

## Aufgabenblatt 3

(Zu bearbeiten bis 13.11.)

## Vorlesung Verteilte Systeme Wintersemester 2025/26

### Aufgabe 1: RPC und Unions

Die Programmiersprache C unterstützt eine Konstruktion, die als Union bezeichnet wird. Dabei kann ein Feld eines Datensatzes eine von mehreren Alternativen enthalten, wobei die Daten in dem Feld für jede Alternative einen unterschiedlichen Typ besitzen können. Es gibt jedoch keine wirklich sichere Methode, zur Laufzeit zu prüfen, welche Variante in einem Feld enthalten sind, d.h., welchen Datentyp das Feld gerade hat. Haben Unions irgendwelche Auswirkungen auf entfernte Prozeduraufrufe und wenn ja, welche?

### Aufgabe 2: Entfernte Objektreferenzen

Nennen Sie eine Beispiel-Implementierung für eine entfernte Objektreferenz, also ein Objekt, das alle Daten beinhaltet, die ein Client benötigt, um mit einem entfernten Objekt zu interagieren.

### Aufgabe 3: Exceptions

Java und andere Programmiersprachen unterstützen Exceptions, die geworfen werden, wenn ein Fehler auftritt. Erläutern Sie, wie Sie Ausnahmen in RPCs und RMIs implementieren würden und welche Besonderheiten beachtet werden müssen.

### Aufgabe 4: Threads (**Pflichtaufgabe, Abgabe über moodle!**)

Ein Client nimmt entfernte Prozeduraufrufe für einen Server vor. Der Client benötigt fünf Millisekunden, um die Argumente für eine Anfrage zu berechnen und der Server benötigt zehn Millisekunden, um eine Anfrage zu verarbeiten. Die lokale Betriebssystemverarbeitungszeit für jede Send- oder Empfangsoperation beträgt 0,5 Millisekunden und die Netzwerkzeit für die Übertragung der Anfrage- oder Antwortnachrichten beträgt drei Millisekunden. Das Marshalling und das Unmarshalling benötigen 0,5 Millisekunden pro Nachricht. Berechnen Sie die Zeit, die der Client benötigt, um zwei Anfragen zu erzeugen und bis dieser die Antworten wieder erhält:

- a) wenn der Client single-threaded ist und
- b) wenn der Client zwei Threads hat, die die Anforderungen auf einem einzigen Prozessor nebenläufig ausführen können.

Zeiten für die Kontextumschaltung können Sie ignorieren.

### Aufgabe 5: RMI- Grundlagen und Architektur (**Pflichtaufgabe, Abgabe über moodle!**)

Mit dem Java *Remote Method Invocation* (RMI) ist es möglich, Methoden von entfernten Objekten aufzurufen und zu nutzen. Dabei liegt RMI RPC (Remote Procedure Call) zu Grunde sowie eine Drei-Schichten-Architektur.

- a) Nennen Sie die Unterschiede von RMI zu RPC.

- b) Zählen Sie die drei verschiedenen Schichten von RMI auf und beschreiben Sie kurz die jeweilige Funktionalität.
- c) Erläutern Sie kurz, worum es sich bei einem Proxy-Objekt handelt.
- d) Was bedeutet in diesem Zusammenhang Reflection?

### **Aufgabe 6: Programmierung: Java-RMI - Hello World!**

In der Vorlesung wurde Ihnen gezeigt, wie Sie ein erstes Hello-World-Programm mit Java RMI umsetzen können. Vollziehen Sie die dazu notwendigen Schritte nun selbst nach: Implementieren Sie

- a) die Schnittstelle für das Remote-Objekt,
- b) die Implementierungsklasse für das Remote-Objekt,
- c) eine Server-Anwendung zur Aufnahme des Remote-Objekts,
- d) eine Client-Anwendung mit Aufrufen des Remote-Objekts.

Zuletzt übersetzen und starten Sie das System. Um den Namensdienst zu starten, rufen Sie `rmiregistry` mit dem Standardport 1099 auf, danach starten Sie den Server und den Client in zwei Konsolenfenstern.