

Übungsblatt 1

zur Vorlesung

Verteilte Systeme/Ubiquitous Computing

Wintersemester 2007/2008

Achtung: Bitte beachten Sie auch die Web-Site der Vorlesung:
<http://www.mobile.ifi.lmu.de/Vorlesungen/ws0708/vs/>

Wiederholen: Kapitel 11 "Verteilte Systeme" aus dem Skript zu Informatik III

Aufgabe 1: (H) Grundlagen Verteilter Systeme

- Welche spezifischen Eigenschaften besitzen Verteilte Systeme?
- In welche Klassen lassen sich Verteilte Systeme einteilen?
- Welche Vor- und Nachteile haben Verteilte Systeme gegenüber zentralen Systemen?

Aufgabe 2: (T) Mobile Endgeräte

- Aktuell kommen die folgenden Betriebssysteme bzw. Plattformen in mobilen Endgeräten zum Einsatz: Brew, Symbian OS, Windows Mobile, Palm OS, diverse Linux Derivate, J2ME und .NET
Nennen Sie deren Hauptcharakteristika und vergleichen Sie diese miteinander.
- Welche Trends bezüglich Hardwarekomponenten und Anwendungen lassen sich gegenwärtig erkennen?

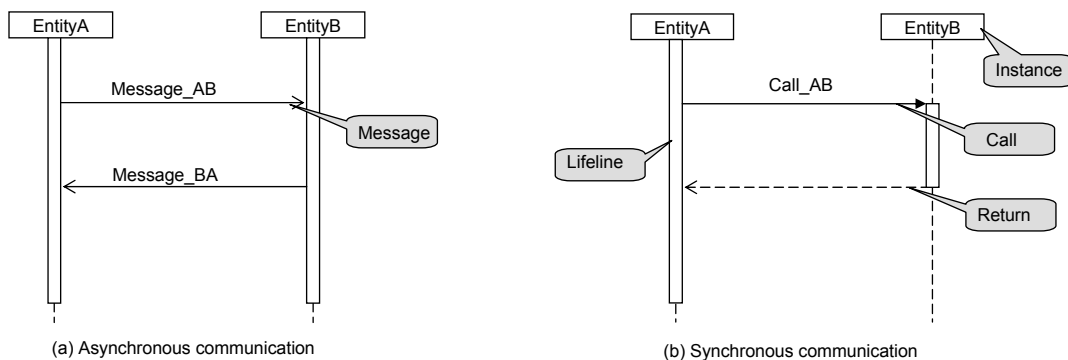
Aufgabe 3: (H) Client/Server-Modell

Das Client/Server-Modell (C/S-Modell) ist eines der Grundmodelle der Verteilten Systeme. Es basiert auf einem einfachen, verbindungslosen Anfrage-Antwort-Protokoll. In der Info III Vorlesung wurden bezüglich der Adressierung, Blockierung, Pufferung und Zuverlässigkeit unterschiedliche Entwurfsmöglichkeiten dargestellt.

- Erläutern Sie die Funktionsweise der verschiedenen Varianten der Adressfindung.
- Erklären Sie den Unterschied zwischen blockierenden (synchronen) und nichtblockierenden (asynchronen) Primitiven mit Hilfe von Sequenz-Diagrammen anhand des folgenden Szenarios:

Ein Benutzer erfragt bei einem Zeit-Server die aktuelle Zeit und wendet sich anschliessend an einen Wetterdienst-Server, um Informationen über das derzeitige Wetter zu erhalten. Wie sieht der jeweilige Ablauf aus, wenn synchrone oder asynchrone Kommunikations-Primitive für die Anfrage benutzt werden?

Bitte verwenden Sie die unten angegebene Notation für die Primitive.



- Ein Entwickler hat die Möglichkeit, Kommunikationsprimitive wahlweise blockierend oder nichtblockierend und gepuffert bzw. nichtgepuffert zu gestalten. Daraus ergeben sich die folgenden vier Möglichkeiten:

	gepuffert	ungepuffert
blockierend	1	3
nichtblockierend	2	4

Welche Kombinationsmöglichkeiten würden Sie implementierungstechnisch nicht empfehlen? Bitte begründen Sie Ihre Antwort.