Exercise „Programming Erlang“

Prof. Dr. Marcel Kyas

Blatt 1, 5. October 2009

Freie Universität Berlin

Exercise 1 (6 Points) This exercise deals with the *leader election* problem. The goal is to write an algorithm that allows n processes to select a leader process among these.

- Write a function that starts n processes in a ring topology and that assigns to each process in that ring some random but unique integer ID.
- Implement the following election algorithm:
  - Let each process send its ID "clockwise" around the ring.
  - A process that receives an ID larger than its own ID lost the election and just passes received IDs around the ring.
  - The process that receives its own ID wins the election.

Exercise 2 (6 Points) This exercise deals with the *leader election* problem. The goal is to write an algorithm that allows n processes to select a leader process among these.

- Write a function that starts n processes in a ring topology and that assigns to each process in that ring some random but unique integer ID.
- Implement the following election algorithm:
  - A process is in two states: It is either participating in the election or forwarding messages (called the relay mode).
  - Let each process send its ID "clockwise" around the ring, receive one from ID and sent ID "clockwise" again and receiving another id, say `Id1` and `Id2`.
  - If `Id1<=max(Id,Id2)`, the node switches to relay mode.
  - Otherwise it continues the next round of the election with assuming `Id1` as its new identity.
  - The process that receives its own ID wins the election.

This algorithm has been proposed by Peterson.

Exercise 3 (6 Points) Write a process that receives strings representing mathematical expressions and sends their value back to the sender.

Now write a process that reads strings from the console, sends it to the process and prints the value on the console.

The strint "q" shall exit the program. Use error trapping to deal with errors.