

Aufgabe 2: Nachrichten-basierte Kommunikation

Lernziele Vertiefen der Kenntnisse über TCP-basierte Kommunikation, nebenläufiger Programmierung mittels Threads, Synchronisation und der Programmiersprache C.

Aufgabe Es soll ein Server implementiert werden, der die Anfragen `GET` und `COMPUTE` bearbeitet. Client und Server kommunizieren über TCP-Sockets. Verwendetes Protokoll:

GET fordert den Inhalt der Datei an, die als Parameter übergeben wird (siehe HTTP).
Beispiel: `GET /pfad/text`

COMPUTE berechnet die Formel, die als Parameter angegeben ist. Zur einfachen Berechnung sei die Formel in Postfix-Notation angegeben und auf die arithmetischen Operationen Addition, Subtraktion, Multiplikation und Division beschränkt.
Beispiel: `COMPUTE 3 7 + 5 * 2 -` entspricht $(3 + 7) * 5 - 2$

In beiden Fällen wird ein Strom von Zeichen (`char`) vom Server als Antwort gesendet.

Implementieren Sie einen Dispatcher, der die Anfrage auswertet und den entsprechenden Dienst als eigenen Thread startet. Soll der Server weitere Dienste unterstützen, muss nur der Dispatcher erweitert werden und der entsprechende Dienst implementiert werden (modularer Aufbau).

Jede Anfrage wird in einem eigenen Thread bearbeitet, damit der Server unmittelbar auf weitere Anfragen reagieren kann. Damit nicht bei jeder Anfrage ein Thread erzeugt werden muss, stellen Sie einen Pool von Threads bereit, aus dem bei einer Anfrage ein Thread ausgewählt wird.

Schreiben Sie als Client ein Testprogramm, das mehrere Threads erzeugt, die mit dem Server kommunizieren. Starten Sie mehrere Instanzen des Testprogramms, so dass gleichzeitig mehr Anfragen beim Server eintreffen, als dessen Thread-Pool groß ist.

Testat Zum Testat müssen Sie obige Programme vorlegen und erklären können.