in the root of the project.

The `x.py` command can be run directly on most systems in the following format:

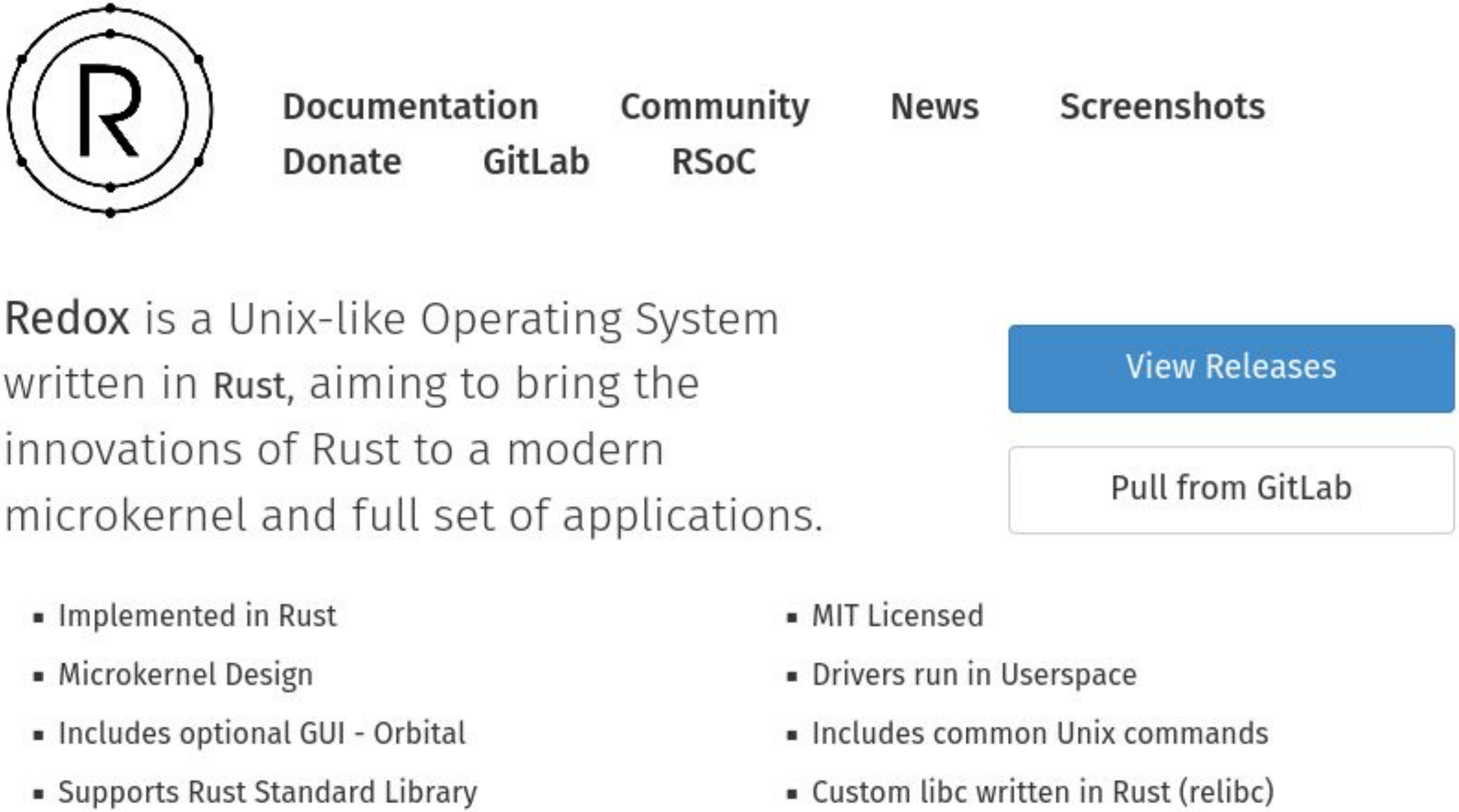
```
./x.py <subcommand> [flags]
```

This is how the documentation and examples assume you are running `x.py`.

Systems such as Ubuntu 20.04 LTS do not create the necessary `python` command by default when Python is installed that allows `x.py` to be run directly. In that case you can either create a symlink for `python` (Ubuntu provides the `python-is-python3` package for this), or run `x.py` using Python itself:

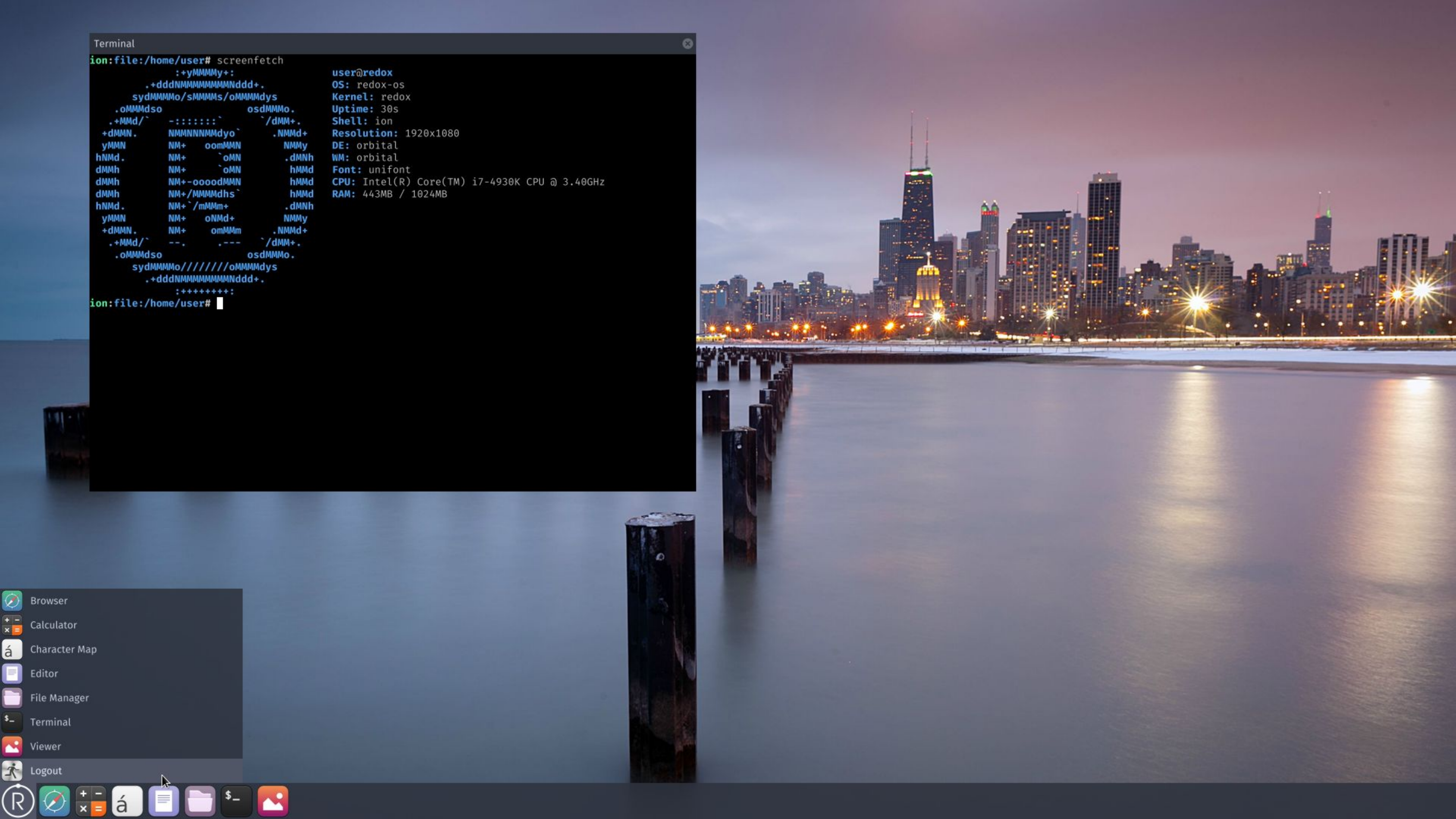
```
# Python 3
python3 x.py <subcommand> [flags]

# Python 2.7
```

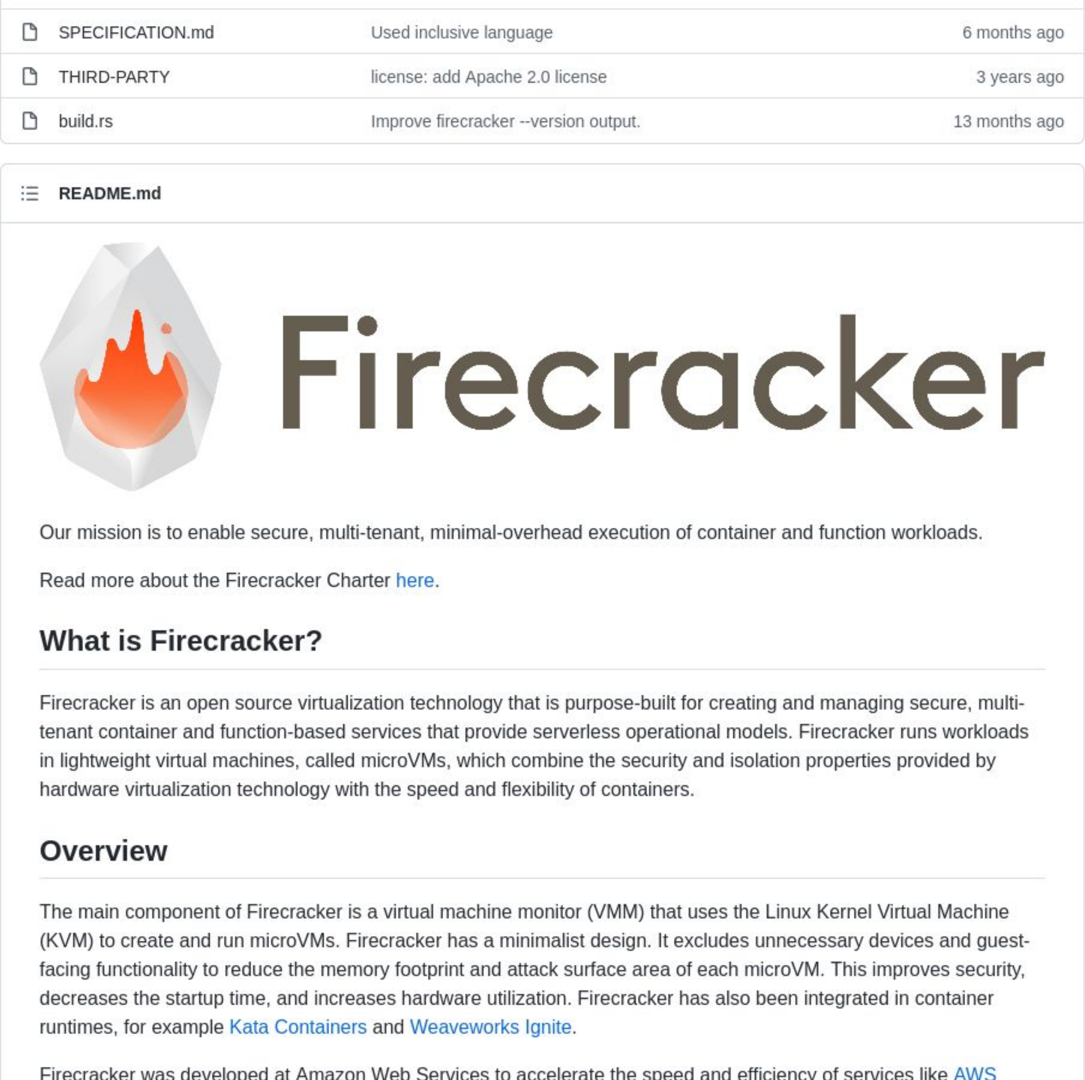


Redox is a Unix-like Operating System written in **Rust**, aiming to bring the innovations of Rust to a modern microkernel and full set of applications.

- Implemented in Rust
- Microkernel Design
- Includes optional GUI - Orbital
- Supports Rust Standard Library
- MIT Licensed
- Drivers run in Userspace
- Includes common Unix commands
- Custom libc written in Rust (relibc)



```
ion:file:/home/user# screenfetch
              :+yMMMMy+:                user@redox
         .+dddNMMMMMMMMMNddd+.          OS: redox-os
       sydMMMMo/sMMMMs/oMMMMdys         Kernel: redox
    .oMMMdso              osdMMMo.      Uptime: 30s
   .+MMd/`   -:::::::`     `/dMM+.     Shell: ion
  +dMMN.     NMMNNNMMdyo`     .NMMd+    Resolution: 1920x1080
  yMMN       NM+   oomMMN       NMMy    DE: orbital
 hNMd.       NM+     `oMN       .dMNh   WM: orbital
 dMMh        NM+     `oMN        hMMd   Font: unifont
 dMMh        NM+-oooodMMN        hMMd   CPU: Intel(R) Core(TM) i7-4930K CPU @ 3.40GHz
 dMMh        NM+/MMMMdhs`        hMMd   RAM: 443MB / 1024MB
 hNMd.       NM+`/mMMm+         .dMNh
  yMMN       NM+   oNMd+        NMMy
  +dMMN.     NM+    omMMm     .NMMd+
   .+MMd/`   --.     .---   `/dMM+.
    .oMMMdso              osdMMMo.
       sydMMMMo////////oMMMMdys
         .+dddNMMMMMMMMMNddd+.
              :++++++++:
ion:file:/home/user#
```



SPECIFICATION.md	Used inclusive language	6 months ago
THIRD-PARTY	license: add Apache 2.0 license	3 years ago
build.rs	Improve firecracker --version output.	13 months ago

README.md

Our mission is to enable secure, multi-tenant, minimal-overhead execution of container and function workloads.

Read more about the Firecracker Charter here.

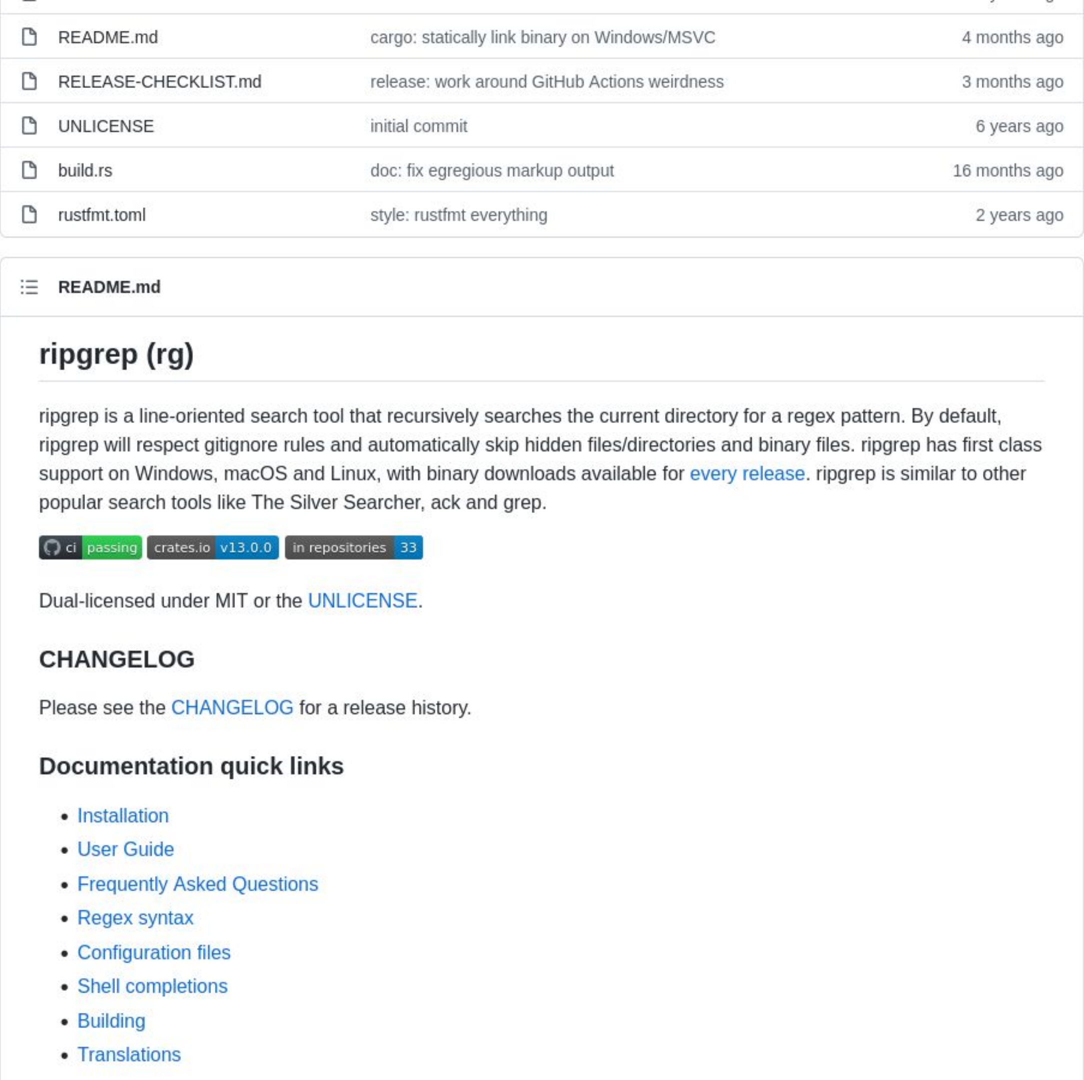
What is Firecracker?

Firecracker is an open source virtualization technology that is purpose-built for creating and managing secure, multi-tenant container and function-based services that provide serverless operational models. Firecracker runs workloads in lightweight virtual machines, called microVMs, which combine the security and isolation properties provided by hardware virtualization technology with the speed and flexibility of containers.

Overview

The main component of Firecracker is a virtual machine monitor (VMM) that uses the Linux Kernel Virtual Machine (KVM) to create and run microVMs. Firecracker has a minimalist design. It excludes unnecessary devices and guest-facing functionality to reduce the memory footprint and attack surface area of each microVM. This improves security, decreases the startup time, and increases hardware utilization. Firecracker has also been integrated in container runtimes, for example Kata Containers and Weaveworks Ignite.

Firecracker was developed at Amazon Web Services to accelerate the speed and efficiency of services like AWS



README.md	cargo: statically link binary on Windows/MSVC	4 months ago
RELEASE-CHECKLIST.md	release: work around GitHub Actions weirdness	3 months ago
UNLICENSE	initial commit	6 years ago
build.rs	doc: fix egregious markup output	16 months ago
rustfmt.toml	style: rustfmt everything	2 years ago

README.md

ripgrep (rg)

ripgrep is a line-oriented search tool that recursively searches the current directory for a regex pattern. By default, ripgrep will respect gitignore rules and automatically skip hidden files/directories and binary files. ripgrep has first class support on Windows, macOS and Linux, with binary downloads available for every release. ripgrep is similar to other popular search tools like The Silver Searcher, ack and grep.

ci passing crates.io v13.0.0 in repositories 33

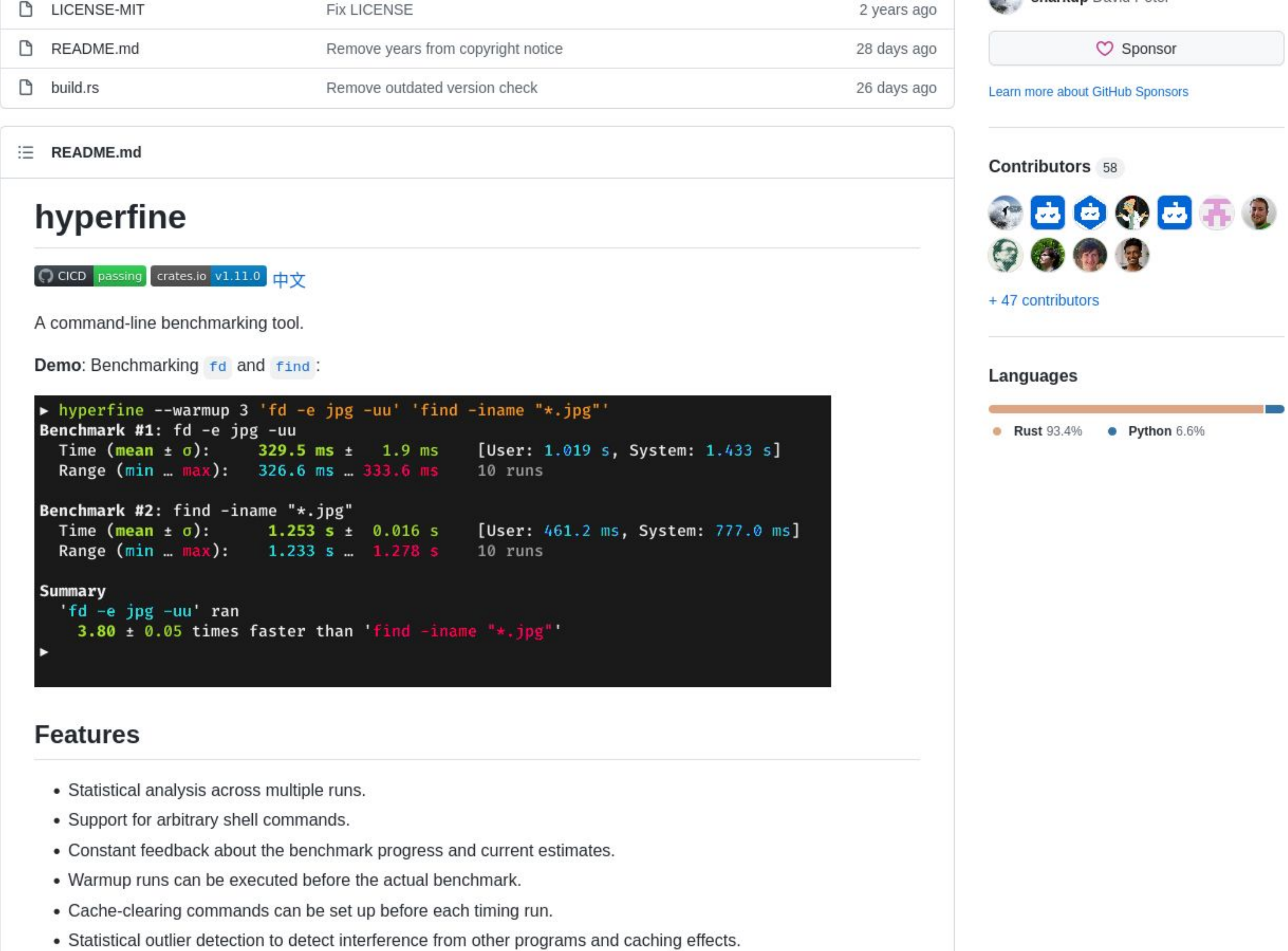
Dual-licensed under MIT or the UNLICENSE.

CHANGELOG

Please see the CHANGELOG for a release history.

Documentation quick links

- Installation
- User Guide
- Frequently Asked Questions
- Regex syntax
- Configuration files
- Shell completions
- Building
- Translations



LICENSE-MIT	Fix LICENSE	2 years ago
README.md	Remove years from copyright notice	28 days ago
build.rs	Remove outdated version check	26 days ago

README.md

hyperfine

CICD passing crates.io v1.11.0 中文

A command-line benchmarking tool.

Demo: Benchmarking `fd` and `find`:

```
▶ hyperfine --warmup 3 'fd -e jpg -uu' 'find -iname "*.jpg"'
Benchmark #1: fd -e jpg -uu
  Time (mean ± σ):     329.5 ms ±   1.9 ms    [User: 1.019 s, System: 1.433 s]
  Range (min … max):   326.6 ms … 333.6 ms    10 runs

Benchmark #2: find -iname "*.jpg"
  Time (mean ± σ):      1.253 s ±  0.016 s    [User: 461.2 ms, System: 777.0 ms]
  Range (min … max):    1.233 s …  1.278 s    10 runs

Summary
  'fd -e jpg -uu' ran
    3.80 ± 0.05 times faster than 'find -iname "*.jpg"'
▶
```

Features

- Statistical analysis across multiple runs.
- Support for arbitrary shell commands.
- Constant feedback about the benchmark progress and current estimates.
- Warmup runs can be executed before the actual benchmark.
- Cache-clearing commands can be set up before each timing run.
- Statistical outlier detection to detect interference from other programs and caching effects.

sharkdp David Peter

♡ Sponsor

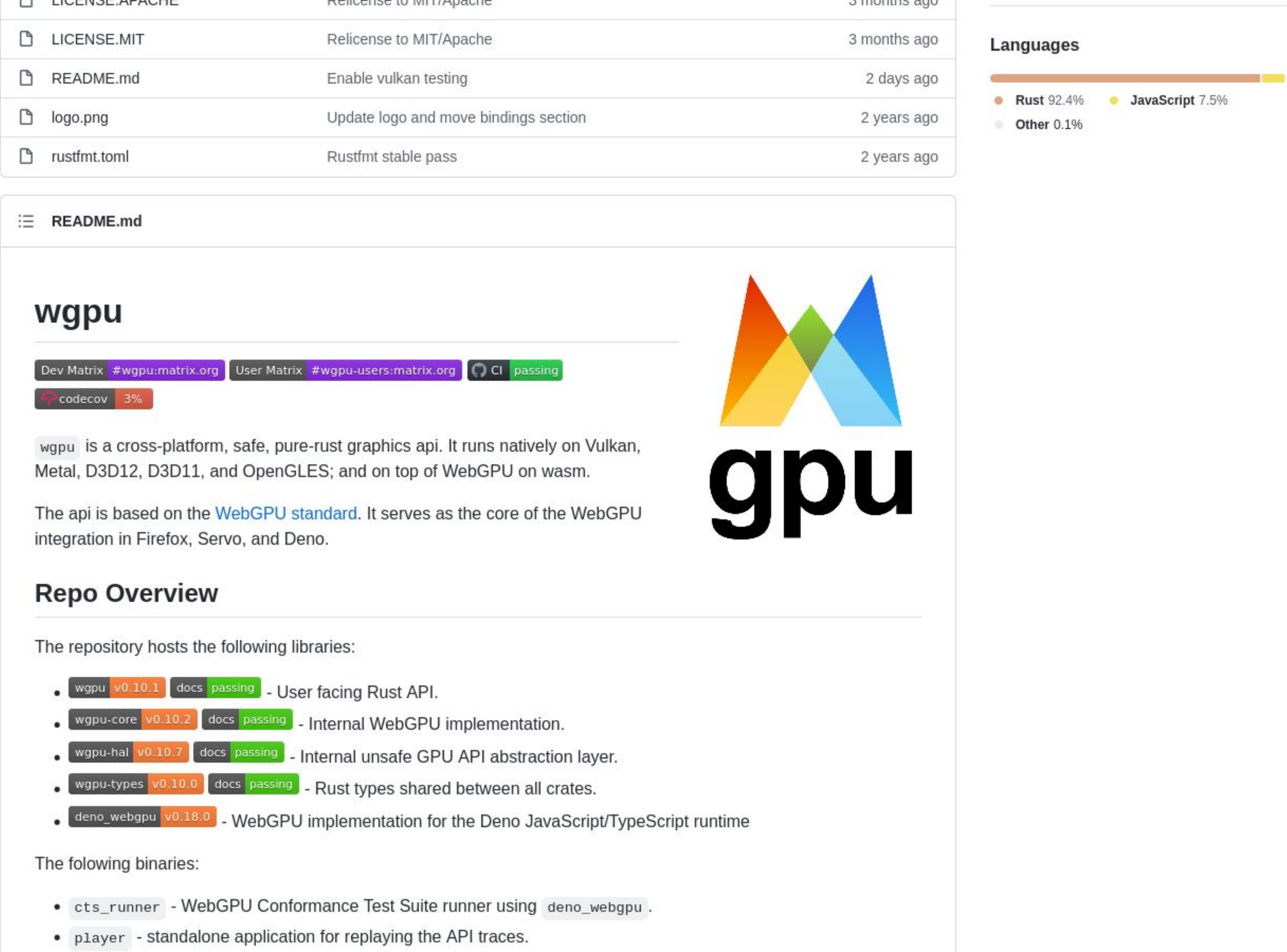
Learn more about GitHub Sponsors

Contributors 58

+ 47 contributors

Languages

● Rust 93.4% ● Python 6.6%



LICENSE.APACHE	Relicense to MIT/Apache	3 months ago
LICENSE.MIT	Relicense to MIT/Apache	3 months ago
README.md	Enable vulkan testing	2 days ago
logo.png	Update logo and move bindings section	2 years ago
rustfmt.toml	Rustfmt stable pass	2 years ago

Languages

● **Rust** 92.4% ● **JavaScript** 7.5%
● **Other** 0.1%

README.md

wgpu

Dev Matrix #wgpu:matrix.org User Matrix #wgpu-users:matrix.org CI passing
codecov 3%

`wgpu` is a cross-platform, safe, pure-rust graphics api. It runs natively on Vulkan, Metal, D3D12, D3D11, and OpenGLES; and on top of WebGPU on wasm.

The api is based on the WebGPU standard. It serves as the core of the WebGPU integration in Firefox, Servo, and Deno.

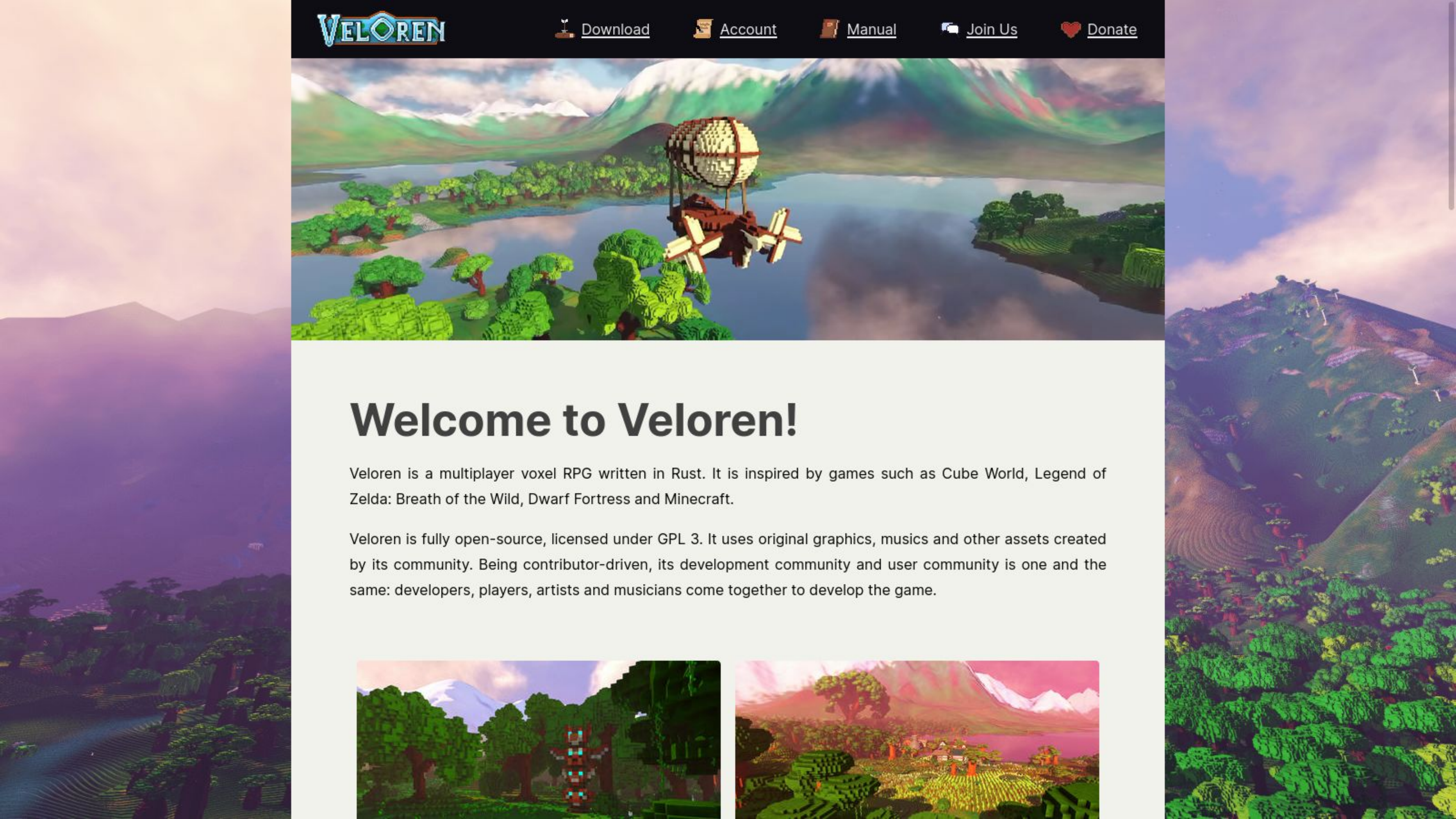
Repo Overview

The repository hosts the following libraries:

- wgpu v0.10.1 docs passing - User facing Rust API.
- wgpu-core v0.10.2 docs passing - Internal WebGPU implementation.
- wgpu-hal v0.10.7 docs passing - Internal unsafe GPU API abstraction layer.
- wgpu-types v0.10.0 docs passing - Rust types shared between all crates.
- deno_webgpu v0.18.0 - WebGPU implementation for the Deno JavaScript/TypeScript runtime

The folowing binaries:

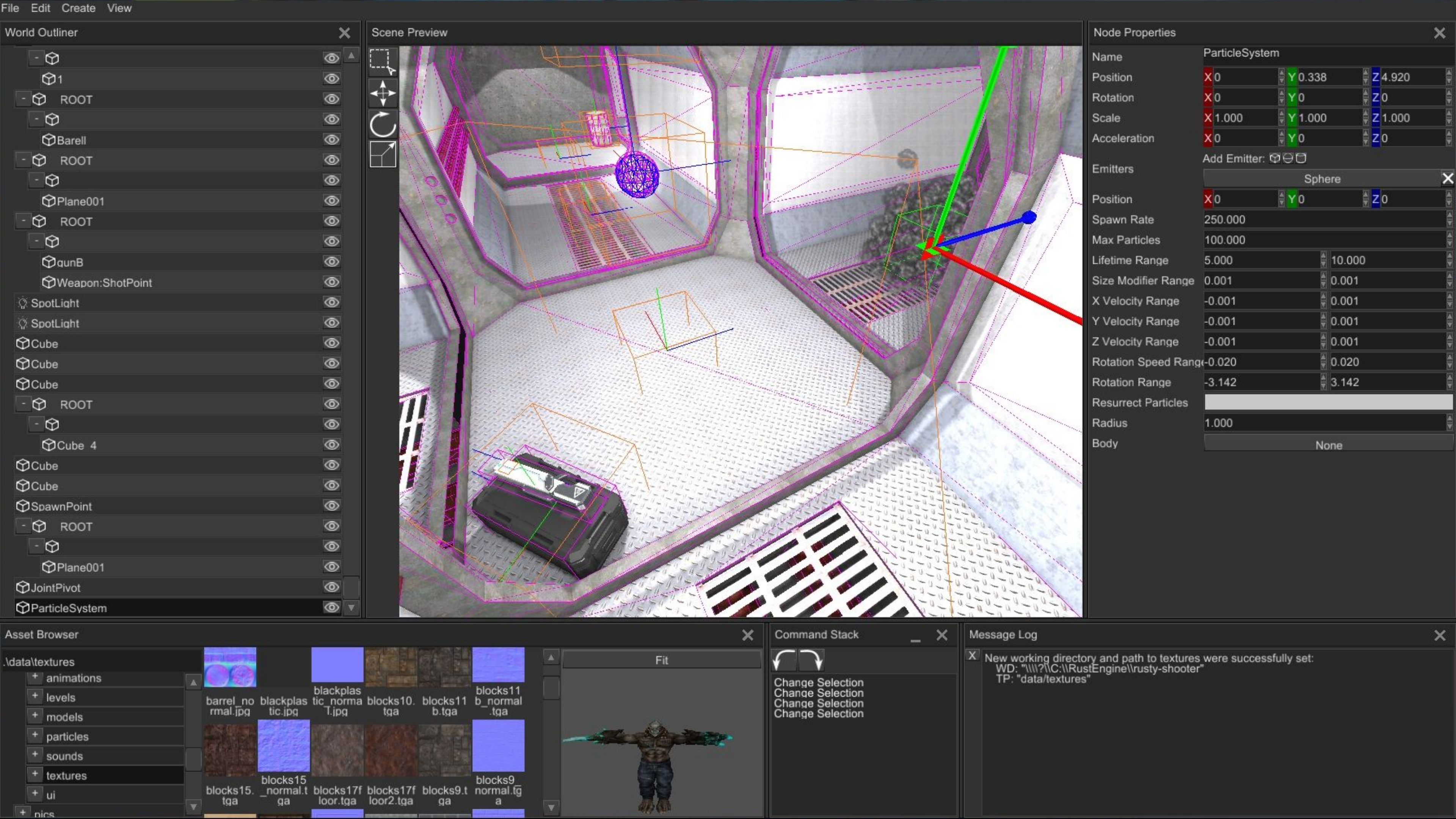
- `cts_runner` - WebGPU Conformance Test Suite runner using `deno_webgpu`.
- `player` - standalone application for replaying the API traces.



Welcome to Veloren!

Veloren is a multiplayer voxel RPG written in Rust. It is inspired by games such as Cube World, Legend of Zelda: Breath of the Wild, Dwarf Fortress and Minecraft.

Veloren is fully open-source, licensed under GPL 3. It uses original graphics, musics and other assets created by its community. Being contributor-driven, its development community and user community is one and the same: developers, players, artists and musicians come together to develop the game.



World Outliner:

```
- 
  1
- ROOT
  - 
    Barell
- ROOT
  - 
    Plane001
- ROOT
  - 
    gunB
    Weapon:ShotPoint
SpotLight
SpotLight
Cube
Cube
Cube
- ROOT
  - 
    Cube 4
Cube
Cube
SpawnPoint
- ROOT
  - 
    Plane001
JointPivot
ParticleSystem
```

Node Properties:

Name	ParticleSystem		
Position	X 0	Y 0.338	Z 4.920
Rotation	X 0	Y 0	Z 0
Scale	X 1.000	Y 1.000	Z 1.000
Acceleration	X 0	Y 0	Z 0
Emitters	Add Emitter:		
	Sphere		
Position	X 0	Y 0	Z 0
Spawn Rate	250.000		
Max Particles	100.000		
Lifetime Range	5.000	10.000	
Size Modifier Range	0.001	0.001	
X Velocity Range	-0.001	0.001	
Y Velocity Range	-0.001	0.001	
Z Velocity Range	-0.001	0.001	
Rotation Speed Rang	-0.020	0.020	
Rotation Range	-3.142	3.142	
Resurrect Particles			
Radius	1.000		
Body	None		

Asset Browser:

```
.\data\textures
+ animations
+ levels
+ models
+ particles
+ sounds
+ textures
+ ui
+ pics
```

barrel_normal.jpg
blackplastic.jpg
blackplastic_normal.jpg
blocks10.tga
blocks11b.tga
blocks11b_normal.tga
blocks15.tga
blocks15_normal.tga
blocks17floor.tga
blocks17floor2.tga
blocks9.tga
blocks9normal.tga

Command Stack:

```
Change Selection
Change Selection
Change Selection
Change Selection
```

Message Log:

```
X New working directory and path to textures were successfully set:
    WD: "\\\\?\\C:\\RustEngine\\rusty-shooter"
    TP: "data/textures"
```

Links

https://servo.org/

https://github.com/rust-lang/rust

https://www.redox-os.org/

https://github.com/firecracker-microvm/firecracker

https://github.com/BurntSushi/ripgrep

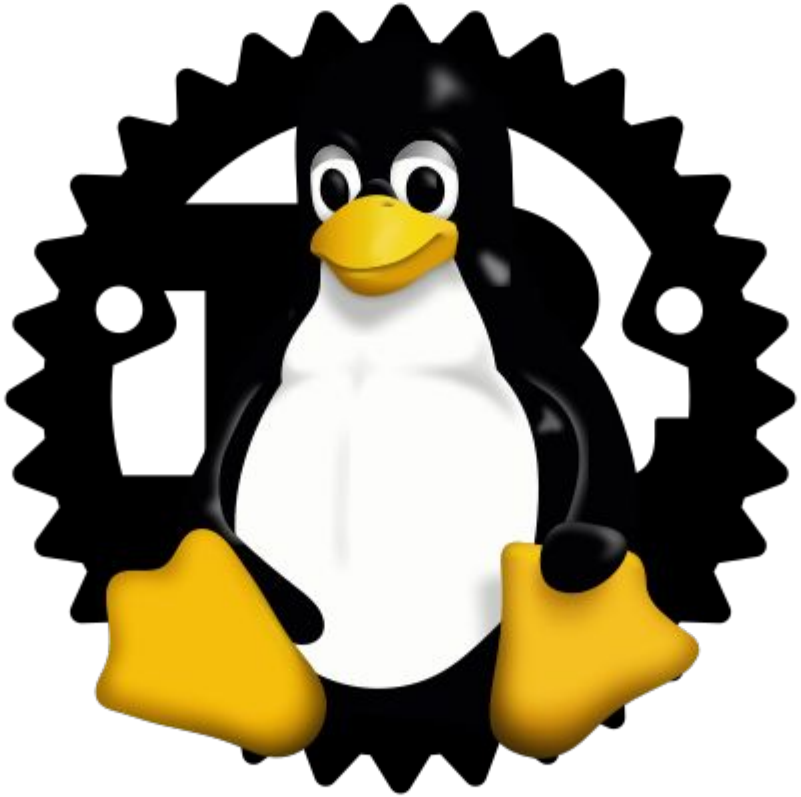
https://github.com/sharkdp/hyperfine

https://github.com/gfx-rs/wgpu

https://veloren.net/

https://rg3d.rs

Projects looking to take advantage of Rust



Rust for Linux

PROSSIMO Initiatives Blog About Support this Work

Memory safety for the Internet's most critical infrastructure

ⓘWhat is Memory Safety?

How We Work

Initiatives

NTP

Let's create a memory safe NTP implementation.

View initiative Project Status: Pending funding

NTP

curl

Let's make TLS and HTTP networking code in curl memory-safe.

View initiative Project Status: In progress

Rustls

Let's get the Rustls TLS library ready to replace OpenSSL in as many projects as possible.

View initiative Project Status: In progress

mod_tls

Let's make it possible to use memory safe TLS networking in Apache httpd.

View initiative Project Status: In progress

Entities supporting Rust

Members

Founding Platinum

aws

Google

HUAWEI

Microsoft

moz://a

Platinum

In the kernel...

“**Google** supports and contributes directly to the Rust for Linux project.

Our Android team is evaluating a new Binder implementation and considering other drivers where Rust could be adopted.”

— https://lore.kernel.org/lkml/20210704202756.29107-1-ojeda@kernel.org/

In the kernel...

“**Arm** recognises the Rust value proposition and is actively working with the Rust community to improve Rust for Arm based systems.

A good example is Arm’s RFC contribution to the Rust language which made Linux on 64-bit Arm systems a Tier-1 Rust supported platform.

Rustaceans at Arm are excited about the Rust for Linux initiative and look forward to assisting in this effort.”

arm

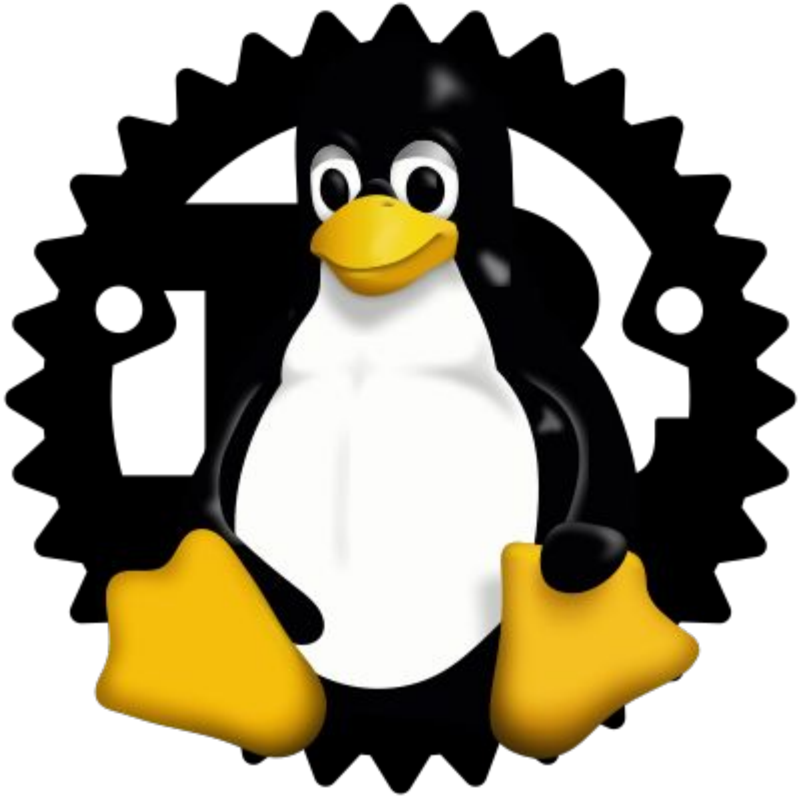
— https://lore.kernel.org/lkml/20210704202756.29107-1-ojeda@kernel.org/

In the kernel...

“**Microsoft**'s Linux Systems Group is interested in contributing to getting Rust into Linux kernel.

Hopefully we will be able to submit select Hyper-V drivers written in Rust in the coming months.”

— https://lore.kernel.org/lkml/20210704202756.29107-1-ojeda@kernel.org/

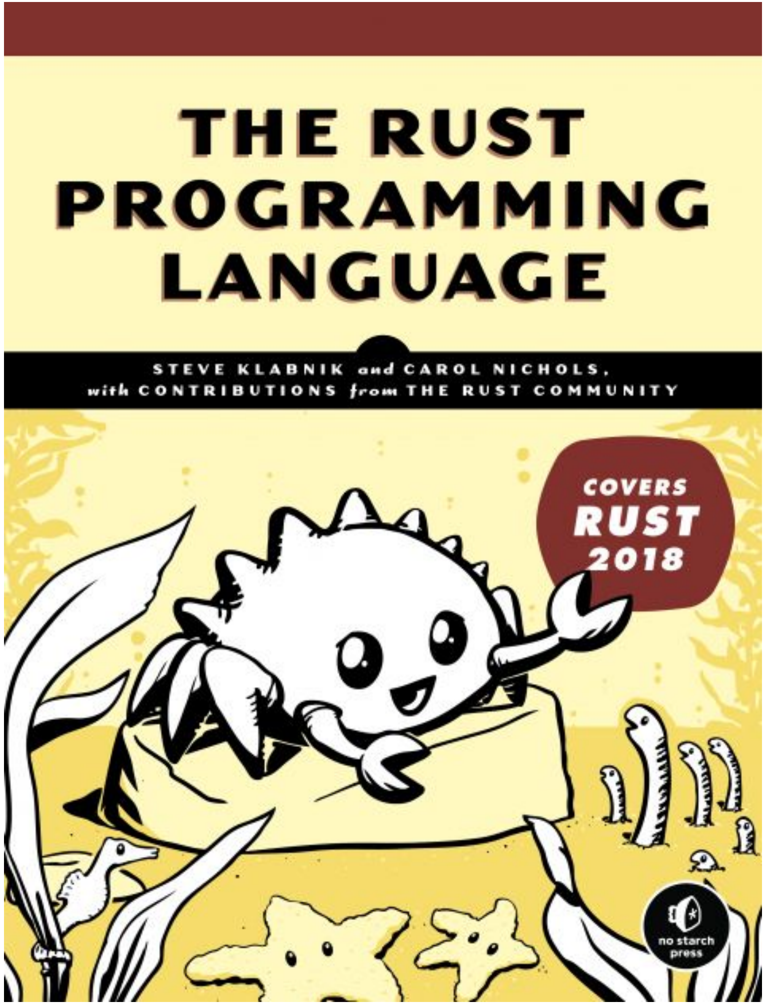


Rust in the Linux ecosystem

Miguel Ojeda

ojeda@kernel.org

Backup slides

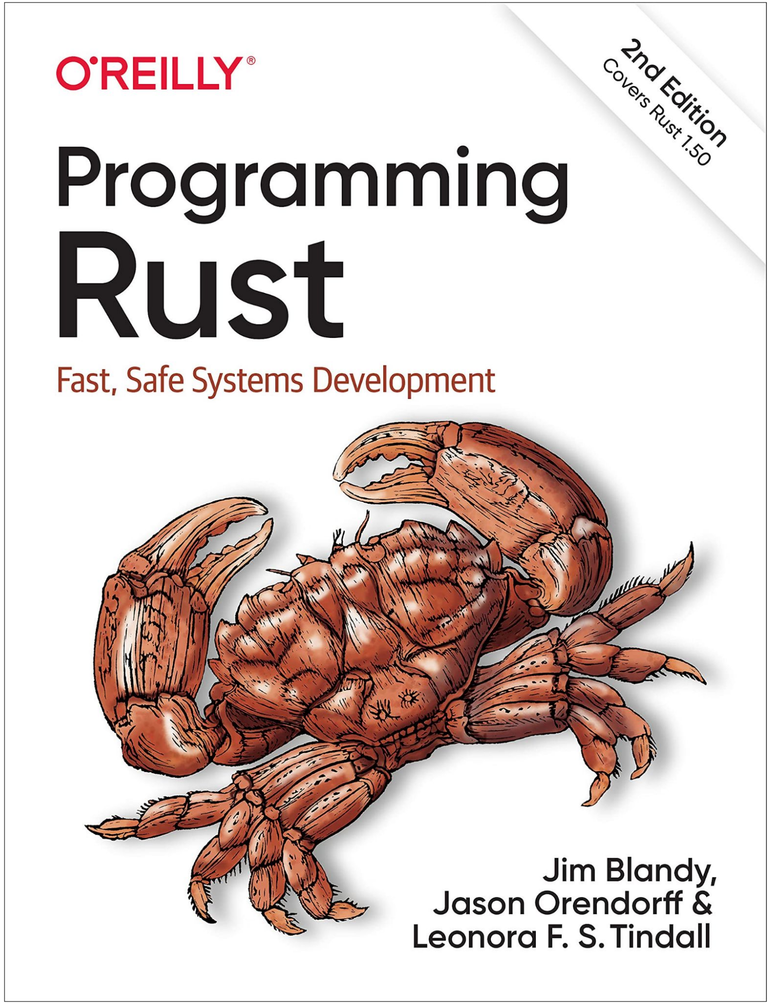


THE RUST PROGRAMMING LANGUAGE

STEVE KLABNIK and CAROL NICHOLS,
with CONTRIBUTIONS from THE RUST COMMUNITY

COVERS RUST 2018

no starch press



O'REILLY®

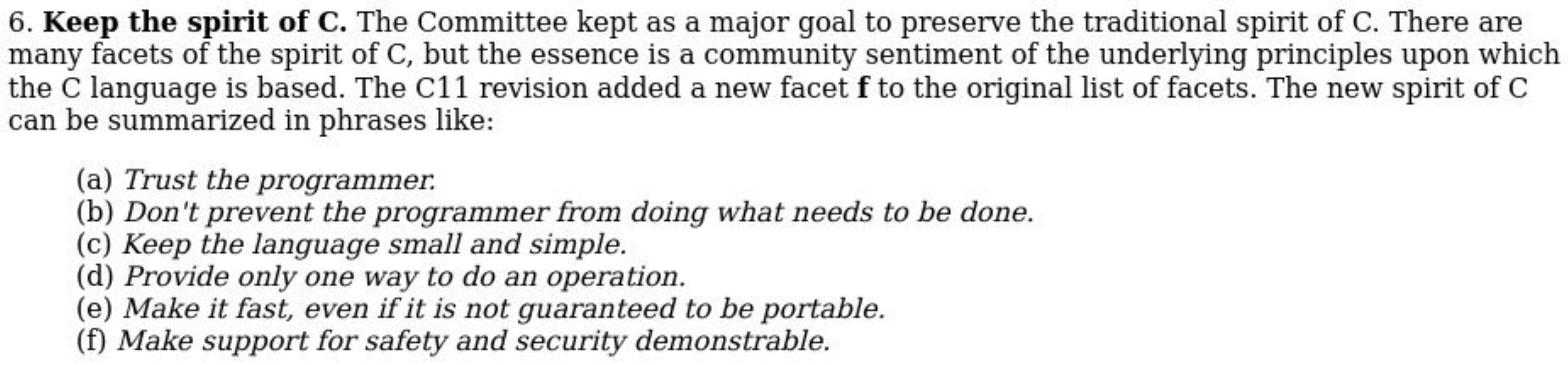
2nd Edition
Covers Rust 1.50

Programming Rust

Fast, Safe Systems Development

Jim Blandy,
Jason Orendorff &
Leonora F. S. Tindall

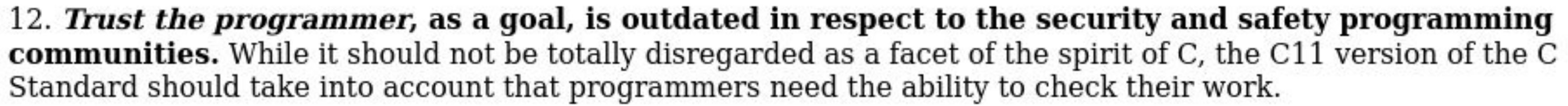
C Charter



6. **Keep the spirit of C.** The Committee kept as a major goal to preserve the traditional spirit of C. There are many facets of the spirit of C, but the essence is a community sentiment of the underlying principles upon which the C language is based. The C11 revision added a new facet **f** to the original list of facets. The new spirit of C can be summarized in phrases like:

(a) *Trust the programmer.*
(b) *Don't prevent the programmer from doing what needs to be done.*
(c) *Keep the language small and simple.*
(d) *Provide only one way to do an operation.*
(e) *Make it fast, even if it is not guaranteed to be portable.*
(f) *Make support for safety and security demonstrable.*

— N2086 C2x Charter - Original Principles



12. ***Trust the programmer*, as a goal, is outdated in respect to the security and safety programming communities.** While it should not be totally disregarded as a facet of the spirit of C, the C11 version of the C Standard should take into account that programmers need the ability to check their work.

— N2086 C2x Charter - Additional Principles for C11