

File-Server nach CRUDL

Projektaufteilung

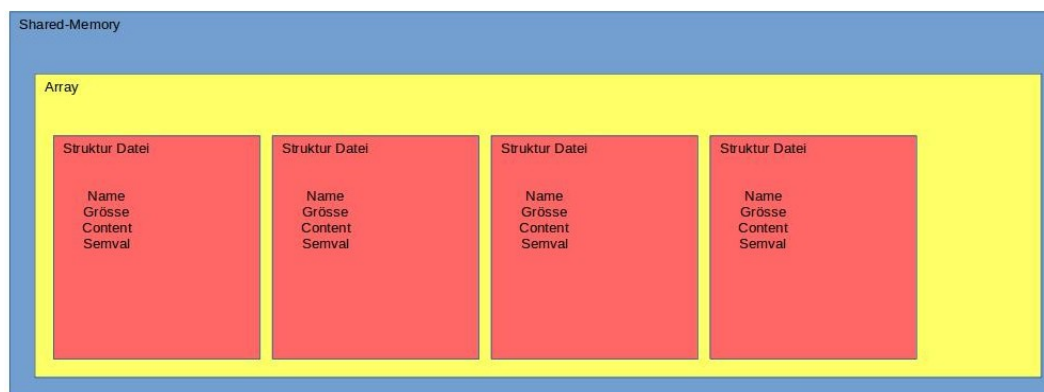
Das Projekt wurde in drei kleinere Probleme unterteilt. Punkt 1 ist der Aufbau des Netzwerksocket, um eine Verbindung zum Server herzustellen. Punkt 2 ist die Erstellung des Shared-Memory, welches die Dateien beinhaltet. Punkt 3 ist der Semaphor, welche die Dateien sperrt. Sobald diese drei Punkte funktionieren, muss nur noch die Logik (Befehle abfangen, Position suchen, usw.) eingefügt werden.

Netzwerkzugriff

Der Server eröffnet einen Server-Socket, welchen er auf alle Adressen abhört. Verbindet nun ein Client, akzeptiert der Server die Verbindung und startet mit diesen Informationen (Adresse des Clients und dessen Port) einen neuen Prozess. Dieser Prozess ist nun ausschliesslich für diesen Client zuständig und bearbeitet die Befehlseingaben.

Shared-Memory

Das Shared-Memory gewährt den Zugriff für verschiedene Prozesse auf die Datei-Struktur. Ist der Speicher reserviert, wird ein Array der Struktur „Datei“ in das Memory gelegt. Die Struktur enthält die Informationen Name, Grösse, Content und Semval. Semval beinhaltet den Wert des Semaphors für die einzelne Datei.



Konzept des Strukturarray im Shared-Memory

Semaphor

Es wird ein Semaphor mit 10 Tickets über alle Prozesse geteilt. So können maximal 10 Clients gleichzeitig auf eine Datei zugreifen. Je nach Anzahl Tickets, die für eine Operation benötigt werden, können mehrere gleichzeitig ausgeführt werden. Ein READ benötigt ein Ticket, während ein UPDATE 10 Tickets benutzt. Die entsprechenden Werte für die einzelnen Dateien sind in der Variable Semval in der Struktur gespeichert.

Herausforderungen

Im ersten Konzept sollte nur ein einziger Semval für alle Dateien benutzt werden. Allerdings hatte dies zur Folge, dass ein UPDATE auf Datei 1 auch ein READ auf eine Datei 2 sperren konnte. Daher musste die Lösung angepasst werden. Innerhalb der Struktur wurde ein Semval jeder Datei zugewiesen. Nun konnte der Wert immer wieder gelesen und zurückgeschrieben werden. Dateien sperrten sich dann nicht mehr gegenseitig. Die Folge davon war, dass der Server für einen Dateizugriff mehr zu tun hatte. Besonders mit dem Auslesen und Zurückschreiben des Semval.

Eine andere Schwierigkeit bestand darin, richtige Stellen für eine neue Datei zu suchen. Ebenso auch die Entfernung und die Freigabe des Platzes. Bei CREATE wird das Array durchsucht bis der Name eine Länge von 0 Zeichen hat. Ist dies der Fall, ist das Ende des Array erreicht und dort wird die neue Datei eingefügt. Wird eine Datei gelöscht, kann nun aber nicht einfach der Name entfernt werden, da sonst das Array nicht mehr bis zum Schluss durchsucht wird. Daher wird der Name auf den Status EMPTY gesetzt. Erstellt man jetzt eine neue Datei, sucht der Server wieder einen Platz. Wenn er einen Eintrag mit dem Status EMPTY gefunden hat, bricht er die Suche ab und benutzt diese Position.