The Rust Programming Language

Pramode C.E

`http://pramode.net`

January 16, 2018

10:10 AM
Google
Media Hub
TeleNav GPS
Web
Email
Phone
Contacts
Messaging
Applications

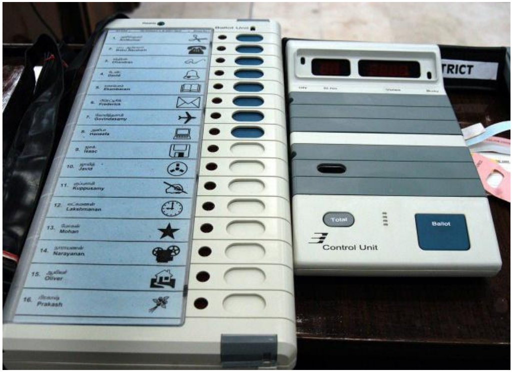
BOEING

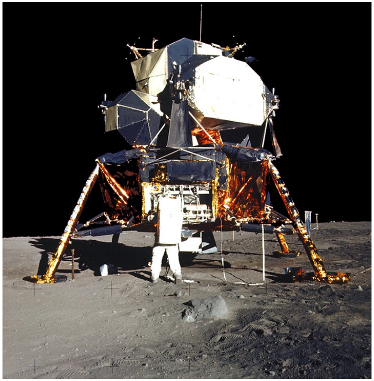
ACCOLADE EL
Boston Scientific
ACCOLADE MRI
Boston Scientific
ESSENTIO EL
Boston Scientific
ESSENTIO MRI
Boston Scientific

ilumi
ilumi
ilumi

www.jklossner.com
©14 c'world klossner
"WE HAVE TO GO OUT FOR DINNER. THE REFRIGERATOR ISN'T SPEAKING TO THE STOVE."

Audi A3


Ballot Unit
Total
Ballot
Control Unit



What is common to all these?

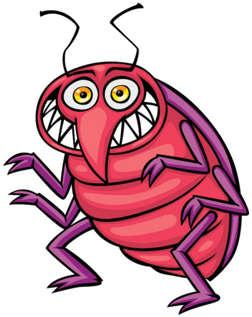
0 1 1 0 1 1 0 1 1 0 1 1 0 0 0 1 1 0 ...

A lot of code ...

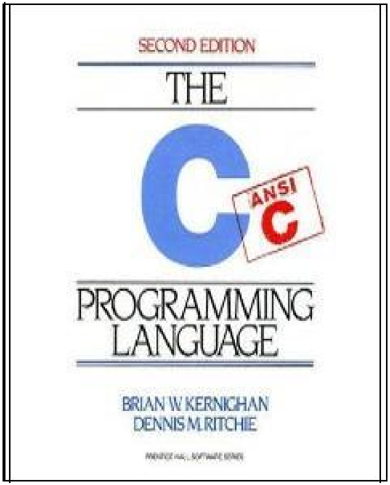
- Cardiac Pacemaker: 80,000 lines of code
- MRI Scanner: 70 lakh lines
- Boeing 787 dreamliner: 70 lakh lines
- Ford GT(Car): 100 lakh lines
- Light bulb: ??
- Door lock: ??
- Oven: ??

Ref: http://www.informationisbeautiful.net/visualizations/million-lines-of-code/

With code comes bugs ...



Code bugs can kill ... and create havoc!


SECOND EDITION
THE
C
ANSI C
PROGRAMMING LANGUAGE
BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

What comes to your mind when you think of C?

SPEED

Bộ-Viện
Võ-Trận Đại-Việt

C Evil Spirits

- Buffer overflow / off-by-one errors
- Null dereferencing
- Use-after-free
- Memory leaks
- Dangling Pointers
- Undefined Behaviours

C Evil Spirits

```
int main()
{
    int a[3];
    a[0] = 1; a[3] = 10;

    return 0;
}
```

C Evil Spirits

```
int main()
{
    char a[5];
    strcpy(a, "hello");

    return 0;
}
```

C Evil Spirits

"The second big thing each student should learn is: How can I avoid being burned by Cs numerous and severe shortcomings? This is a development environment where only the paranoid can survive" John Regehr — http://blog.regehr.org/archives/1393

OUR DEVICES ARE NOW 100% SECURE.
HOW DID YOU DO THAT?
I TURNED THEM ALL OFF.
© D.Fletcher for CloudTweaks.com

Bugs!

Bugs!

Morris Worm: 1988 https://en.wikipedia.org/wiki/Morris_worm

Bugs!

How to smash the stack for fun and profit: 1996
http://phrack.org/issues/49/14.html#article

The most recent MacOS release fixed 32 vulnerabilities caused due to memory safety (https://support.apple.com/en-us/HT208221).

The Most recent Google Chrome release fixed 10 memory safety issues (https://chromereleases.googleblog.com/2017/10/stable-channel-update-for-desktop.html) and Firefox fixed 38 (https://www.mozilla.org/en-US/security/advisories/mfsa2017-24/).

Google's OSS-Fuzz and Project Zero have uncovered a large number of memory safety related vulnerabilities.

The Promise of Rust

- Guaranteed Memory SAFETY without using Garbage Collection
- As fast as C/C++
- Can write OS kernels, embedded systems code
- High level abstractions like algebraic data types / pattern matching with very little performance overhead
- Minimal run-time
- Easy interfacing with C
- Threads without data races

Ref: https://www.rust-lang.org/

The Promise of Rust

- Graydon Hoare started working on Rust in 2006
- Mozilla became interested - idea was to use it in the experimental Servo browser engine project
- Rust becomes an official Mozilla project
- Several major design changes before hitting v1.0 in May 2015. Language stabilizes
- Regular six week release cycle. Stable and Nightly "channels"

Installation

Visit: https://rustup.rs/

Documentation

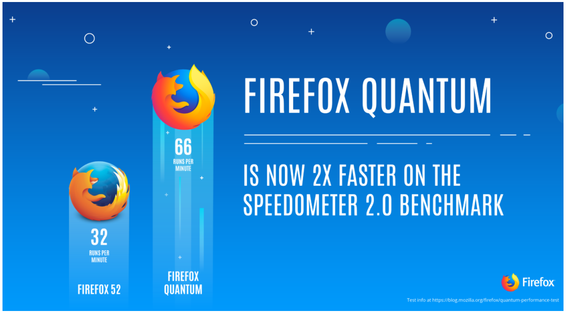
- Official "book" (2nd edition): https://doc.rust-lang.org/book/second-edition/
- Rust By Example: http://rustbyexample.com/
- O'Reilly book: http://shop.oreilly.com/product/0636920040385.do

Who is using it?

Remember, Rust is just a 2 year old baby!

- Organizations running Rust in Production: https://www.rust-lang.org/en-US/friends.html
- Mozilla, for the Servo project.
- Dropbox story: http://www.wired.com/2016/03/epic-story-dropboxs-exodus-amazon-cloud-empire/
- Coursera is using Rust: https://building.coursera.org/blog/2016/07/07/rust-docker-in-production-coursera/

Who is using it?



Who is using it?

Interesting Projects

- Servo: the secure, parallel browser engine from Mozilla:https://servo.org/
- Redox OS: Unix-like microkernel OS written in Rust: https://www.redox-os.org/
- Tock OS: an embedded OS for microcontrollers written in Rust: http://www.tockos.org/about/

Rust, the most loved programming language (stackoverflow devloper survey, 2016):
http://stackoverflow.com/research/developer-survey-2016

Some limitations

- Complex language.
- Library ecosystem not yet mature.
- Use of "unsafe" under the hood.
- Slow compilation.

About this workshop

This workshop is divided into 3 parts:

- Simple
- Cool
- Hard (and Cool!)

The Simple parts

- We will quickly look at: variables, control structures, function basics, basic types etc.

The Cool parts

- Algebraic Data types
- Pattern Matching
- Generics and Traits
- Iterators
- Closures
- Cargo

and some other things ...

The Hard(and Cool) Parts

This is the *core* of Rust - where the real innovation comes in.

- Systems programming basics - understanding buffer overflows, use-after-free, memory leaks, dangling pointers etc.
- Ownership and move semantics
- Borrowing
- Lifetimes
- Unsafe Rust

Let's dive into code!