

# Entwurfsdokument

## PRAKTIKUM 2 THREADS

HAUKE GOLDHAMMER (2286057)

MARTIN WITTE (2285586)

6. November 2017

### 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:

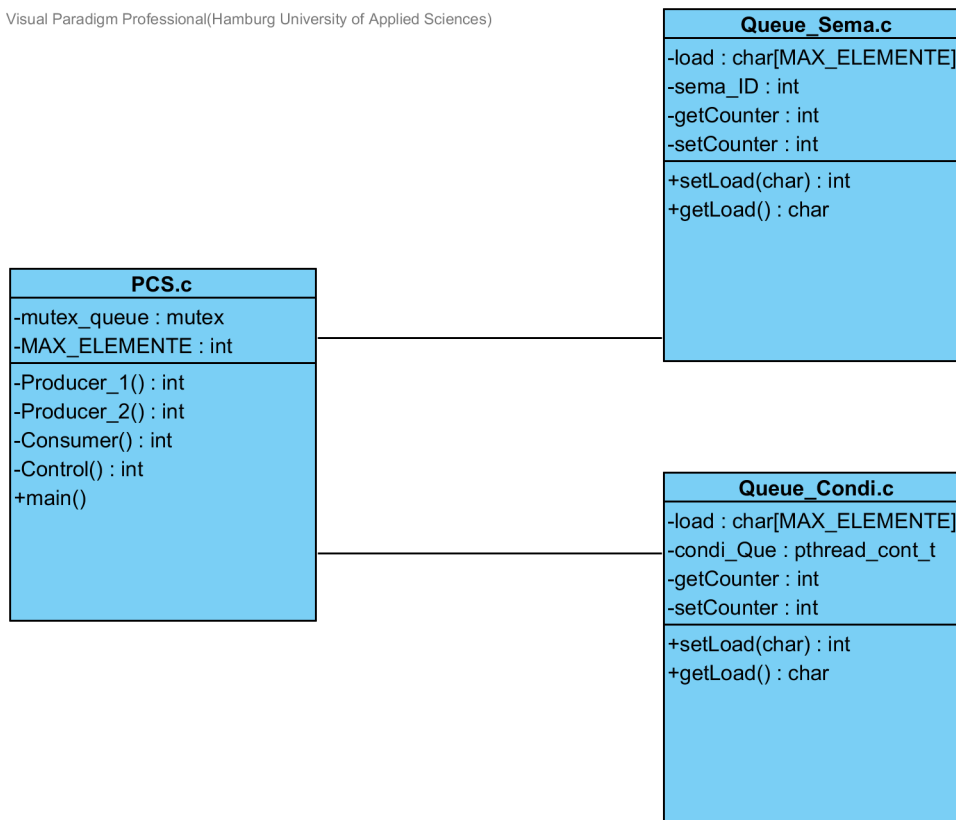
#### I.1 Tastatureingaben:

- '1' → Start/Stopp Producer\_1
- '2' → Start/Stopp Producer\_2
- 'c' oder 'C' → Start/Stopp Consumer
- 'q' oder 'Q' → Main-Thread beendet das System
- 'h' → hilfe

### II. Weitere Anforderungen:

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	
Beenden von Threads nicht in kritischen Abschnitten	//Zwei Möglichkeiten!!
Mutexe+Semaphore am Ende wieder an BS zurückgeben	destroy-Funktionen (vorher unlock beachten)
Funktionen möglichst generisch	

Visual Paradigm Professional(Hamburg University of Applied Sciences)



v1-0.png