

OPERATING SYSTEMS BEISPIEL 3

Aufgabenstellung – storetool

Schreiben Sie eine Store-Tool-Anwendung. Diese Anwendung besteht aus zwei Programmen.

SYNOPSIS

```
storeserver
storetool  <-s key data | -g key | -r key | -q>
```

Die beiden Programme kommunizieren über Shared Memory. Das Store-Tool schickt Kommandos an den Server. Danach wartet es, bis der Server das Kommando abgearbeitet hat, überprüft das Ergebnis und terminiert.

Gespeichert werden Strings, die Sie jeweils auf 64 Zeichen inklusive terminierendem `'\0'` (Konstante definieren!) begrenzen können. Jeder String wird mit einem positiven, ganzzahligen Schlüssel (**key**) assoziiert.

Kommandos

`storetool` akzeptiert bei jedem Aufruf genau eines der folgenden Kommandos:

store	-s key data	Der Server soll die Daten (data) assoziiert mit key speichern. Existiert key bereits, meldet der Server einen Fehler ans Store-Tool. Verwenden Sie zum Speichern eine einfach verkettete Liste.
get	-g key	Das Store-Tool fordert vom Server die mit key assoziierten Daten an. Wenn keine Daten gespeichert sind, soll der Server einen Fehler an das Store-Tool melden. Tritt kein Fehler auf, soll das Store-Tool die Daten auf <i>stdout</i> ausgeben.
remove	-r key	Der Server soll einen Wert aus der Liste löschen. Existiert key nicht, wird an das Store-Tool ein Fehler gemeldet.
quit	-q	Erhält der Server dieses Kommando, soll er sich beenden.

Hinweis

Nicht-kritische Fehlermeldungen (jene, die sich auf die Liste beziehen und vom Server an das Store-Tool gemeldet wurden) sollen vom Store-Tool auf *stderr* ausgegeben werden.

Es ist darauf zu achten, dass mehrere Store-Tools gleichzeitig gestartet werden können. Die anderen Store-Tools müssen in diesem Fall so lange warten, bis das erste das Ergebnis vom Server gelesen hat. Bei dem Kommando *quit* dürfen schon wartende Store-Tools mit einem Fehler terminieren, nicht aber der Server und jenes Store-Tool, das das Kommando sendet.

Anleitung

Schreiben Sie zwei Programme, die die Prozesse mittels einer Client/Server-Struktur realisieren. Achten Sie auf eine saubere Terminierung.

Richtlinien

Bitte beachten Sie auch die *Richtlinien für die Erstellung von C-Programmen* sowie die *Allgemeinen Hinweise zur Beispielgruppe 3* auf der Übungswebsite.