Aufgabe 4: jbuffer (12.0 Punkte)

Programmieren Sie ein Modul *jbuffer* (**j**ob **b**uffer), das `int`-Werte in einem *Bounded Buffer* verwaltet. Der Puffer soll für einen Schreiber und mehrere konkurrierende Leser ausgelegt sein.
Der Puffer soll in der nächsten Aufgabe eingesetzt werden, um Aufträge (Netzwerkverbindungen), die von einem Hauptthread angenommen werden, zur Abarbeitung an mehrere Arbeiter-Threads zu verteilen.

a) Ringpuffer-Modul

Implementieren Sie einen Ringpuffer im Modul *jbuffer*, der für einen einzelnen Produzententhread und mehrere Konsumententhreads konzipiert ist. Verwenden Sie die vorgegebene Semaphor-Implementierung (`sem.o`, `sem.h`) aus dem Verzeichnis `/proj/i4sp2/pub/aufgabe4` zur Vermeidung von Über- und Unterlaufsituationen, so dass Produzent bzw. Konsumenten beim Einfügen in einen vollen bzw. bei der Entnahme aus einem leeren Puffer blockieren. Benutzen Sie zur nichtblockierenden Koordinierung der Konsumenten die *CAS*-Funktion `__sync_bool_compare_and_swap()`, so dass mehrere Konsumenten gleichzeitig den kritischen Abschnitt durchlaufen können (keine Locks!).

b) Semaphore

Programmieren Sie nun selbst auf der Basis von Mutex- (**pthread_mutex_init(3)**) und Condition-Variablen (**pthread_cond_init(3)**) einen zählenden Semaphor. Die Schnittstelle des Moduls und passende Funktionsrümpfe finden Sie im Vorgabeverzeichnis. Kopieren Sie die Datei `sem.c` in Ihr Arbeitsverzeichnis und implementieren Sie die Funktionen.

c) Makefile

Erstellen Sie ein zu der Aufgabe passendes Makefile, welches das Testprogramm `jbuffer-test` aus der Quelldatei `/proj/i4sp2/pub/aufgabe4/jbuffer-test.c` erzeugt. Das Makefile soll die Standard-Targets `all` und `clean` enthalten. Bei Änderungen an einer Quelldatei soll sichergestellt sein, dass nur die benötigten Module neu übersetzt werden.

d) Dokumentation

Beschreiben Sie in einer Datei `jbuffer.txt`, welche Koordinierungsprobleme in der Aufgabe auftreten und wie Sie diese jeweils mit welchen Hilfsmitteln synchronisieren.

Hinweise zur Aufgabe:

- Erforderliche Dateien: `Makefile, jbuffer.c, jbuffer.txt, sem.c`
- Die Pthread-Funktionen sind in einer speziellen Funktionsbibliothek (libpthread) zusammengefasst, die Sie beim Kompilieren bzw. Binden Ihres Programms mit angeben müssen (Option `-pthread`).
- `__sync_bool_compare_and_swap()` ist eine sogenannte Builtin-Funktion des GCC, die vom Compiler auf entsprechende atomare CPU-Instruktionen abgebildet wird. Die Dokumentation finden Sie unter `http://gcc.gnu.org/onlinedocs/gcc-4.4.5/gcc/Atomic-Builtins.html`.

Hinweise zur Abgabe:

Bearbeitung: Zweiergruppen
Bearbeitungszeit: 6 Werktage
Abgabezeit: 17:30 Uhr