

Aufgabe K16: Client/Server-Kanal

Ein spezieller Kanal für synchrone Aufträge an Server bietet folgende Operationen an:

callSyn: Ein Client sendet einen Auftrag und wartet auf ein Ergebnis.

takeSyn: Ein Server nimmt einen Auftrag entgegen.

replyAsyn: Ein Server beantwortet einen Auftrag.

Der Kanal verwendet ausschließlich die Referenzablage. Beim Zustellen eines Auftrags an einen Server generiert er eine Auftragskennung. Diese Kennung ist Rückgabewert von **takeSyn** und Argument von **replyAsyn**. Die Implementierung verwendet folgende Datenstrukturen:

CmA: Wartemenge für Clients mit Auftrag. Initial leer.

Enthält Tupel aus (Client, Nachrichtenreferenz, Antwortpufferreferenz).

SoA: Wartemenge für Server ohne Auftrag. Initial leer.

Enthält Tupel aus (Server, Auftragspufferreferenz).

CoE: Wartemenge bzw. Tabelle für Clients ohne Ergebnis. Initial leer.

Enthält Tupel aus (Kennung, Client, Antwortpufferreferenz).

In CoE speichert der Kanal, welcher Client auf das Ergebnis für welche Kennung wartet. Der Zugriff auf die Elemente erfolgt über die Kennung.

Diese Art von Kanal gibt es zum Beispiel in QNX Neutrino:

https://www.qnx.com/developers/docs/7.0.0/#com.qnx.doc.neutrino.sys_arch/topic/ipc_sync_messaging.html

- a) Beschreiben Sie den Ablauf der Client-Operation **callSyn**. Argumente sind die Referenzen auf eine Auftragsnachricht und einen Empfangspuffer für die Antwort.

Falls ein Server wartet, wird ihm der Auftrag mit einer generierten Kennung zugestellt und der Client in CoE gesteckt. Ansonsten landet der Client in CmA. In beiden Fällen blockiert also der aufrufende Client.

- Aufrufer, d.h. Client blockieren +1
 - SoA enthält Element? +1
- Ja:
- (Server, Auftragspufferreferenz) aus SoA nehmen +0.7
 - Kennung generieren, als Rückgabewert für Server setzen +1
 - (Kennung, Client, Antwortpufferreferenz) in CoE stecken +1
 - Auftragsnachricht in Auftragspuffer des Servers kopieren +1
 - Server deblockieren +0.3
- Nein:
- (Client, Nachrichtenreferenz, Antwortpufferreferenz) in CmA stecken +1

- b) Beschreiben Sie den Ablauf der Server-Operation **takeSyn**. Argument ist die Referenz auf einen Empfangspuffer für den Auftrag, Rückgabewert ist die Auftragskennung.
-

Falls ein Client in CmA wartet, wird dessen Auftrag abgeholt und eine Kennung dafür generiert. Der Client wandert von CmA nach CoE. Ansonsten landet der Server in SoA.

- CmA enthält Element? +1
- Ja:
 - (Client, Nachrichtenreferenz, Antwortpufferreferenz) aus CmA nehmen +0.5
 - Kennung generieren, als Rückgabewert für Aufrufer setzen +1
 - (Kennung, Client, Antwortpufferreferenz) in CoE stecken +0.5
 - Auftragsnachricht in Empfangspuffer des Aufrufers kopieren +1
- Nein:
 - Aufrufer, d.h. Server blockieren +0.3
 - (Server, Auftragspufferreferenz) in SoA stecken +0.7

*Man könnte die Kennung auch immer in **callSyn** generieren. Dann muss sie aber mit in CmA gespeichert werden, was laut Aufgabenstellung nicht der Fall ist.*

- c) Beschreiben Sie den Ablauf der Server-Operation **replyAsyn**. Argumente sind die Auftragskennung und eine Referenz auf die Antwortnachricht.
-

Bei einem gültigen Aufruf wartet der passende Client bereits in CoE, das Ergebnis kann also direkt zugestellt werden. Falls der Server eine ungültige Kennung übergibt oder den Auftrag bereits vorher beantwortet hat, fehlt der passende Client.

- (*Auftragskennung*, Client, Antwortpufferreferenz) aus CoE nehmen +0.7
 - Kennung nicht gefunden? $\Rightarrow$ Abbruch mit Fehler +1
- Ergebnis in Antwortpuffer des Clients kopieren +1
- Client deblockieren +0.3