

Lab exercise 5

Cooking with Signals

Due date

Oct 23, 2023 11:59 PM

Academic Integrity	2
Keywords	2
Getting started	3

Academic Integrity

The following actions are strictly prohibited and violate the honor code. **The minimum penalty for plagiarism is a grade of zero and a report to the Aggie honor system office.**

- Uploading assignments to external websites or tutoring websites such as Chegg, Coursehero, Stackoverflow, Bartleby, etc.
- Copying code from external websites or tutoring websites such as Chegg, Coursehero, Stackoverflow, Bartleby, etc.
- Copying code from a classmate or unauthorized sources and submitting it as your work.
- Sharing your code with a classmate.

Keywords

Unix signals, Unix timers

Introduction

This lab is a practical application of signals, both with the base signal/handler formula and timers.

Your program takes in a set of recipe instructions as a CSV file with columns ID, Dependencies, Duration, and Description and execute the steps in order based on their dependencies and duration. This process is very similar to a Gantt chart.

Expected Functionality

Your program's input will be the `-i` flag for the name of the file. The code already parses the input file in the `StepList` class. You will need to determine which steps can be run and set up a timer for the ready steps. If a step is already running, you shouldn't create another timer. Once the time has elapsed, a timer handler function will be called. This will add the step to the completed list and remove the step as a dependency from any relevant steps.

There are two signals that we are concerned with. The first one is **SIGRTMIN**; this gets called when a timer has completed. The second one is **SIGUSR1**, we will use this to asynchronously remove dependencies from the step list. Whenever we have added a step to the completed list, we should raise this signal to remove the step from any dependency list.

Starter Code

You are given the `StepList` and `Step` classes that contain all relevant functions for those classes. These will handle the file parsing, `StepList::extractStepInfo()`, checks what steps are ready to be executed,

StepList::getReadySteps(), removes dependencies, StepList::RemoveDependency() (Note: this needs to be called from a handler). The Step::PrintComplete() should also be called when that step completes.

You are given the file **MasterChef.cpp**, which contains the main function of your program. This file has 3 sections that you are responsible for completing.

- **Section 1:** main() - This section will associate the SIGRTMIN and SIGUSR1 signals to the appropriate handlers using sigaction() and signal(), respectively. We use sigaction() for the timers as this can pass pointers to the step object to the handler. Review the man pages for details on these commands. The main will also continuously check if there are steps ready to be started and initialize timers when they are using makeTimer().
- **Section 2:** timerHandler() - This section will be called when the timer has elapsed; you will need to print that the step has been completed and add it to the completed list. It will also raise the SIGUSR1 signal to remove the step from the dependency lists.
- **Section 3:** RemoveDepHandler() - This section checks the completed step list and erases the steps from all other steps using the StepList::RemoveDependency() method.

Rubric

There are three tests that check the elements of the program which you have been asked to fix, and grades will be assigned as the score you see on GitHub classrooms (under the Actions tab):

1. Compilation (5 pts)
2. No Leaks (5 pts)
3. Correct output (33 pts)
4. Use of timers (24 pts)
5. Use of signal (33 pts)

Getting started

1. Go to the assignment's GitHub classroom: <https://classroom.github.com/a/X0fNJehn>
2. Create a repository for your project.