

Koordination in verteilten Systemen

Wir wollen Wahl-Algorithmen in Erlang realisieren.

Als Wahlwerte können die Prozess-IDs verwendet werden, die sich über `<` vergleichen lassen. Wenn wir aber das Ersetzen eines Prozesses behandeln wollen, eignen sich ggf. eigenvergebene Wahlwerte besser.

Die benötigten Listen-Operationen (`max`, `append`, `member`) können dem Module `lists` entnommen werden.

Ein Prozess gilt als nicht erreichbar, wenn er auf eine Nachricht nicht innerhalb einer viertel Sekunde (250ms) antwortet. Weil wir Antworten erwarten, kann keine einfache asynchrone Erlang-Kommunikation verwendet werden. Statt dessen kann synchrone Kommunikation etwa über die Funktion `rpc`:

```
rpc(Pid, Request) ->  
    Pid ! {self(), Request},  
    receive  
        {Pid, Response} ->  
            Response  
    after 250 ->  
        unreachable  
    end.
```

eingesetzt werden.

23. Bully-Algorithmus

Schreiben Sie bitte ein Erlang-Modul `bully`, in dem das Verhalten für Prozesse definiert wird, die den Bully-Algorithmus zur Wahl eines Koordinators innerhalb einer Gruppe von Prozessen realisieren.

Der Zustand der Prozesse enthält sinnvollerweise zum einen den aktuellen Koordinator (ggf. den eigenen Wahlwert und) zum anderen die Liste aller anderen Prozesse der Gruppe.

a) Realisieren Sie bitte die Grundfunktionen des Bully-Algorithmus:

- Senden von `election`-Nachrichten an alle Prozesse mit höherer Nummer
- Verarbeiten von `election`-Nachrichten
- Senden und Verarbeiten von `coordinator`-Nachrichten (mit Ausgabe des Wahlergebnisses)

Mit Hilfe einer `startElection`-Nachricht wird ein Prozess dazu veranlasst, eine neue Wahl abzuhalten.

b) Schreiben Sie bitte eine Funktion `setup`, die eine Gruppe von Prozessen aufbaut.

c) Testen Sie die Wahl des Koordinators durch Versenden von `startElection`-Nachrichten an ausgewählte Prozesse. Beenden Sie einzelne Prozesse und wiederholen Sie die Wahl. Starten Sie neue Prozesse und nehmen Sie sie in die Gruppe auf.

d) Messen Sie, wie viele `election`-Nachrichten verschickt werden, indem einem Zähl-Prozess jeweils eine Kopie der Nachricht zugeschickt wird.

24. Ring-Algorithmus

Die Implementierung des Ring-Algorithmus ist ähnlich zum Bully-Algorithmus, aber im Detail unterschiedlich. Insbesondere muss ein Ring von Prozessen anstatt einer Menge von Prozessen realisiert werden.

Schreiben Sie bitte ein Erlang-Modul `ring`, in dem das Verhalten für Prozesse definiert wird, die den Ring-Algorithmus zur Wahl eines Koordinators realisieren.

Der Zustand eines Prozesses enthält wiederum sinnvollerweise zum einen den aktuellen Koordinator (ggf. den eigenen Wahlwert und) zum anderen die Liste seiner Nachfolgerprozesse.

a) Realisieren Sie bitte die Grundfunktionen des Ring-Algorithmus:

- Senden und Verarbeiten von `election`-Nachrichten (mit Überspringen von inaktiven Nachfolgern)
- Erkennung eines vollständigen Umlaufs
- Senden und Verarbeiten von `coordinator`-Nachrichten (mit Ausgabe des Wahlergebnisses)

Mit Hilfe einer `startElection`-Nachricht wird ein Prozess dazu veranlasst, eine Wahl abzuhalten.

b) Schreiben Sie bitte eine Funktion `setup`, die einen Ring von Prozessen aufbaut.

c) Testen Sie die Wahl des Koordinators durch Versenden von `startElection`-Nachrichten an ausgewählte Prozesse. Beenden Sie einzelne Prozesse und wiederholen Sie die Wahl. Starten Sie neue Prozesse und nehmen Sie sie in den Ring auf.

d) Messen Sie, wie viele `election`-Nachrichten verschickt werden, indem einem Zähl-Prozess jeweils eine Kopie der Nachricht zugeschickt wird. Wie verhält sich das zur Anzahl der Nachrichten im Bully-Algorithmus?

25. Zusatzaufgabe

a) **Beitritt**

Neue Prozesse sollen der Bully-Gruppe bzw. dem Ring zu beliebiger Zeit beitreten können:

- Mit Hilfe der Nachricht `{join Pid}` an einen Prozess der Bully-Gruppe bzw. des Rings, kann ein Prozess signalisieren, dass er der Gruppe beitreten will. Die internen Verwaltungsinformationen aller Prozesse sollen dadurch aktualisiert werden. Überlegen Sie, wie es dabei zu Problemen kommen kann, wenn zwei neue Prozesse zeitgleich versuchen, an unterschiedlichen Stellen beizutreten. Wie lässt sich dieses Problem lösen?
- Beim Beitreten der Gruppe soll unmittelbar eine Wahl abgehalten werden, um einen neuen Koordinator zu bestimmen.

b) **Warten oder Ersetzen von Prozessen**

Modifizieren Sie bitte einige Prozesse so, dass sie länger als 250ms pausieren oder nach dem Beenden durch neue Prozesse mit gleichem Wahlwert ersetzt werden. Überlegen Sie, wie es dann zu Fehlverhalten bei der Wahl des Koordinators kommen kann und provozieren Sie bitte diese Situation.