

Entwurfsdokument

PRAKTIKUM 2 THREADS

HAUKE GOLDHAMMER (2286057)

MARTIN WITTE (2285586)

Sonntag, 05 November 17

I. AUFGABENSTELLUNG:

Zu erstellen ist ein Programm, mit folgenden Anforderungen:

- Einen Mainthread hat, welcher die anderen Threads startet und wartet, bis alle anderen Threads fertig sind.
- Einen Producer_1 Thread, der a, b,..., y, z ; in einen Puffer schreibt und ausgibt, was er getan hat. Er blockiert, wenn der Puffer voll ist und schreibt weiter, wenn wieder Platz ist.
- Einen Producer_2 Thread, wie Producer_1, aber Großbuchstaben.
- Einen Consumer Thread, welcher alle 2 Sekunden ein Zeichen aus dem Puffer nimmt und dieses ausgibt. Wenn der Puffer leer ist blockiert der Thread, bis wieder Zeichen im Puffer sind.
- Einen Control Thread, dieser soll mit Tastatureingaben, das System steuern.

I. Bedienungsanforderungen:

I.1 Tastatureingaben:

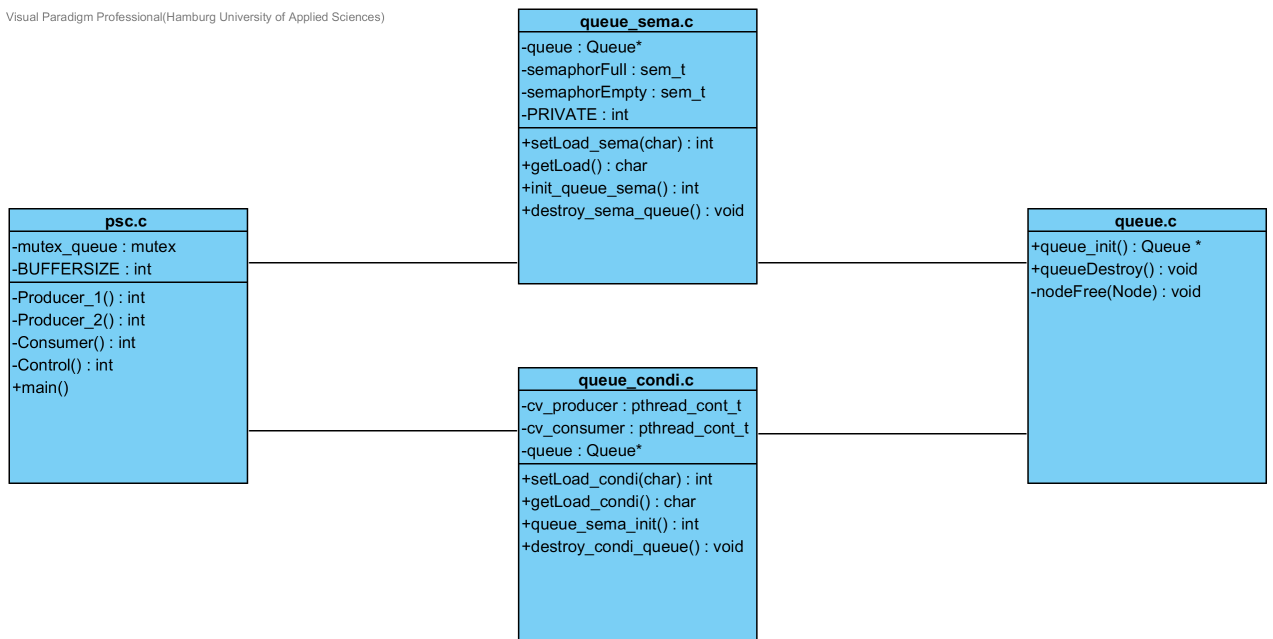
- '1' → Start/Stop Producer_1
- '2' → Start/Stop Producer_2
- 'c' oder 'C' → Start/Stop Consumer
- 'q' oder 'Q' → Main-Thread beendet das System
- 'h' → hilfe

II. WEITERE ANFORDERUNGEN UND LÖSUNGSANSÄTZE:

Anforderung	Lösungsansatz
Wahl zwischen den 2 Varianten	args bei Programmstart(-s,-c), dann arbeitet das Prog entweder mit Queue_sema oder Queue_condi
Kein Polling	zum blockieren mutex nutzen
Angehaltene Threads müssen blockiert werden	mutex nutzen
FIFU_1 mit Semaphoren	
FIFU_2 mit Conditional Variables	
Makefile erstellen	
Fehlermeldungen gem. POSIX bzw. errno	Fehler, wenn möglich, im Prog abhandeln, gem. POSIX
Beenden von Threads nicht in kritischen Abschnitten	Threads mitteilen, dass sie sich beenden sollen. Dies tun sie dann außerhalb des kritischen Bereiches.
Mutexe + Semaphore am Ende wieder an BS zurückgeben	destroy-Funktionen (vorher unlock beachten)
Funktionen möglichst generisch	

III. ANGESTREBTE PROGRAMMSTRUKTUR:

Visual Paradigm Professional(Hamburg University of Applied Sciences)



v1-0.png