

Лабораторна робота 5.

Крок 1: Підключення бібліотек

Для початку створимо новий проект Rust та додамо необхідні бібліотеки. Для багатопоточності ми використовуватимемо стандартну бібліотеку `std::thread`.

Внесемо зміни до `Cargo.toml`, щоб додати залежність від `rand` - бібліотеки для генерації випадкових чисел:

```
[dependencies]
rand = "0.8.4"
```

Крок 2: Створення потоків

Тепер створимо функцію, яка виконуватиметься в окремих потоках. У нашому прикладі ми будемо генерувати випадкові числа і виводити їх на екран.

```
use std::thread;
use rand::Rng;

fn generate_random_numbers()
{
    let mut rng = rand::thread_rng();
    for _ in 0..5{
        let num = rng.gen::<u32>();
        println!("Згенероване випадково число: {}", num);
    }
}
```

Крок 3: Запуск потоків

Тепер запусимо кілька потоків, щоб вони виконували функцію `generate_random_numbers()` одночасно.

```
fn main() {
    let handles: Vec <_>=(0..5).map(|_|{
        thread::spawn(|| {
            generate_random_numbers();
        })
    }).collect();
}
```

```
        for handle in handles {  
            handle.join().unwrap();  
        }  
    }  
}
```

У цьому коді ми створюємо вектор `handles`, у якому зберігатимемо хендли наших потоків. Потім з допомогою методу `map()` ми створюємо 5 потоків, кожен із яких виконує функцію `generate_random_numbers()`. У циклі ми чекаємо на завершення кожного потоку за допомогою методу `join()`.

Крок 4: Запуск програми

Тепер збережемо файл і запустимо наш багатопотоковий додаток:

В результаті ми повинні побачити виведення 5 потоків, кожен з яких згенерував 5 випадкових чисел.

Завдання:

1. Модифікуйте програму, щоб кожен потік виконував обчислення чисел Фібоначчі в заданому діапазоні та виводив їх на екран. Підрахуйте час виконання кожного потоку та виведіть його на екран.

2. Реалізувати два способи обробки елементів масиву: послідовний та паралельний. При послідовному способі елементи масиву обробляються і одному потоку, при паралельному – в m потоках. При цьому масив розбивається на m частин, кожна частина обробляється в окремому потоці. Операція обробки - пошук елемента, який зустрічається найчастіше в масиві чисел типу `int`. Масив довжиною 100 000 елементів повинен заповнюватися випадковими числами в межах $(-200, 100)$. Провести тестування та проаналізувати, як залежить час виконання операції обробки від параметра m .