

# Übung zur Vorlesung Soft- und Hardware-Systeme

SS 2006

Blatt 3

## **Aufgabe 7:**

Betrachten Sie das Erzeuger-Verbraucher-Problem mit einem einzigen Erzeuger und  $n$  Verbrauchern, wobei Erzeuger und Verbraucher als eigenständige Prozesse realisiert sind (mittels `fork` und `exec`).

Der Erzeuger generiert zufällige Zahlen und schreibt diese in einen Puffer. Jeder der Verbraucher (zu einem Zeitpunkt immer höchstens einer) liest eine Zahl aus dem Puffer und gibt diese gemeinsam mit seiner Prozessnummer aus.

Implementieren sie ein Verfahren, bei dem der Erzeuger den Zugriff auf den Puffer mittels gemeinsamer Variablen synchronisiert,

- bei dem jeder Verbraucher, der Interesse hat, auch nach endlicher Zeit bedient wird
- und das nicht blockiert, wenn ein Verbraucher ausscheidet (mittels `kill` den Prozess beenden).

Zur Synchronisation sollen gemeinsame Variablen verwendet werden, keine Semaphore, Bedingungsvariablen oder ähnliches.

## **Aufgabe 8:**

Im afrikanischen Urwald wird ein Seil über eine tiefe Schlucht gespannt, so dass Paviane daran von einer auf die andere Seite hangeln können. Mehrere Paviane können gleichzeitig die Schlucht überqueren, solange sie sich in die gleiche Richtung bewegen. Falls zur gleichen Zeit Paviane in verschiedenen Richtungen das Seil überqueren, tritt ein Deadlock auf, da die Paviane nicht übereinander hinweg klettern können.

Schreiben Sie mit Hilfe von Semaphoren oder Bedingungsvariablen ein Programm, das Deadlocks und Starvation vermeidet. Die Aktionen der Paviane sollen innerhalb von Threads ablaufen.