

Week 07 Workshop

EXERCISE:

Workshop 7 - Scheduling

PRE-WORKSHOP PREPARATION:

The only pre-workshop work is to watch the lectures, and download the [starter code](#), and play around with it.

IN-WORKSHOP REVISION:

Revision is worked into the flow of the workshop, so there's no specific revision this week.



This week's workshop will explore Rust's function traits and macros in a little bit more detail. This tutorial will start with more of a design discussion than previous tutorials have.

Note that our plan is that we will continue to build on this code next week once we've learned how to use concurrent code. You might want to take that into account while working on this code.

The [starter code](#) implements a simple "scheduler" system. The idea is that the system allows you to create tasks. Tasks have "prerequisites", and they cannot run until all their "prerequisites" have been met.

Specifically, this program implements a "build system". It will be able to build and run multiple programs that depend on each other. Once they've been built, it'll run those programs.

At the start of the workshop, your tutor will ask the following questions:

- Where might this sort of system be required?
- What are the categories of closures Rust has?
- What sort of closure is stored by the scheduler? What does that mean?
- What is the lifetime of that closure? Where does it begin and end?
- What issues does that present?
- Can you construct an example program that would be great to work, that doesn't?
- How might you change the system to overcome those issues?

Once groups have figured out issues, and the solutions to them, groups will have time to implement those solutions.

If there's time at the end of the tutorial, explore creating an `add_task!()` macro. Why might this be useful?

If there's still time, explore adding the ability to use a prerequisite that is a function (i.e. rather than just checking if all prerequisites, writing a function which returns true or false whether the prerequisites have been met).

Important: You'll be reusing the same code for next week's workshop. We strongly suggest saving a copy at the end of the workshop, so that you can work off the same piece of code in the next workshop.