

Synchronisation in Distributed Systems

Algorithm of Cristian

We want to program the algorithm of Cristian in Erlang.

1. Time Server Process

Please define a central time server process as an extension of the clock service of the last exercise. It has a local time and can also access the Erlang system time by means of the `erlang:timestamp()` function. We assume this to be an acceptable approximation of UTC for our purposes.

The Erlang standard function `erlang:timestamp()` returns a tuple of the form

$$\{\text{MegaSecs}, \text{Secs}, \text{MicroSecs}\}$$

with three integer values which determine the number of micro seconds since January, 1st. 1970, 00:00 GMT. See <https://www.erlang.org/doc/man/erlang#timestamp-0> for details.

The time server process should react to the following messages:

- `{get, Pid}`
Ask the time server for the current time. It sends an appropriate message of the form `{Cutc, t2, t3}` to the Process `Pid`, where t_2 is the the server's local time immediately after receiving the request message, t_3 the local time just before sending the reply, and C_{UTC} the determined global time (either a tuple of the above structure or a summed up number of microseconds since 1970-01-01T00:00Z).
- `show`
Display the time servers internal state.

2. Client Processes

Please define client processes as an extension of the clock service of the last exercise. In addition to the existing messages client processes should also react to these messages:

- `adjust`
The client processes will adjust its time by capturing the current local time t_1 before adjusting the time, then sending a `get`-message to the time server process and finally using the time server's result along with the current local time t_4 when receiving the time server response to update its clock value.
 - `show`
Display the client's internal state.
3. Please create a network of one server process and three client processes with different clock speeds (frequencies) and let the processes synchronize their clocks by manually sending them `adjust` messages. What messages are exchanged in consequence?
 4. Please extend the client process so that they self-adjust every now and then. Output their internal state and observe the system running, deviating and adjusting.

Please be prepared for face-to-face technical discussions. These will happen during the on-site tutorial sessions.