Prof. Franz Schweiggert
Christian Ehrhardt

Systemnahe Software (SS 2006)

Abgabetermin 20.Juni 2006

Pipes

Pipe Tests (3 Punkte)

Schreibt ein oder mehrere Programme, die folgende Eigenschaften von Pipes testen:

- Eine normale Kommunikation zwischen zwei Prozessen.
- Das Blockieren beim Lesen bzw. Schreiben.
- Wieviele Daten kann man in eine Pipe schreiben, ohne zu blockieren (wenn niemand die Daten liest)?
- Wann tritt ein SIGPIPE auf?

Die Programme sollen (auch wenn es verwandte Beispiele im Skript gibt) selbst geschrieben sein, und Ihr solltet Eurem Tutor wie immer erklären können, warum die Programme sich so verhalten, wie sie sich verhalten.

Josephus oder Prozesse auszählen (7 Punkte)

Die Legende

Das sogenannte Josephus-Problem geht auf den Historiker Flavius Josephus zurück, der der Legende nach im Jahre 67 mit einer Gruppe von Widerstandskämpfern von den Römern in einer Höhle umzingelt wurde. Die Widerstandskämpfer beschlossen, sich selbst umzubringen, um nicht in römische Gefangenschaft zu geraten. Die Reihenfolge wurde durch einen Auszählreim bestimmt, wie er auch heute noch von Kindern verwendet wird. Die Leistung von Josephus bestand darin, den Platz im Kreis zu finden, der sich als letzter selbst hätte töten sollen. Statt dessen ergab er sich als einziger Überlebender den Römern.

Das Problem

Beim Josephus-Problem stehen also N Personen in einem Kreis. Bei der ersten Person beginnend wird reihum bis zu einer Zahl k gezählt. Die Person an dieser Stelle muß den Kreis verlassen, anschließend wird bei der nächsten Person im Kreis beginnend wieder bis k gezählt. Ziel ist es, als letzte Person noch im Kreis übrig zu bleiben.

Simulation mit Hilfe von Prozessen

Dieses Verfahren soll jetzt mit Hilfe von Pipes und Prozessen simuliert werden, es werden also N Prozesse erzeugt und jeder Prozeß kann über eine Pipe Daten an den nächsten Nachbarn im Kreis schicken. Nachdem dieser Kreis von Prozessen aufgebaut ist, beginnt der erste damit, seinem Nachbarn die Zahl 1 zu schicken. Sobald ein Prozeß von seinem Nachbarn eine Zahl zugeschickt bekommt, erhöht er sie um 1. Ist damit der Wert k erreicht scheidet der Prozeß aus und gibt die Zahl 0 statt der Zahl k weiter.
Allerdings darf durch das Ausscheiden der Kreis von pipes natürlich nicht unterbrochen werden. Deshalb gibt ein ausgeschiedener Prozeß immer genau die Daten, die er von seinem Nachbarn empfängt unverändert an den anderen Nachbarn weiter.

Das Ende

Dadurch, daß ausgeschiedene Prozesse nicht gleich terminieren müssen wir uns noch überlegen wie das Ende vorsich gehen soll. Dazu wird eine zweite Zahl im Kreis herum geschickt, nämlich die Zahl der noch nicht ausgeschiedenen Prozesse. Ein Prozeß der ausscheidet reduziert diese Zahl natürlich um 1. Wenn die Zahl bei 0 ankommt schickt er keine Daten weiter sondern schließt seine ausgehende Pipeline Verbindung.
Alle Prozesse terminieren, wenn es aus der Pipeline nichts mehr zu lesen gibt (End-of-File).