Universität Bielefeld
Technische Fakultät

AG Genominformatik
Dr. Lucas Robidou

The Rust programming language
Summer 2024 / 2025

Exercises

This exercise sheet requires tokio (`https://tokio.rs/`) as the asynchronous runtime. Add `tokio = version = "1.45.1", features = [full"]` to your dependencies. You'll have to annotate your main function with `#[tokio::main]`.

1 A simple web server with axum

1. Using `https://crates.io/crates/sysinfo`, write a few functions that return data from your system, e.g., `fn get_cpu_info() -> YourOwnStructHoldingDataAboutYourCPU`
2. Taking inspiration from `https://github.com/tokio-rs/axum?tab=readme-ov-file#usage-example`, write a web server that, e.g,. returns a JSON about your CPU when receiving a query on /cpuinfo. Hint: your structs have to be `Serialize` and the function you access has to return `axum::Json`.
3. Visit `http://0.0.0.0:3000/cpuinfo` and chack that everything is OK.

2 A kombucha factory

Kombucha is fermented tea that you can prepare at home[1]. Starting from 0.3L of kombucha:

- Add 2.7L of sweet tea and 0.3L of kombucha in a big glass
- Wait 5 days, so that the kombucha ferments
- You now have 3L of kombucha.

Then you rinse[2] and repeat.

1. Write a Jar struct that holds some amount of liquid, with methods to add and remove liquid, and another one to check if there is at least a certain quantity in the Jar.

Starting from a jar holding 0.3L of kombucha, let's apply the recipe 20 times, asynchronously.

2. Write an async function `fermentation_task` that waits with a while loop until the Jar has at least 0.3L, takes 0.3L, waits 5 seconds, and puts 3L in the Jar.
3. Take a look at the documentation of `https://docs.rs/tokio/latest/tokio/sync/struct.Notify.html#method.notify_waiters`
4. Use a Notify to prevent the active wait.

[1] Do not follow this recipe. Look for a real recipe instead.
[2] literally