PCM – Assignment 3

Actor Architecture:

The actor architecture is used in concurrent computation, treating every “Actor” as a different individual. Actors can make local decisions, create more Actors and send messages to other Actors. Actors can only modify their own private state, but they can affect and communicate with other Actors through messages.

Implementation:

To implement a Binary Tree that uses Actors the Actors have the following attributes:

* A mailbox (LinkedBlockingQueue)
* A number (unique id)
* A left and right reference to other Actors

Each Actor has its own thread that is always running till it receives a removal message. The system is capable of producing and receiving seven different types of messages, Add, Contains, Remove, Reorganize, AddResponse, ContainsResponse, RemoveResponse. Every message contains a number that indicates the recipient or the response of the message and the sender of the message, in the case of a Response message it’s also sent the supervisor of the recipient and the side of the leaf Actor corresponds to.

To find the recipient of the message the Actor compares the number present in the message to its own, and in the case of being inferior it send to the left leaf or to right leaf in case of it being superior.

The add message is forwarded down the tree to find if the number in the message is already present in the tree, in case it reaches the bottom of the tree without finding an Actor with the number of the message it is then created an Actor with that number. The Actor responsible for creating the new Actor sends an AddResponse message to the RootActor. In the case the number is already attributed to an Actor present in the tree it is sent an AddResponse message to the RootActor notifying the number is already present.

The contains message is forwarded down the tree to find if the number in the message is present matches an Actor present in the tree. When it finds the Actor with the number present in the message or it reaches the bottom of the tree without finding the number it sends a ContainsResponse message to the RootActor.

The remove message is forwarded down the tree till it finds the Actor with the number present in the message. When that Actor is found, it then proceeds to determine how many leaf Actors it has. In the case of only one it sends a message to his supervisor with the reference of that Actor. In the case it has two leaf Actors it starts a process to find the Actor with the smallest number on its right subtree. Once it’s found that Actor sends a message to the supervisor of the Actor that it was removed with his reference so that it can replace the position of the removed Actor.

Regardless of the number of leaf Actors the removed Actor then deletes all messages present on his mailbox, the references it contained of his leaf Actors and signals himself to stop processing messages, followed by sending a RemoveResponse message to the RootActor signalling the success of the operation.

Garbage Collection:

When an Actor is removed every reference to the Actor is removed and his mailbox is emptied.