

CS563 Assignment 2: Programming with Actors

Instructor: Xinghui Zhao

Due: 11:59pm, February 2, 2020

1 Overview

The purpose of this assignment is to give you some practice on programming using the Actor model of concurrency. You will implement a simple Actor system and use it to solve a problem. For this assignment, you can choose a programming language to use.

2 Actor System

First, you should implement a simple Actor system based on the Actor model of concurrency. Actors are autonomous computational entities which communicate with each other using buffered, asynchronous, point-to-point messages. An actor encapsulates a state, a number of methods (which can change the state of the actor), and a thread of control, as shown in Figure 1. Actors are distributed over time and space. Each actor has a globally unique mail address, and it maintains a queue of the unprocessed messages it has received.

The messages in an actor's message queue are processed one by one according to the order of arrival. While processing a message, an actor may carry out one of the three primitive actions:

- Create a finite number of new actors with some predefined behaviors. The creator actor knows the addresses of the new actors;
- Send messages to other actors. An actor can send a message to another actor only if it knows the name of the destination actor.
- Change the actor's own state and be ready to process the next message.

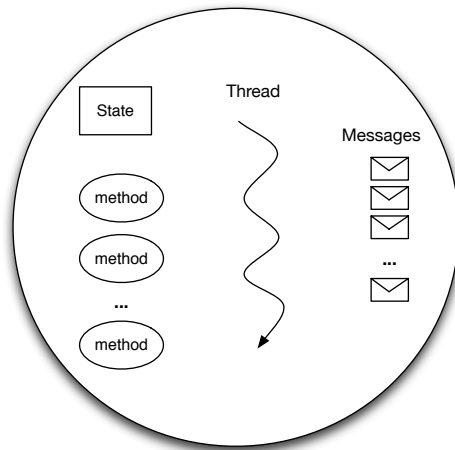


Figure 1: Actor

Assignment Project Exam Help

An Actor system provides a foundation for implementing concurrent, message-passing applications. Follow the instructions below to implement a simple actor system:

<https://tutorcs.com>

1. Design a naming mechanism. Each actor must have its own unique name.
2. Design a data structure for the mailbox, which is used to keep unprocessed messages.
3. Implement the “Actor” class. This class provides a template for all actor computations. The basic components include a unique name, a mailbox (message queue), and a thread which keeps processing the messages in the queue.
4. The Actor class may contain basic primitives (methods) for creating a new actor, and sending a message to another actor.
5. Design a message delivery mechanism, i.e., how does the system deal with actor messages?

WeChat: cstutorcs

Once these steps are done, you have a simple actor system which works locally on one computer. An actor application can be implemented by extending the Actor class to include application specific methods.

Note that GUI is not required for this assignment.

3 Actor Application

Now that you have an actor system, you can use it to write concurrent applications. In this section, you will use the Actor system you have implemented to solve the gravitational N-body problem.

3.1 N-body problem

The n-body problem is the problem of predicting the motion of a group of celestial objects that interact with each other gravitationally. Here is a wikipedia page for your reference: http://en.wikipedia.org/wiki/N-body_problem. In addition, a more detailed description of this problem can be found in Section 11.2 of the recommended textbook. A scanned copy of that section is attached.

3.2 Your tasks

You need to do the following:

1. Read Section 11.2.1 and make sure that you understand the problem and the sequential algorithm (Figure 11.9).
2. Ignore Section 11.2.2 (Shared-Variable Program), and read Section 11.2.3 (Message-Passing Programs). There are three message passing algorithms described in Section 11.2.3, and they use different communication patterns: Manager/Worker, Heartbeat, and Pipeline. In this assignment, you only need to focus on the Manager/Worker pattern.
3. Make sure you understand the Manager/Worker algorithm, then try to implement it using your actor system.
4. Run your program for different “n” values and compare their performance.
5. Fix the “n” value, and run the program using different “w” values (number of workers), and compare the performance.
6. Write a performance analysis report to describe your findings and try to explain them.

3.3 What to Hand In

Submit the following:

1. Source code.
2. A ReadMe file in which you describe how to run your program.
3. Some sample outputs from your program (you may use a screenshot).
4. The performance analysis report.

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs