
ÜBUNGEN ZU THEORIEBLOCK 2

ÜBUNGSAUFGABE 2.1 - PROZESSE UND THREADS

Sie finden in GIT ein Programm `printData`

In dem Programm wird über `fork` ein Kindprozess erzeugt. d.h. nach dem Starten des Programms gibt es zwei Prozesse, den Vaterprozess und den Kindprozess.

- Sehen Sie sich die Ausgaben beider Prozesse an. Haben Sie diese Ausgaben so erwartet?
- Bauen Sie in die beiden Prozesse ein `sleep` ein. Danach starten Sie das Programm und sehen sich in der Konsole mit dem Kommando `ps` oder `top` Ihre Prozessliste an.
- Gegeben ist das Programm `myProgramm.c`. Überladen Sie Ihren Kindprozess damit und starten erneut das Programm. Versuchen Sie auch den Fehlerfall für `exec` zu testen.
- Schreiben Sie das Programm jetzt so um, dass anstelle der beiden Prozesse zwei Threads auf die globale Variable `data` zugreifen. Was erwarten Sie in diesem Fall für Ausgaben?

ÜBUNGSAUFGABE 2.2 - SEMAPHORE UND PIPE

Sie finden in GIT das Programm `random`.

Der Prozess berechnet fortlaufend Zufallszahlen. Diese Zufallszahlen werden auf der Konsole ausgegeben. Durch Drücken der `return`-Taste auf der Tastatur kann die Berechnung der Zufallszahlen angehalten und wieder gestartet werden. Mit Eingabe von `Ctrl+D` (→EOF) kann der Prozess beendet werden.

Der Prozess besteht aus 3 Threads:

- **RandomThread.**
Dieser berechnet fortlaufend neue Zufallszahlen und schreibt jede neue Zufallszahl in eine Pipe. Über eine Semaphore wird dieser Thread gesteuert d.h ist die Semaphore gesetzt, können fortlaufend Zufallszahlen generiert werden. Ist die Semaphore nicht gesetzt hält der Thread an.
- **AusgabeThread:**
Dieser Thread liest von der pipe eine Zufallszahl und gibt diese auf der Konsole aus
- **StartStop-Thread:** Dieser Thread ist der **main-Thread!**
Er liest von der Konsole eine Eingabe. Ist die Eingabe `newline` und arbeitet der `randomThread` gerade wird der `randomThread` über die Semaphore gestoppt. Ist der Zufallsthread angehalten wird dieser wieder aktiviert. Bei Eingabe von `Ctrl+D` wird der Thread und damit der Prozess beendet

In dem Programm fehlt die Synchronisation über Semaphore und pipe. Bitte vervollständigen Sie das Programm.