

Week8 Summary

General idea of Chapter 3 – Computation in Actor Systems of the book:

In this chapter, the author examined the structure of actor paradigm. The author generally used an informal way to share his ideas in three sections, including explaining actors and communications, outlining the constructs which can define a minimal actor language and also giving some examples and “pseudo-code” of actor programs.

Details about the paper:

In the first section, the author talked about how to define an actor system. We first need to define a task. It is a tag that distinguishes it from all other tasks and a target which is the mail address to which communications are to be delivered. The second thing is the actor's behavior, which is similar to a data-driven system like dataflow. An actor processes only those tasks whose target corresponds to its address. Therefore, you can define an actor by specifying its mail address (corresponds to a sufficiently large mail queue) and its behavior (a function that the communication accepted).

In the second section, the author talked about programming with actors. In this part, the author defined the constructs necessary for the kernel of a minimal actor language. Also, this part illustrates some general mechanisms for implementing control structures. An actor generally accepts a single communication as “input”. An actor program generally consists of behavior definitions, new expressions, send commands, receptionist declarations and external declarations. To illustrate the implementation process, the author gave several examples including a stack, a recursive factorial and external actors.

In the third section, the author gave the syntax for two minimal languages, SAL and Act. Behavior definitions in SAL are declarative and the language avoids specifying any type structures in the programming languages. As for Act, it is a sufficient kernel for the Act3 language. One basic distinction between Act and SAL is that the former uses a keyword-based notation while the latter uses a positional notation.

My Personal Thoughts:

The actor's programming style is implemented in concurrent programming. OOP languages like C++, Java and many other languages have such features. By using context switch, we can take advantage of the CPU while one thread is idle. The week 8 practice just uses this kind of programming skills, by using send function to interact between different threads. Different threads cooperate together to produce the final output. This chapter just gave me more in-depth theory analysis of such programming style.