Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

BS Praktikum SS 2016 - Gruppe 2
Aufgabenblatt 02 - Entwurf
Synchronisation und Thread Generierung

Alexander Mendel & Karl-Fabian Witte

Abgabe: 1. Mai 2016

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

1 Aufgabe

Es soll ein Contoller-Producer-Consumer System in C geschrieben werden. Wobei der Aufgaben wie folgt untereilt werden:

Producer_1 Thread Schreibt in den Fifo alle 3s ein kleinen Buchstaben aus dem Alphabet.

Producer_2 Thread Schreibt in den Fifo alle 3s ein Großen Buchstaben aus dem Alphabet.

Consumer Thread Nimmt alle 2s ein Zeichen aus dem Puffer und gibt es auf der console aus.

Control Thead Steuerung des Verbraucher-Erzeuger Systems mit Tastatureingaben

Main Thread Initialisierung und Starten der Threads. Beendet das Programm erst, wenn alle Threads terminiert sind.

2 Entwurf

2.1 module

main.c main(), init()

pcc.c control(), consume(), produce_1(), produce_2() +, terminierungsMethoden
(ggf. für die Übersicht wird für jede Funktionalität ein eigenes modul erstellt.)

fifo.c initFifo(), pop(), push(input), destroy()

errInfo.h MACRO für Errorhandling

Es ist anzumerken, dass die Parameterlisten sowie Rückgabetypen noch nicht entschieden sind. Dies soll nur ein grober überblick der Module sein. Es gibt momentan nämlich noch Probleme mit der Übergabe der Argument mittels Struktur. Im Notfall wird auf andere Übergaben zurückgegriffen.Header werden entsprechend angelegt.

2.2 Synchronisation und implementierungs Entwurf

Der über den Controller werden mittels pthread_conds der Consumer und die Producer gesteuert, wo bei über getchar() aus stdio.h der Befehl über das Keyboard gelesen wird und mittels swich-case wird dieser ausgewertet.

Der Fifo ist ein Ring-buffer mit Array und wird mit initFifo() initialisiert. Der Overflow und Underflow wird über eine zwei Semaphoren gesteuert (empty: Anzahl der freien plätze, taken:

Anzahl der belegten Plätze). Dabei werden die beiden entgegengesetzt in-/dekrementiert. Die Manipulation pop und push auf das Array sind über einen Mutex synchronisiert. Beispiel pop: wait(taken); lock(mutex); (auslesen und first auf den nächsten platz setzten) unlock(mutex), post(empty).

Der Controler hat für jeden Thread eine Conditional Variable, mit der er die Threads blocken bzw. straten kann. Die Prüfung CV wird bei den Producern und beim Consumer jeweils am Anfang der Whileschleife gemacht.

für die Fehlermeldungen wird ein Macro geschrieben der mit perror den wert errno auswertet, wenn kein EXIT_SUCCESS zurückgegeben wird.

In der main-Methode werden die Threads und der Fifo initialisiert. Dann wartet die Main auf die Anweisung des Controllers (via Conditional Variable), bevor diese die Terminierung aller Threads einleitet.