Programación Concurrente - Recuperatorio - 06 de Noviembre de 2024

Nombre: …….. Padrón: ……

Ejercicio 1

Enumere y justifique qué modelo de concurrencia es más conveniente utilizar para cada uno de los siguientes casos de uso

- Calcular productos de matrices para entrenar redes neuronales
- Realizar llamadas a diferentes APIs y combinar sus resultados
- Procesar los archivos de log de acceso de un sitio web muy concurrido
- Backend de una app para editar documentos de forma colaborativa en línea, como un Google Doc

Ejercicio 2

En una revisión de código se encontró el siguiente fragmento

```
const N:usize = 10;
fn main() {

    let buffer = Arc::new(Mutex::new(Vec::<u32>::with_capacity(N)));

    let buffer_local = buffer.clone();
    let handle = thread::spawn(move || {
        loop {
            let mut buf = buffer_local.lock().unwrap();
            if buf.len() < N {
                buf.push(rand::thread_rng().gen());
            } else {
                drop(buf);
                thread::sleep(Duration::from_secs(1));
            }
        }
    });

    loop {
        let mut buf = buffer.lock().unwrap();
        if !buf.is_empty() {
            println!("{}", buf.pop().unwrap());
        } else {
            drop(buf);
            thread::sleep(Duration::from_secs(1));
        }
    }

    handle.join().unwrap();
}
```

a. Describa qué problemas presenta este código y proponga una solución
b. Describa por qué son necesarias las llamadas a drop en esta implementación.

Ejercicio 3

Modelar una Red de Petri para la solución propuesta en el punto anterior

Ejercicio 4

Verdadero o Falso. Justifique

a. El estado interno de un actor se encuentra protegido por una exclusión mutua.
b. La llamada al método poll de un Future es bloqueante.
c. Programando en el modelo fork-join se debe tener especial cuidado en evitar condiciones de carrera.
d. Un hilo esperando sobre una condvar sólo puede despertarse cuando otro hilo hace signal de la misma.

Ejercicio 5

Nos encontramos implementando un sistema para una cafetería de especialidad. Los clientes llegarán al mostrador y serán atendidos por algún cajero para tomar su pedido y cobrarlo. Los baristas irán tomando los pedidos ingresados y los prepararán. Por el momento se disponen únicamente de dos máquinas de café, las cuales solo puede utilizar un barista por vez. Finalmente, al tener el pedido listo, el barista llamará al cliente por su nombre para entregárselo.

Diseñe el sistema utilizando el modelo de actores. Para cada uno de ellos especifique su estado interno en pseudocódigo de Rust. Detalle los mensajes que reciben, sus payloads y cómo los procesan también en pseudocódigo.