

ОСОБЛИВОСТІ МОВИ ПРОГРАМУВАННЯ RUST

За останні роки мова програмування Rust здобула значну популярність серед розробників програмного забезпечення. Rust була розроблена з метою боротьби з типовими та потенційно небезпечними вразливостями безпеки пам'яті [1, 2]. Розглянемо переваги використання мови Rust для підвищення надійності та безпеки програмного забезпечення.

Досить складною темою для програмістів C++ є невизначена поведінка (Undefined behavior). Навіть досвідчені розробники найчастіше неспроможні чітко сформулювати причини його виникнення. На рис.1 показано однаковий код, але результат буде різний при чому результат може змінюватися в залежності від компілятора, оптимізації тощо.

<pre>int val1 = INT_MAX; int val2 = val1 + 1; bool x = val1 > val2; cout << x << endl;</pre>		<pre>int val1 = INT_MAX; bool x = val1 > (val1 + 1); cout << x << endl;</pre>
---	--	--

Рис. 1. Порівняння коду на мові C++ з невизначеною поведінкою

Одна з найважливіших переваг Rust – це безпека пам'яті. Rust має унікальну модель безпеки, яка обіцяє безпеку пам'яті та конкурентну безпеку, забезпечуючи при цьому продуктивність рівня C/C++. На рис.2 показано код C++ (зліва) та Rust (справа) при роботі з покажчиками.

<pre>vector<int> v = {1, 2, 3}; int& v0 =v[0]; v.push_back(4); cout << v0 << endl;</pre>		<pre>let muty=vec![1, 2, 3]; letv0=&y[0]; y.push(4); println!("{}",v0);</pre>
--	--	---

Рис. 2. Порівняння кода при роботі з покажчиками

Rust забезпечує безпеку пам'яті через концепції «володіння» та «запозичення». Концепція «володіння» передбачає, що кожна частина пам'яті має одного власника, що запобігає помилкам використання після звільнення. Концепція «запозичення» дозволяє декільком фрагментам коду отримувати доступ до пам'яті одночасно, але лише одному з доступом до зміни даних, запобігаючи «гонці» даних, коли декілька потоків або процесів одночасно намагаються змінити одні й ті ж самі дані. Rust також забороняє нульові покажчики, автоматично змінює розмір структур даних, включає перевірки переповнення для цілих чисел і багато іншого.

Однією з переваг Rust є наявність високоякісної документації та інструментів, які роблять мову доступною для вивчення та дозволяють розробникам створювати програмне забезпечення високої якості. Cargo – це інструмент управління пакетами та збіркою проектів. Він автоматизує завдання, такі як завантаження залежностей, компіляція та виконання тестів. Cargo також допомагає управляти версіями пакетів і спрощує процес створення та розповсюдження програм у Rust.

Rust широко використовується в різноманітних програмних проектах, включаючи ядра операційних систем, такі як Windows та Linux [3, 4]. Він також застосовується у безпекових рішеннях, наприклад, системах захисту від DDoS атак. Крім того, Rust використовується у сфері розробки ігор та графічних програм.

Завдяки тому, що Rust забезпечує надійність, безпеку та продуктивність, він набирає оберти у програмуванні вбудованих систем (embedded systems). Вбудовані системи – це спеціалізовані обчислювальні системи, які є частиною більших систем і виконують певні функції у режимі часу. Ці системи вбудовуються в інші пристрої, такі як мікроконтролери, і використовуються в різних сферах: автомобільні системи, пристрої Інтернету речей (IoT), промислові системи керування та інші.

Підсумовуючи можна відмітити, що Rust є відносно новою мовою програмування, але вона швидко набуває популярності. Це потужна мова програмування, яка забезпечує надійність, безпеку та продуктивність. Використовується майже у всіх напрямках програмування. Rust має потенціал, щоб стати однією з найважливіших мов програмування у майбутньому.

Список використаних джерел

1. Nichols C., Klabnik S. Rust Programming Language. No Starch Press. 2019. 560 p.
 2. Learn rust. Rust Programming Language. URL: <https://www.rust-lang.org/learn> (date of access: 11.11.2023).
 3. Gatlan S. New Windows 11 build ships with more Rust-based Kernel features. Bleeping Computer. URL: <https://www.bleepingcomputer.com/news/microsoft/new-windows-11-build-ships-with-more-rust-based-kernel-features/> (date of access: 11.11.2023).
- Rust in the linux kernel. The New Stack. URL: <https://thenewstack.io/rust-in-the-linux-kernel/> (date of access: 11.11.2023)