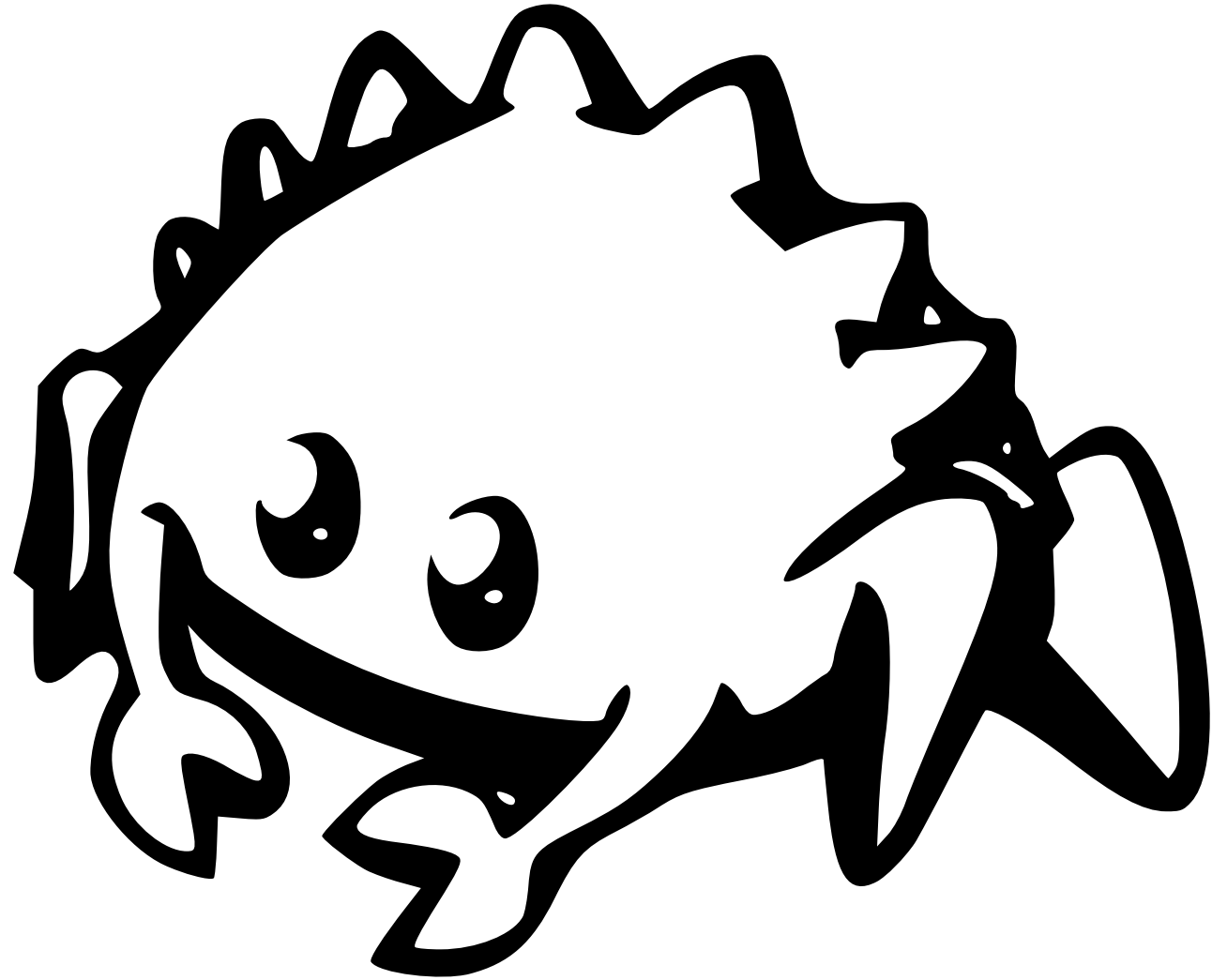


Rust: OVUNQUE

LinuxDay Torino 2023

Luca Barbato

- lu_zero@gentoo.org
- lu_zero@videolan.org



Intro

Chi sono io?

- Sviluppo Gentoo, una distribuzione Linux molto flessibile e duttile.
- Contribuisco a diversi progetti sotto l'ombrellone VideoLan
- Ultimamente contribuisco a Rust e scrivo software multimediale in questo linguaggio
 - [rav1e](#) ad esempio.
- E per non farmi mancare nulla lo uso pure in ambito IoT implementando [Web Of Things](#).

Chi siete voi?

- Chi ha sentito parlare di Rust?
- Quanti di voi programmano in Rust?

Rust in poche parole

Un linguaggio che permette a chiunque di sviluppare software affidabile ed efficiente.

rust-lang.org

Punti chiave

- È un linguaggio per **tutti**.
- Per scrivere programmi **efficienti**.
 - ed **affidabili**

Rust in altre parole

- È un linguaggio di programmazione di sistema che offre garanzie forti sul comportamento del codice scritto
 - Il compilatore evita una serie di tipi di errori **molto** comuni.
 - Scrivere codice molto **complicato** (and esempio **multi-thread** o asincrono) diventa di conseguenza più **semplice**.
- È un linguaggio in cui **ergonomia** e documentazione non sono questioni secondarie.

Rust in altre parole

- È un linguaggio che **costringe** ad essere consci di quello che si fa.
 - Il compilatore ti tiene **per mano** e spesso ti guida.
 - Quando il compilatore non può stabilire la correttezza del codice è necessario marcarlo di conseguenza con `unsafe`
- Il compilatore è in grado di ottimizzare meglio il codice prodotto come **effetto collaterale** dell'avere maggiori informazioni.

Questo l'avevo già detto nel 2019, ed apparentemente in questi 4 anni molti si sono convinti della sua bontà.

- Ora lo possiamo trovare in un sacco di posti ed usato da moltissime persone ed organizzazioni.

Ovunque?

- Rust possiamo dire che se non è assolutamente **OVUNQUE**, decisamente lo troviamo in un sacco di posti.

Pensiamo al software con cui interagiamo

- Browser
- Cellulare
- Desktop
- Server
- Tostapane

Nei browser?

I browser sono fra i software più usati

- E fra i software più complessi
 - E fra i più sfruttati per far danni
- Ha perfettamente senso evitare a priori problemi quando li si scrivere

Mozilla aveva dato il LA

"The style component is the part of a browser that applies CSS rules to a page. This is a top-down process on the DOM tree: given the parent style, the styles of children can be calculated independently: a perfect use-case for parallel computation. By 2017, Mozilla had made two previous attempts to parallelize the style system using C++. Both had failed"

[Mozilla](#)

E Google non è da meno ormai

"We are pleased to announce that moving forward, the Chromium project is going to support the use of third-party Rust libraries from C++ in Chromium. To do so, we are now actively pursuing adding a production Rust toolchain to our build system. This will enable us to include Rust code in the Chrome binary within the next year. We're starting slow and setting clear expectations on what libraries we will consider once we're ready."

[Google Security Blog](#)

E apparentemente anche la Linux Foundation pensa che sia una buona idea

The Linux Foundation, the nonprofit organization enabling mass innovation through open source, today announced it will host the Servo web engine. Servo is an open source, high-performance browser engine designed for both application and embedded use and is written in the Rust programming language, bringing lightning-fast performance and memory safety to browser internals. Industry support for this move is coming from Futurewei, Let's Encrypt, Mozilla, Samsung, and Three.js, among others.

[Linux Foundation Press Release](#)

The independent, modular, and embeddable web rendering engine written in Rust to nurture a stronger European open source ecosystem

[Linux Foundation Europe Newsroom](#)

Ovunque? Nel tuo cellulare!

Nel cellulare?

Se usi Android decisamente sì

- Alcuni componenti delle versioni recenti di Android sono scritte in rust, sono partiti da una reimplementazione dello stack bluetooth che ancora troviamo [qui](#).
- E come da tradizione han finito per inventare modi creativi per farsi male con i [build system](#).
- Ora c'è addirittura ha un mini-corso su come [usarlo su android](#)
 - ed è uno dei linguaggi supportati da [Android Studio](#).

E su iOS?

- Dato che Apple ha abbracciato Swift, non c'è (ancora) un ottimo supporto ufficiale
 - Ma terze parti rendono facile usarlo.
 - Ad esempio [tauri](#).

Ovunque? Nel tuo desktop!

Sul desktop?

Microsoft adora Rust

"As we've seen, roughly 70% of the security issues that the MSRC assigns a CVE to are memory safety issues. This means that if that software had been written in Rust, 70% of these security issues would most likely have been eliminated."

Microsoft

- Quindi ora sta [riscrivendo](#) parti di Windows in Rust.
- E contribuendo crate per scrivere sia [applicazioni](#) sia [driver](#).

Ma è il LinuxDay!

- Anche per **Linux** abbiamo desktop scritti in Rust
 - [COSMIC](#) di [Pop!_OS](#) sta venendo riscritto in Rust.
 - [smithay](#) è una buona implementazione di Wayland
 - non dissimile a [wlroots](#) come spirito.
 - [jay](#) è ormai impressionante e segue la filosofia opposta.

Terminali

- [alacritty](#) è stato fra i primi, spartano ma potente ed embeddabile.
- [rio](#) è più amichevole come UI
 - anch'esso dimostra come sia facile scrivere software gpu-accelerato.
- [wezterm](#) è ancora più completo e rimane nello stesso filone.

Command line utility

- [ripgrep](#) è un grep più veloce ed amichevole.
- [bat](#) è un cat con syntax highlighting e molto altro.
- [gitoxide](#) è una reimplementazione di Git.
- [uutils](#) fornisce una reimplementazione di [coreutils](#) e [findutils](#).
- [bottom](#) è una valida alternativa ad [htop](#).

Applicazioni Grafiche

- [lapce](#) mira ad essere un valido sostituto di Code.
- [graphite](#) è un editor grafico scritto in rust e che si può usare dal browser.
- [servo](#) è un browser web in sviluppo.
- [veloren](#) è un RPG cubettoso simile a Minecraft per certi versi.

Librerie

- Riscrivere tutto in Rust, buttando via tutto in un colpo solo non ha sempre senso
 - Spesso è meglio usare un approccio incrementale
- Ecco perchè abbiamo Rust assieme ad altri linguaggi e librerie con parti di Rust.

Librerie con parti di Rust

- [librsvg](#) è stata un apripista
- [gstreamer](#) ha molti nuovi plugin scritti in rust.
- [libcurl](#) ora ha backend per [hyper](#), [rustls](#) e [quiche](#).
- [cryptography](#) implementa alcuni cifrari in Rust.

Librerie nate in Rust (Data Science)

- [polars](#) è una valida alternativa al famoso [pandas](#).
- [ndarray](#) risolve gli stessi problemi che `numpy.ndarray` risolve in Python.
 - Con tanto di [guida](#) alla conversione.
- [evcxr](#) Permette di integrare Rust dentro [Jupyter](#).
- [poloto](#) e [plotters](#) per visualizzare i dati.

Librerie nate in Rust (Machine Learning)

- [burn](#) mira ad essere una soluzione completa per costruire allenare ed eseguire reti neurali.
 - Supporta differenti backend, da `ndarray` per girare ovunque ci sia una cpu a `wgpu` per sfruttare hardware grafico.
- [candle](#) è un approccio più minimalista simile a PyTorch.
 - Arriva con implementazioni di [whisper](#), [LLaMA2](#), [T5](#), [yolo](#), [Segment Anything](#).
 - E come molti altri crate, lo si può usare anche nel browser e in generale da altri linguaggi

Crate rust sotto mentite spoglie

- Crate che si fingono pacchetti python:
 - [polars](#), una alternativa a `pandas`
 - [deltalake](#), per accedere a data lake di `delta.io`
 - [candle](#), così migrare da PyTorch diventa ancora più indolore.
- Crate che si fingono librerie C:
 - [rav1e](#), un encoder AV1.
 - [rustls-ffi](#), una implementazione di TLS
 - [libimagequant](#), una libreria per ottimizzare PNG e GIF.
 - [pkgcraft](#), una reimplementazione di Portage in rust.

E sul server?

Se Rust è così buono vogliam lasciare solo Microsoft a giocarci?

- Da Linux 6.1 c'è un parziale supporto di Rust che via via cresce
 - E ci sono già driver notevoli scritti in Rust, ad esempio quello per la [GPU dell'M1](#).
- FreeBSD [sperimenta anche](#) lui.
- Ed esiste [Redox](#) se qualcuno vuole un intero sistema operativo scritto in Rust sin dall'inizio.

Application Server Web

- [rocket](#)
- [actix-web](#)
- [axum](#)
- [pavex](#)

Application Server RPC

- Prost
- Tonic
- Tarpc
- Cap'n Proto

Serverless / WASI

- [wasmtime](#)
- [wasmer](#)

Message Broker

- [zenoh](#) (MQTT, DDS, ...)
- [libp2p](#) (IPFS, DHT, ...)
- [lapin](#) (AMQP)

Database Engine

- SQL
 - [gluesql](#)
 - [datafusion](#)
- Key-Value
 - [sled](#)
 - [TiKV](#)

Rust embedded

- Rust grazie a `no_std` parte in vantaggio
 - l'approccio di avere la libreria standard "spegribile" permette di graduare cosa si supporta
 - `std` per avere tutta la libreria standard.
 - `core` per avere solo le componenti che non devono allocare.
 - `alloc` per avere anche le parti di std che allocano.
 - Moltissimi crate nascono per funzionare anche su bare metal.

Hardware supportato

- [expressif](#) ha abbracciato Rust per le sue architetture `ESP32`.
 - in particolare la sua versione a base di RISC-V permette di fare [moltissimo](#).
- [embassy](#) fornisce astrazioni asincrone e supporta molteplici sistemi
 - inclusi `STM32` e `nRF*`, oltre a `ESP32` in ogni sua forma o quasi.
- Esistono diverse `HAL` per molteplici architetture su cui poter scrivere il proprio sistema senza partire da zero.
- L'organizzazione [rust-embedded](#) è un buon punto di partenza per chi è curioso.

Rust nelle Automobili?

- Grazie a [Ferrous System](#) è ora possibile usare Rust in ambito automotive ed industriale
 - Han faticato molto, ma ora esistono compilatori Rust che supportano `ISO 26262 ASIL-D` and `IEC 61508 SiL4`.
 - E se quanto stan facendo assieme all'[ESA](#) porta i suoi frutti...

Rust su Marte!

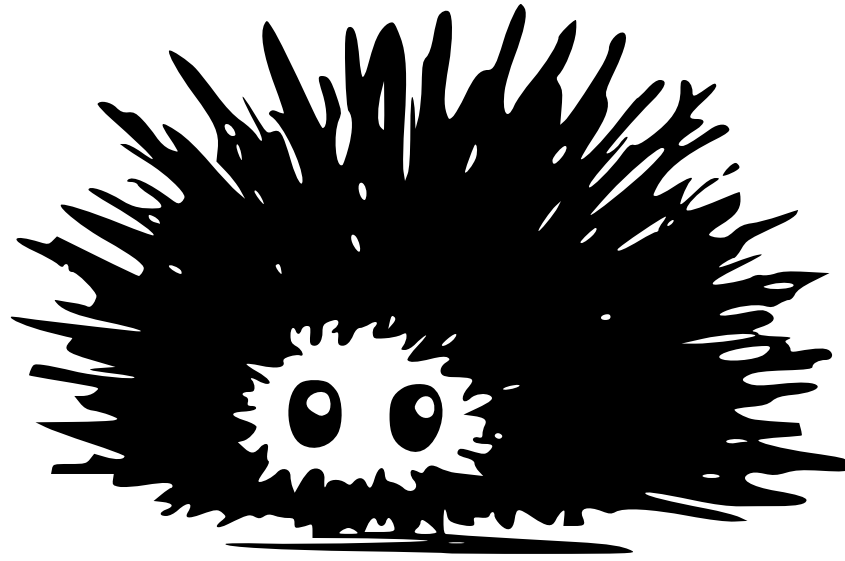
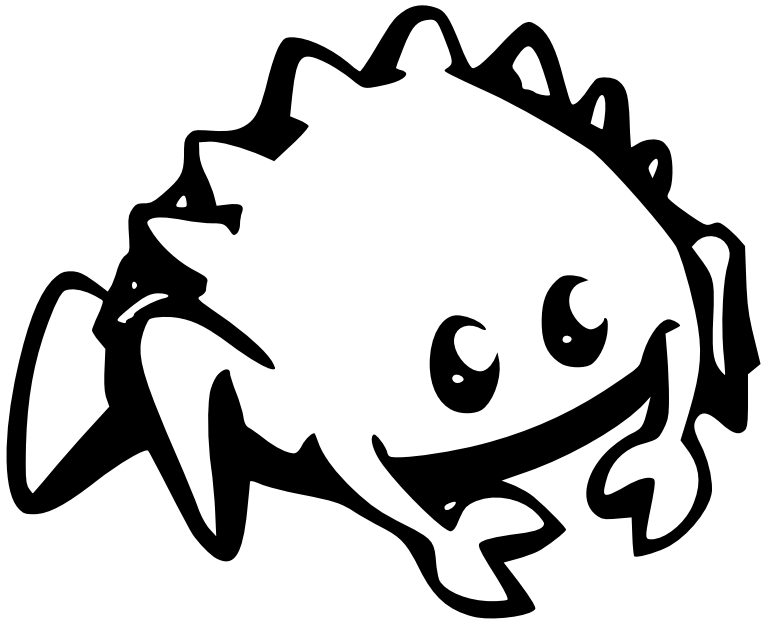
- O almeno oltre la troposfera.

Conclusione

Sono ben 4 anni che parlo di Rust al LinuxDay:

- Nel **2017** ho raccontato Rust, spiegando perchè **non** lo si dovesse usare per riscrivere il mondo.
- Nel **2018** ho illustrato come mai Rust sia stato scelto per scrivere [rav1e](#).
- Nel **2019** vi ho raccontato perchè le ragioni per non usarlo eran poche.
- **Oggi** vi ho raccontato in quanti posti potete trovarlo
 - ed in quanti altri posti lo troveremo in futuro.

Domande?



Riferimenti

- <https://github.com/rust-torino> e <https://github.com/rust-italia>
- <https://github.com/lu-zero/linuxday2023>
- <https://www.rust-lang.org/it/>
- Twitter: [@lu_zero_](https://twitter.com/lu_zero_)