

Synchronisation in verteilten Systemen

21. Uhren-Prozesse

Schreiben Sie bitte ein Erlang-Modul `clock`, in dem ein Prozess definiert wird, der als Zustand eine Uhrzeit (gezählt in Ticks) verwaltet, die sich in konfigurierbarer Geschwindigkeit erhöht.

a) `clock`-Prozesse sollen folgenden Nachrichten bearbeitet:

- `{set, Value}`
Die interne Uhrzeit wird auf den Wert `Value` gesetzt.
- `{get, Pid}`
Die interne Uhrzeit wird, z.B. in der Form `{clock, Value}` an den Prozess `Pid` geschickt.
- `stop`
Der `clock`-Prozess wird beendet.

Mit Hilfe des Konstrukts `receive ... after` sollen die `clock`-Prozesse in der konfigurierten Frequenz ihren internen Zähler erhöhen.

Außerdem kann es sinnvoll sein, `clock`-Prozesse mit einer Trace-Möglichkeit inklusive Nachrichten zum An- und Ausschalten zu versehen, die ihr Arbeiten visualisiert.

b) `start`-Funktion

Schreiben Sie bitte im Modul `clock` eine Funktion `start`, die `clock`-Prozesse startet. Sie soll einen Parameter besitzen, der die Geschwindigkeit des `clock`-Prozesses bestimmt.

c) `get`-Funktion

Kapseln Sie bitte das Rücksenden der Uhrzeit von `clock`-Prozessen in einer `get`-Funktion im Modul `clock`, die zunächst eine asynchrone `get`-Nachricht verschickt, dann eine `clock`-Antwort erwartet und die Uhrzeit aus dieser Nachricht extrahiert.

22. Berkeley-Algorithmus

a) Definieren Sie bitte Prozesse, die den Client-Teil des Berkeley-Algorithmus zur Uhrensynchronisation realisieren. Diese Prozesse sollen einen assoziierten Uhrenprozess besitzen und auf folgende Nachrichten reagieren:

- `{get_skew, Systime, Pid}`
Die Unterschied zur Systemzeit `Systime` wird bestimmt und an den Prozess `Pid` gesendet.
- `{adjust, Delta}`
Die interne Uhr wird um den Wert `Delta` korrigiert.

b) Definieren Sie bitte einen Prozess, der den Daemon-Teil des Berkeley-Algorithmus realisiert. Er hält eine Liste von Client-Prozessen und koordiniert die Synchronisation durch Versenden von `get_skew`- und `adjust`-Nachrichten.

Er selbst soll auf die Nachricht `synchronize` reagieren.

c) Erstellen Sie ein Netz aus vier Client-Prozessen und einem Daemon-Prozess mit jeweils unterschiedlich gehenden Uhren und führen Sie eine Synchronisation durch.