

УДК 004.42:004.43

Неласа Г.В.¹, Кацюба В.В.², Кацюба М.В.²

¹ канд. техн. наук, доц. НУ «Запорізька політехніка»

² студ. гр. РТ-710 НУ «Запорізька політехніка»

ОСОБЛИВОСТІ ВИКОРИСТАННЯ МОВИ ПРОГРАМУВАННЯ RUST

Мови програмування C/C++ завдяки своїм унікальним властивостям наразі є потужним інструментом для системного програмування і програмування криптоперетворень, оскільки дають можливість достатньо низькорівневого програмування та отримання прийнятної швидкодії критичних ділянок коду.

Вони є компільованими, що дозволяє працювати в умовах «обмежених ресурсів», коли розмір та швидкість виконання мають значення. Проте, ці мови програмування мають й недоліки.

Виникнення невизначеної поведінки (undefined behavior або UB) – поведінки, яка є наслідком використання переносної або некоректної програмної конструкції або некоректних даних, для якої в цьому Міжнародному стандарті немає жодних вимог [1]. Причиною такої поведінки може бути: переповнення змінної, відсутність перевірки покажчиків на NULL або їх некоректна перевірка, звернення до невизначеного місця в пам'яті тощо. Результат UB залежить від компілятора, оскільки в ньому не передбачено місця визначення «проблемних місць» в коді програми і, відповідно, попередження про їх наявність. І тому для уникнення UB програміст повинен самостійно передбачити виникнення неоднозначностей в коді програми.

Ручне управління пам'яттю. У C та C++ не має систем автоматичного очищення пам'яті, що, з одного боку, зменшує розмір і час виконання

програми, а з іншого боку, зобов'язує програміста самостійно слідкувати за пам'яттю. Це є недоліком, оскільки це може стати причиною чисельних помилок, витоків пам'яті та інших вразливостей.

При розробці криптографічних бібліотек це дає не нульову вірогідність виконання атаки на реалізацію навіть при використанні стійких криптографічних примітивів.

Через ці недоліки постала потреба у розробці інструменту, який би забезпечував типобезпеку (type safe), тобто вбудовані перевірки, що гарантують коректність програми, але, в той самий час, дозволяв розробляти ефективні за швидкістю та розміром програми. Таким інструментом стала досить нова мова програмування Rust. Метою роботи є дослідження її особливостей.

Rust [2] – це мультипарадигмна компільована мова загального призначення, що поєднує парадигми функціонального та процедурного програмування з об'єктною системою, що оснований на типажах. Подібно до C і C++, Rust надає засоби точного контролю за використанням пам'яті і підтримує близьку відповідність між примітивними операціями мови та машинними командами, що дозволяє заздалегідь оцінити швидкодію написаного коду [1].

Ключем до виконання обіцяної надійності й швидкодії є новаторська система володіння, передачі та запозичення, що перевіряються на етапі компіляції, ретельно спроектована так, що доповнює гнучку систему статичної типізації в Rust.

Система володіння встановлює час життя кожного значення, що робить непотрібним складання сміття в ядрі мови та забезпечує надійні, але разом з тим гнучкі інтерфейси для управління такими ресурсами, як сокети та дескриптори файлів.

Передача (move) дозволяє передати значення від одного власника іншому, а запозичення (borrowing) – використовувати значення тимчасово, не змінюючи власника [1].

Наведемо приклади сучасних проектів на мові програмування Rust: Servo – движок для браузера Firefox; елементи кодової бази операційної системи Fuchsia від Google; криптовалюти: ZEN[3], Parity, Exonum; TiKV – розподілена база даних [4].

Також на Rust надає можливість портування свого коду в веб-асемблер (wasm) для реалізації веб-застосунків [5].

Мова програмування Rust є гарним наступником C/C++ для розробки сучасних криптографічних систем, оскільки написані на ній програми мають високу швидкість виконання, та при цьому забезпечують високу надійність при виконанні програмного коду без додаткових зусиль.

На цей час мова програмування Rust вже має досить велике ком'юніті прихильників та велику кількість криптографічних проєктів різного масштабу, що дозволяє прогнозувати її перспективність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бленді Д. Программирование на языке Rust / Д. Бленді, Д. Орендорф. – Москва: ДМКПресс, 2018. – 550 с.
2. Rust (язык программирования) [Електронний ресурс] – Режим доступу: [https://ru.wikipedia.org/wiki/Rust_\(язык_программирования\)](https://ru.wikipedia.org/wiki/Rust_(язык_программирования)).
3. Horizen - Zen Blockchain Foundation [Електронний ресурс] – Режим доступу: <https://github.com/HorizenOfficial>.
4. Кладов А. А. Введение (Программирование на Rust) [Електронний ресурс] – Режим доступу: https://www.youtube.com/watch?v=Oy_VYovfWyo.
5. wasm-tetris [Електронний ресурс] – Режим доступу: <https://github.com/liona24/wasm-tetris>.