

Aufgabe 4: jbuffer (12.0 Punkte)

Programmieren Sie eine Bibliothek *jbuffer* (**j**ob **b**uffer), die `int`-Werte in einem *Bounded Buffer* verwaltet. Der Puffer soll für einen Schreiber und mehrere konkurrierende Leser ausgelegt sein.

Der Puffer wird 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 im Modul `jbuffer.c` einen Ringpuffer, 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 beziehungsweise Konsumenten beim Einfügen in einen vollen beziehungsweise bei der Entnahme aus einem leeren Puffer blockieren.

Die Konsumenten sollen untereinander nicht-blockierend koordiniert werden. Benutzen Sie hierfür die CAS-Funktion `__sync_bool_compare_and_swap()`, so dass mehrere Konsumenten gleichzeitig den kritischen Abschnitt durchlaufen können (keine Locks!).

b) Semaphor-Modul

Programmieren Sie nun selbst auf der Basis von Mutex- (`pthread_mutex_init(3)`) und Condition-Variablen (`pthread_cond_init(3)`) einen zählenden Semaphor in der Datei `sem.c`. Die Schnittstelle des Moduls finden Sie im Vorgabeverzeichnis.

c) Makefile

Erstellen Sie ein zu der Aufgabe passendes Makefile, welches das Ringpuffer- und das Semaphor-Modul zu einer statischen Bibliothek `libjbuffer.a` zusammenbindet. Unter Verwendung dieser Bibliothek soll zusätzlich das Testprogramm `jbuffer-test` aus der vorgegebenen Quelldatei `jbuffer-test.c` erzeugt werden.

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 bei der Implementierung des Ringpuffer-Moduls (Teilaufgabe a) 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