RUST: UNA NUEVA ALTERNATIVA PARA LA PROGRAMACIÓN DE SISTEMAS CRÍTICOS

FLISol

Armenia - Quindío
COLOMBIA

Abril 22 - 2023

Alexander López-Parrado, PhD

Agenda

- ¿Qué son sistemas críticos?
- Seguridad de lenguaje C
- ¿Porqué Rust?
- Rust para principiantes
- Demo

FLISoL 2023
Armenia - Quindío

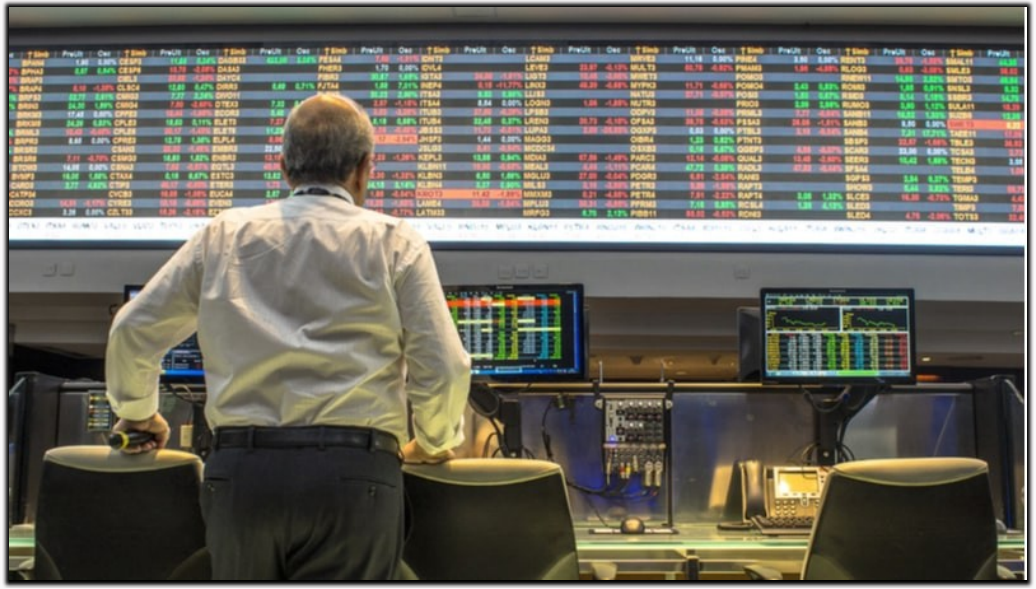
¿Qué son sistemas críticos?

Crítico a nivel de seguridad

Crítico a nivel de misión

FLISoL 2023
Armenia - Quindío

¿Qué son sistemas críticos?



Crítico a nivel de negocios

FLISoL 2023 Armenia - Quindío

¿Qué son sistemas críticos?

- 2.5 millones de líneas de código en C
- Scripts de prueba en Python

VxWorks

Rover Curiosity 2011
2500 millones USD

FLISoL 2023 Armenia - Quindío

Seguridad de lenguaje de C

- **¿Porqué C?**
  - Acceso a bajo nivel del hardware del computador.
  - Compiladores optimizados.
  - Facilita asociación de restricciones de tiempo real.
  - Código del legado de sistemas operativos en C.
  - Más de 50 años de existencia y actualización.
  - *Pero...*

FLISoL 2023 Armenia - Quindío

Seguridad de lenguaje de C

- **Problemas de lenguaje C**
  - Falta de verificación de límites de arreglos.
  - Dereferenciación de punteros nulos.
  - Funciones de la biblioteca estándar con vulnerabilidades.
  - Errores simples pueden desencadenar grandes vulnerabilidades de memoria explotables.

FLISoL 2023 Armenia - Quindío

```
SIGSEGV invalid memory access (segmentation fault)
SIGABRT   abnormal termination condition
```

Seguridad de lenguaje de C

EJEMPLOS

FLISoL 2023 Armenia - Quindío

Seguridad de lenguaje de C

WebKit entry point

WebKit is the open source layout engine which renders web pages in the browsers for iOS, Wii U, 3DS, PS Vita, and the PS4.

Although so widely used and mature, WebKit does have its share of vulnerabilities; you can learn about many of them by reading Pwn2Own write-ups.

In particular, the browser in PS4 firmware 1.76 uses a version of WebKit which is vulnerable to CVE-2012-3748, a heap-based buffer overflow in the `JSArray::sort(...)` method.

In 2014 nas and Proxima announced that they had successfully been able to port an exploit using this vulnerability, originally written for Mac OS X Safari, to the PS4's internet browser, and released the PoC code publicly as the first entry point into hacking the PS4.

This gives us arbitrary read and write access to everything the WebKit process can read and write to, which can be used to dump modules, and overwrite return addresses on the stack, letting us control the instruction pointer register (`rip`) to achieve ROP execution.

Since then, many other vulnerabilities have been found in WebKit, which could probably be used as an entry point for later firmwares of the PS4, but as of writing, no one has ported any of these exploits to the PS4 publicly.

If you have never signed into PSN, your PS4 won't be able to open the Internet Browser, however you can go to "Settings", and then "User's Guide" to open a limited web browser view which you can control the contents of with a proxy.

FLISoL 2023 Armenia - Quindío

https://cturt.github.io/ps4.html

¿Porqué Rust?

Noviembre del año 2022

FLISoL 2023 Armenia - Quindío

National Security Agency/Central Security Service

About Press Room Careers History

NSA Releases Guidance on How to Protect Against Software Memory Safety Issues

FORT MEADE, Md. — The National Security Agency (NSA) published guidance today to help software developers and operators prevent and mitigate software memory safety issues, which account for a large portion of exploitable vulnerabilities.

The "Software Memory Safety" Cybersecurity Information Sheet highlights how malicious cyber actors can exploit poor memory management issues to access sensitive information, promulgate unauthorized code execution, and cause other negative impacts.

"Memory management issues have been exploited for decades and are still entirely too common today," said Neal Ziring, Cybersecurity Technical Director. "We have to consistently use memory safe languages and other protections when developing software to eliminate these weaknesses from malicious cyber actors."

https://media.defense.gov/2022/Nov/10/2003112742/-1/-1/0/CSI_SOFTWARE_MEMORY_SAFETY.PDF

¿Porqué Rust?

- **C/C++ son flexibles en el manejo de la memoria**
  - Delegan la responsabilidad al programador del manejo de las referencias. *Se puede lograr con buenas prácticas*.
- **Se recomienda hacer uso de lenguajes de programación seguros a nivel de memoria en la medida de lo posible.**
- **La NSA menciona específicamente lenguajes como**
  - C#, Go, Java®, Ruby™, **Rust®**, y Swift®

FLISoL 2023 Armenia - Quindío

¿Porqué Rust?

- C# → Compilado para lenguaje intermedio
- Go → Completamente compilado
- Java® → Compilado para máquina virtual
- Ruby™ → Interpretado
- **Rust®** → Completamente compilado
- Swift® → Completamente compilado

FLISoL 2023 Armenia - Quindío

¿Porqué Rust?

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

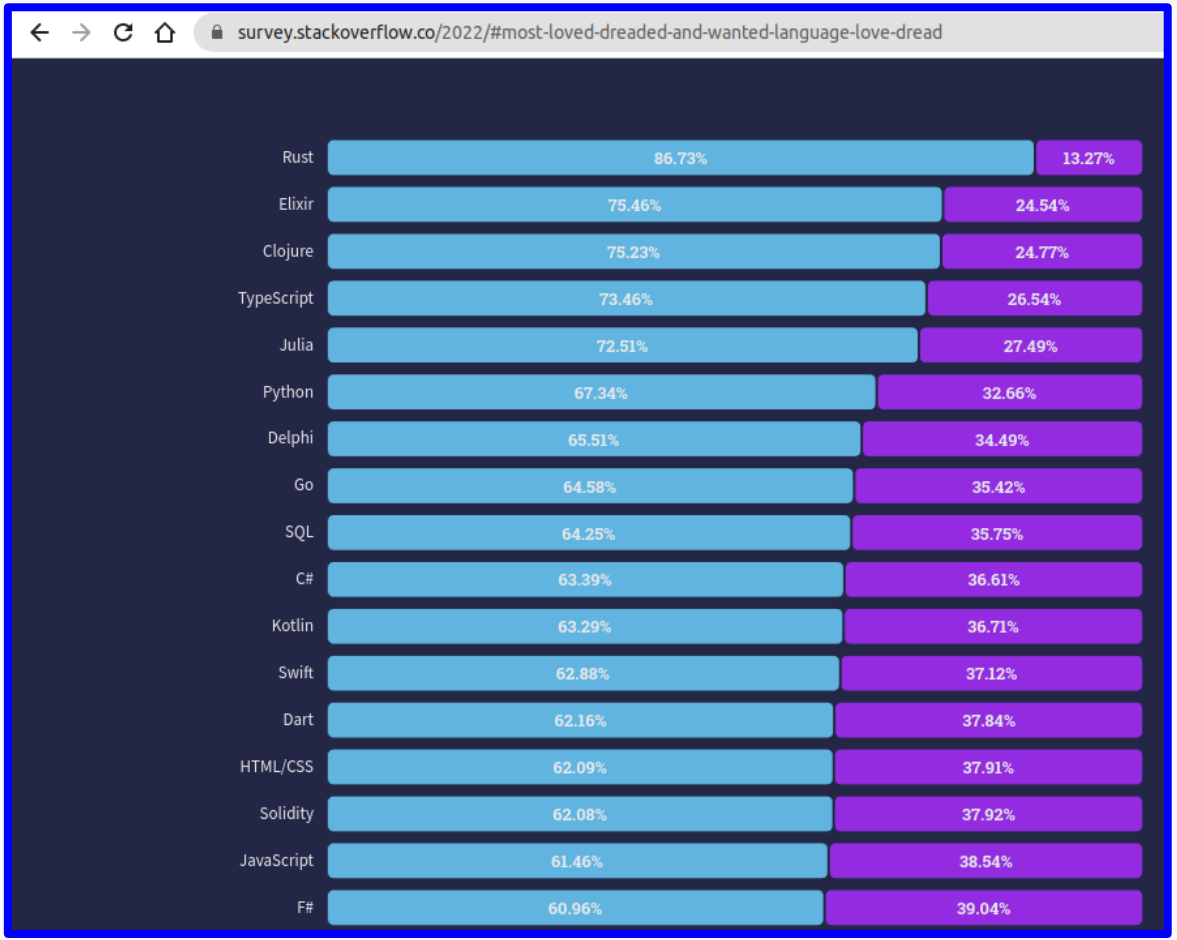
FLISoL 2023 Armenia - Quindío

https://greenlab.di.uminho.pt/wp-content/uploads/2017/09/paperSLE.pdf

¿Porqué Rust?

stackoverflow

"Rust is on its seventh year as the most loved language with 87% of developers saying they want to continue using it."

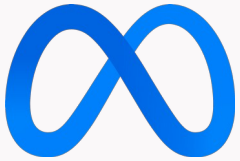


FLISoL 2023
Armenia - Quindío

¿Porqué Rust?

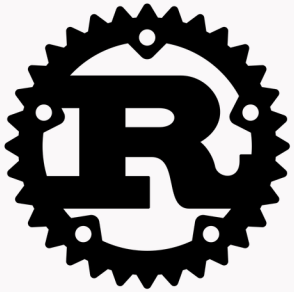
- Desde diciembre de 2022 el kernel de Linux soporta Rust
  - Segundo lenguaje soportado después de C desde la creación de Linux en 1991
- Diversas compañías han adoptado Rust

FLISoL 2023 Armenia - Quindío



Rust para principiantes

Rust es un lenguaje de programación multiparadigma de tipos estáticos de bajo nivel que se centra en la seguridad y el rendimiento.



Rust comenzó como un pequeño proyecto paralelo en Mozilla. La empresa vio su potencial como sucesor de C/C++ y comenzó a patrocinar el nuevo lenguaje de programación.

FLISoL 2023 Armenia - Quindío

Lanzado en el año 2015

Rust para principiantes

```rust
fn main() {
    println!("Hello World!");
}
```

FLISoL 2023
Armenia - Quindío

Rust para principiantes

Length	Signed	Unsigned
8-bit	`i8`	`u8`
16-bit	`i16`	`u16`
32-bit	`i32`	`u32`
64-bit	`i64`	`u64`
128-bit	`i128`	`u128`
arch	`isize`	`usize`

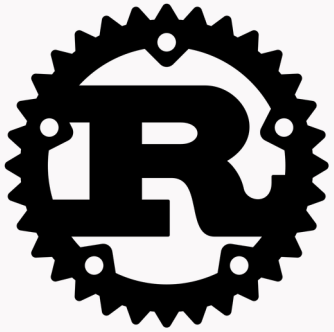
Number literals	Example
Decimal	`98_222`
Hex	`0xff`
Octal	`0o77`
Binary	`0b1111_0000`
Byte (`u8` only)	`b'A'`

```rust
fn main() {
    let x = 2.0; // f64

    let y: f32 = 3.0; // f32
}
```

FLISoL 2023
Armenia - Quindío

Rust para principiantes



- Ejemplos
  - Tipos de datos
  - Seguridad en acceso a memoria
  - Control de flujo
  - Funciones

FLISoL 2023
Armenia - Quindío

Rust para principiantes

- Aspectos básicos de manejo de errores
  - Rust incluye un mecanismo basado en enumeraciones para el manejo de los errores de las funciones

```
pub enum Result<T, E> {
    Ok(T),
    Err(E),
}
```

`Ok(T)` → **Contiene el valor cuando no hay error**

`Err(E)` → **Contiene el valor de error**

FLISoL 2023 Armenia - Quindío

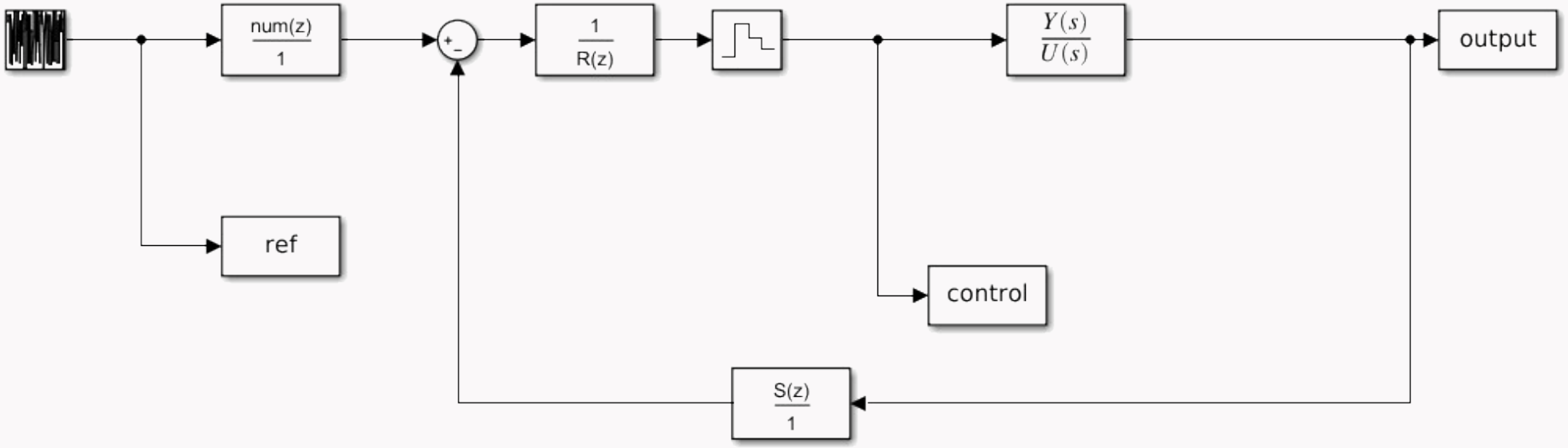
Rust para principiantes

- Software usado
  - cargo y rustc que pueden ser instalados mediante rustp (Linux y Windows)
  - Visual Studio Code junto con las extensiones: rust-analyzer y CodeLLDB. En Windows usar la extensión Microsoft C++.

https://forge.rust-lang.org/infra/other-installation-methods.html

FLISoL 2023 Armenia - Quindío

Demo



FLISoL 2023 Armenia - Quindío

Control RST de un doble integrador

GRACIAS!

parrado@uniquindio.edu.co

FLISol

Armenia - Quindío
COLOMBIA

Abril 22 - 2023