Отчет по лабораторной работе №2 по диспиплине “Парадигмы и конструкции языков программирования”

use std::sync::{Arc, RwLock};

struct User {

**id**: *i32*

}

fn main() {

let rw\_user = RwLock::new(User { **id**: 1 });

let static\_user\_ref: Arc<RwLock<User>> = Arc::new(rw\_user);

let threads = (0..3).map(|\_| {

let user\_ref\_copy: Arc<RwLock<User>> = static\_user\_ref.clone();

std::thread::spawn(move || {

let mut user = user\_ref\_copy.write().unwrap();

user.**id** += 1;

})

});

threads.map(|t| t.join()).for\_each(drop);

println!("{:?}", static\_user\_ref.read().unwrap().**id**);

}