
Concurrent Programming

Blatt 5

Abgabe: 3.6.2002

1. [15 Punkte] Viele Client/Server-Anwendungen verwenden *Timeouts* um festzustellen, ob der Prozeß mit dem kommuniziert wird noch lebt. Programmiere einen Timeout-Mechanismus mit Hilfe von Rendezvous! Der Aufruf von `wake-at-rv` mit dem Parameter t soll ein Rendezvous erzeugen, das sich zum Zeitpunkt t — und nur dann — synchronisieren läßt.

In Scheme 48 sind zu diesem Zweck eine Prozedur `real-time` (in der Struktur `time`) und eine Prozedur `sleep` (in der Struktur `threads`) definiert: (`real-time`) liefert die aktuelle Zeit seit dem Programmstart in Millisekunden, und (`sleep t`) blockiert den momentanen Thread für t Millisekunden.

- (a) Programmiere eine möglichst einfache Version von `wake-at-rv`!
 - (b) Programmiere `wake-at-rv` mit einem Client/Server-Modell:
 - Schreibe dazu zunächst eine Prozedur `make-tick-rv`, die eine Zeitspanne t als Argument akzeptiert. Die Prozedur soll einen Event zurückgeben, auf dem sich alle t Millisekunden (und nur dann) synchronisieren läßt.
 - Schreibe dann einen Uhrzeit-Server. Der Uhrzeit-Server verwaltet eine Anzahl schlafender Klienten, die zu bestimmten Zeitpunkten aufgeweckt werden wollen. Der Server nimmt dazu *jederzeit* Anfragen entgegen, die von Klienten via `wake-at-rv` an den Server verschickt werden. Außerdem soll der Uhrzeit-Server in regelmäßigen Intervallen alle Klienten aufwecken, deren Zeit verstrichen ist. Achte darauf, daß der Server auch dann funktioniert, wenn ein Klient vor Verstreichen der Frist ein anderes Rendezvous wählt.
2. [15 Punkte] Ein `ack-timeout-channel` ist eine besondere Art synchroner Kanal, bei der ein Prozeß eine Nachricht sendet, die dann von allen Prozessen aus diesem Kanal gelesen werden kann. Die lesenden Prozesse müssen diese Nachricht innerhalb einer bestimmten Zeitspanne lesen und bestätigen, da die Nachricht sonst ungültig wird. Die Operation `broadcast-message` verteilt eine Nachricht mittels selektiver Kommunikation genau einmal an jeden Empfänger. Überschreitet mindestens ein Empfänger die erwähnte Zeitspanne, liefert `broadcast-message` den Wert `#f` zurück, sonst `#t`. Programmiere eine solche Datenstruktur und die Operation `broadcast-message` und gib geeignete Testfälle an!