

1 Παρουσίαση της γλώσσας προγραμματισμού Rust

1.1 Ιστορία

- Ιστορική αναδρομή σε γλώσσες προγραμματισμού
- Bugs σε C,C++ που δεν μπορούν να αποφευχθούν εύκολα¹ ή χρειάζονται ιδιαίτερα προσεκτική προσπάθεια από τον χρήστη²

1.2 Κύρια πλεονεκτήματα

- Παρέχει memory safety και type safety. Δεν έχει προβλήματα με data races (δυσκολεύουν την δημιουργία προγραμμάτων που χρησιμοποιούν πολλά threads).
- Εύκολος παραλληλισμός (concurrency)
- Μεγαλύτερη ταχύτητα και απόδοση σε σχέση με τις περισσότερες γλώσσες προγραμματισμού
- Εύκολη κλήση της Rust από C/C++ και το αντίθετο (κλήση της C/C++ από Rust)^{3 4}

1.3 Άλλα χαρακτηριστικά

- Χαρακτηριστικά από συναρτησιακές γλώσσες
- Package Manager - Βιβλιοθήκες (Cargo, crates.io)
- Έλεγχοι για σφάλματα που γίνονται από τον compiler (αποφεύγονται προβλήματα στη συνέχεια)

¹Dietz W., Li P., Regehr J. & Adve V. (2012). Understanding integer overflow in C/C++. In *34th International Conference on Software Engineering*. Zurich, Switzerland: IEEE Press, 760–770.

²Akritidis P. (2011). *Practical memory safety for C (Technical Report Number 798)*. University of Cambridge.

³The Rust FFI Omnibus (<http://jakegoulding.com/rust-ffi-omnibus/>)

⁴Calling C from Rust (<https://jvns.ca/blog/2016/01/18/calling-c-from-rust/>)

- Standard library (μικρό μέγεθος, μπορεί να χρησιμοποιηθεί σε ενσωματωμένα συστήματα κτλ.)⁵

1.4 Οικοσύστημα

- Αξιόλογα software (Servo^{6 7 8}, Redox, rustc, ripgrep⁹, font-rs¹⁰)
- Ενσωμάτωση της γλώσσας (IDEs, plugins)
- Documentation (βιβλία, τεχνικά έγγραφα κτλ.)
- Εμπορική υποστήριξη, εταιρίες (Mozilla, Dropbox^{11 12}, Facebook?¹³) και software που χρησιμοποιούν τη γλώσσα (Firefox, librsvg¹⁴, Tor?¹⁵)
- Κοινότητα, open source ανάπτυξη της γλώσσας^{16 17}

⁵Fearless concurrency in your microcontroller (<http://blog.japaric.io/fearless-concurrency/>)

⁶Anderson B., Bergstrom L., Herman D., Matthews J., McAllister K., Goregaokar M., Moffitt J. & Sapin S. (2015). *Experience report: Developing the Servo web browser engine using Rust*. Self-Published.

⁷Anderson B., Bergstrom L., Goregaokar M., Matthews J., McAllister K., Moffitt J. & Sapin S. (2016). Engineering the servo web browser engine using Rust. In *34th International Conference on Software Engineering*. Austin, Texas: IEEE Press, 81-89.

⁸Zambre R., Bergstrom L., Beni L. A. & Chandramowlishwaran A. (2016). *Parallel Performance-Energy Predictive Modeling of Browsers: Case Study of Servo*. Self-Published.

⁹"ripgrep is faster than grep, ag, git grep, ucg, pt, sift" (<http://blog.burntsushi.net/ripgrep/>)

¹⁰"Inside the fastest font renderer in the world" (<https://medium.com/@raphlinus/inside-the-fastest-font-renderer-in-the-world-75ae5270c445>)

¹¹"The Epic Story of Dropbox's Exodus From the Amazon Cloud Empire" (<https://www.wired.com/2016/03/epic-story-dropboxs-exodus-amazon-cloud-empire/>)

¹²"Lossless compression with Brotli in Rust for a bit of Pied Piper on the backend" (<https://blogs.dropbox.com/tech/2016/06/lossless-compression-with-brotli/>)

¹³"Facebook is writing a Mercurial server in Rust." (<https://groups.google.com/forum/#!topic/mozilla.dev.version-control/nh4fITfEMk>)

¹⁴"Librsvg 2.41.0 is released" (<https://mail.gnome.org/archives/desktop-devel-list/2017-January/msg00001.html>)

¹⁵"Tor in a safer language: Network team update from Amsterdam" (<https://lists.torproject.org/pipermail/tor-dev/2017-March/012088.html>)

¹⁶Rust on GitHub: <https://github.com/rust-lang/rust>

¹⁷Rust on Reddit: <https://www.reddit.com/r/rust/>

1.5 Μειονεκτήματα

- Τομείς στους οποίους η γλώσσα δεν είναι ώριμη ακόμα (πχ. γραφικά, ενσωμάτωση με γνωστά IDEs)
 - Η Rust υποστηρίζει προς το παρόν μόνο τις πιο δημοφιλείς αρχιτεκτονικές και αυτό έχει δημιουργήσει προβλήματα σε κάποιες διανομές Linux^{18 19}
- Δυσκολία εκμάθησης, απαιτητική γλώσσα
- Προβλήματα κατανόησης των θετικών στοιχείων της γλώσσας· πολλοί χρήστες δεν καταλαβαίνουν σε ποιες περιπτώσεις χρησιμεύει και φαίνεται να την αντιμετωπίζουν σαν κάτι “μαγικό” που θα λύσει όλα τα προβλήματα τους
 - RIIR (ένα μικρό “τοξικό” μέρος της κοινότητας)²⁰

1.6 Θέματα προς ανάλυση

- Μελλοντική πορεία
 - Πιθανό μερίδιο αγοράς^{21 22}
 - Χρήση σε συγκεκριμένους τομείς (από ενσωματωμένα συστήματα²³ κτλ. έως web development²⁴)
- Στοιχεία προγραμματισμού σε Rust
 - Συντακτικό και σύγκριση με άλλες γλώσσες
 - Υλοποίηση απλών προγραμμάτων-παραδειγμάτων σε Rust

¹⁸https://bugzilla.mozilla.org/show_bug.cgi?id=1284816#c26

¹⁹https://bugzilla.mozilla.org/show_bug.cgi?id=1284816#c45

²⁰<https://transitiontech.ca/random/RIIR>

²¹Meyerovich L. A. & Rabkin A. S. (2013). Empirical analysis of programming language adoption. *ACM SIGPLAN Notices*, 48(10), 1-18.

²²Rabai L. B. A., Cohen B. & Mili A. (2015). Programming Language Use in US Academia and Industry. *Informatics in Education*, 14(2), 143-160.

²³Uzlu T. & Saykol E. Utilizing Rust Programming Language for EFI-Based Bootloader Design.

²⁴Are we web yet? (<http://www.arewewebyet.org/>)

2 Άλλες αναφορές / πηγές

- Klabnik St. & Nichols C. (2017). *The Rust Programming Language*, No Starch Press. San Francisco, California.
- Blandy J. & Orendorff J. (2016). *Programming Rust*, O'Reilly Media. Sebastopol, California.
- Rust by Example (<http://rustbyexample.com/>)
- Beingessner A. (2015). *You can't spell trust without Rust*. Carleton University.
- Reed E. (2015). *Patina: A formalization of the Rust programming language*.
- Allberg F., Lindström A. D., Hörnquist P., Lindqvist J. & Marklund E. (2016). *A cursory overview of the Rust programming language*. Self-published.