

Übungsblatt WT:I, WT:II

Bis zum 23.04.2013 sind Lösungen zu folgenden Aufgaben abzugeben: 2, 4, 8, 9, 10.

Aufgabe 1 : Web-basierte Informationssysteme

- (a) Informationssysteme sind offen, dynamisch, komplex. Erläutern Sie diesen Sachverhalt anhand eines Ihnen bekannten Web-basierten Informationssystems.
- (b) Nennen Sie 2 Evolutionsstufen Web-basierter Informationssysteme und jeweils ein Beispiel.

Aufgabe 2 : Eclipse und CVS

Besuchen Sie die Seite <http://www.eclipse.org> und laden Sie sich die Eclipse IDE für Java Entwickler herunter. Zum Installieren des Programms genügt es, das Archiv in Ihren Programmordner zu entpacken. Starten Sie `eclipse.exe`. Falls Sie noch keine Java Runtime Environment (JRE) installiert haben, finden Sie das Installationspaket auf den Download-Seiten von [Oracle](#).

- (a) Lösen Sie anhand des [Eclipse Java-Tutorials](#) folgende Aufgabe: Erstellen Sie ein Java-Projekt, dessen Name Ihrer Matrikelnummer entspricht. Das Projekt soll einen eigenen Ordner `/src` für Java-Quellcode enthalten.
- (b) Schreiben Sie im Ordner `/src` ein Java-Programm, das eine URL entgegennimmt und die zugehörige IP-Adresse ausgibt. Wie lautet die IP-Adresse von `www.uni-weimar.de`?
Tipp: Hilfe beim Schreiben von Java-Programmen bietet das Buch [Java ist auch eine Insel](#).
- (c) Informieren Sie sich über den Zweck und die Funktionsweise des [Concurrent Versions System \(CVS\)](#). Fügen Sie das zuvor angelegte Projekt mit dem Java-Programm in das folgende CVS-Repository ein:

Host:	<code>webis50.medien.uni-weimar.de</code>
Repository Path:	<code>/home/webtec13/cvsroot</code>
User:	<code>webtec13</code>
Password:	<code>webtec2013</code>
Connection Type:	<code>extssh</code>

Befolgen Sie dazu die Schritte des Eclipse Tutorials zum [Einrichten eines CVS-Repositories](#) und [Einchecken eines Projektes](#).

Aufgabe 3 : Verteilte Systeme

- (a) Nennen Sie ein konkretes Anwendungsbeispiel für eine 3-Tier-Architektur. Erläutern Sie anhand eines UML-Sequenzdiagramms [vgl. [Folie](#)] den prinzipiellen Ablauf in Ihrem Beispiels.
- (b) Nennen Sie drei Herausforderungen an verteilte Systeme und geben Sie jeweils ein konkretes Problem an.

Aufgabe 4 : Verbindungsorientierte und verbindungslose Dienste

- (a) Was ist der Unterschied zwischen Vermittlungsart und Verbindungsart (Dienstparadigma)?
- (b) Geben Sie je 2 verbindungslose und verbindungsorientierte Dienste an und erläutern Sie den Unterschied dieser beiden Paradigmen.
- (c) Informieren Sie sich über den Begriff „Quality of Service“. Erläutern Sie in diesem Zusammenhang, wieso auch bei verbindungsorientierter Kommunikation über paketvermittelnde Netzwerke die Garantie der Dienstgüte problematisch ist.

Aufgabe 5 : Netzsoftware

- (a) Welche Aufgaben hat die Netzsoftware in Rechnernetzen?
Welche Rollen spielen Dienste und Protokolle?
- (b) Ordnen Sie die Schichten des TCP/IP-Modells denen des ISO/OSI-Modells zu.

Aufgabe 6 : IP-Protokoll

Informieren Sie sich über das IP-Protokoll und lösen Sie die folgenden Aufgaben.

- (a) Beschreiben Sie den prinzipiellen Aufbau eines IP-Paketes. Suchen Sie dafür nach der RFC 791.
- (b) In welchem Zusammenhang stehen IP und ICMP?
- (c) Installieren Sie das Programm [Wireshark](#) und verschaffen Sie sich damit einen Überblick über den Netzwerk-Traffic auf Ihrem Rechner.

Aufgabe 7 : Client-Server-Interaktionsmodell und Uniform Resource Locator

- (a) Ein Server bietet verschiedene Dienste an (z.B. Auslieferung von Webseiten und Datenbankzugriff), die in unterschiedlichen Prozessen laufen.
Wie kann jeder Dienst eindeutig über das Internet adressiert werden? Auf welchen Schichten des TCP/IP-Modells werden die Adressbestandteile bereitgestellt?
- (b) Wofür werden Ports bei der Rechnerkommunikation benötigt? Warum reicht nicht die IP-Adresse für die Kommunikation aus?

Aufgabe 8 : HTTP

Informieren Sie sich über das HTTP-Protokoll.

- (a) Aus welchen Elementen sind HTTP-Request-Messages im Allgemeinen aufgebaut (genauer Aufbau)? Wie unterscheidet sich ihr Aufbau zu dem von HTTP-Response-Messages?
- (b) Nennen Sie zwei HTTP-Request-Methoden und erklären Sie diese kurz.
- (c) Besuchen Sie die Internetseite <http://web-sniffer.net/> und lassen Sie sich die HTTP-Header für folgende Internetadressen mit den jeweiligen Methoden anzeigen:
 - (c1) <http://www.uni-paderborn.de> mit der Methode GET

- (c2) <http://www.uni-paderborn.de> mit der Methode TRACE
- (c3) <http://www.uni-weimar.de/medien/webis/tmp/ss13/ex-np011/test.html> mit der Methode GET.
- (c4) <http://www.uni-weimar.de/medien/webis/tmp/ss13/ex-np011/test1.html> mit der Methode GET.
- (c5) <http://www.uni-weimar.de/medien/webis/tmp/ss13/ex-np011/forbidden/test.html> mit der Methode GET.

Notieren Sie sich die entsprechenden Ergebnisheader und erläutern Sie diese. Ordnen Sie die Zeilen im Header den Gruppen Request-/Response-Header, Entity-Header und General-Header zu [RFC 1945].

Aufgabe 9 : SMTP

Informieren Sie sich über das Simple Mail Transfer Protocol (SMTP). Gute Quellen hierfür sind die [RFC 821](#) und Wikipedia.

- (a) Ist SMTP, genauso wie HTTP, ein zustandsloses Protokoll? Begründen Sie Ihre Antwort.
- (b) Beschreiben Sie mithilfe eines UML-Sequenzdiagramms das erfolgreiche Versenden einer Nachricht mit dem SMTP-Protokoll.
- (c) Beschreiben Sie grafisch den Aufbau einer Textnachricht nach [RFC 822](#). Orientieren Sie sich dabei an der grafischen Notation der [HTTP-Request-Message](#).
- (d) Verbinden Sie sich mittels des `telnet`-Kommandos mit `mailgate.uni-weimar.de` auf Port 25. Versenden Sie eine Mail an sich selbst mit Absender `joachim.gauck@bundespraesident.de` und Betreff „Einladung zur Verleihung des Verdienstkreuzes 1. Klasse“, in der Sie benachrichtigt werden, dass Ihnen das Verdienstkreuz erster Klasse verliehen werden soll. Dokumentieren Sie Ihr Vorgehen, indem Sie Ihre Befehle sowie die Antworten des Mailservers kopieren und als Lösung abgeben. (Hinweis: Sie können sich nur von einem Rechner innerhalb der Uni, z. B. einem Pool-Rechner, mit dem Mailserver an Port 25 verbinden.)
- (e) Warum lässt sich der Mailserver der Uni Weimar nicht zum Versenden von anonymen SPAM-Nachrichten missbrauchen?

Aufgabe 10 : Programmieraufgabe: HTTP File Server P

Schreiben Sie zunächst in Java ein Server-Programm, das die GET-Methode des HTTP-1.1-Protokolls implementiert. Das Programm nimmt auf einem Socket GET-Anfragen mit einem Dateipfad entgegen und antwortet protokollgemäß mit dem Inhalt der Datei. Eine Anfrage nach einem Verzeichnispfad wird mit einer Auflistung der enthaltenen Dateien beantwortet. Fehlerfälle (wie eine nicht vorhandene Datei) werden protokollgemäß behandelt. Verwenden Sie angemessene HTTP-Header und Statuszeilen. Testen Sie Ihr Programm mit einem `telnet`-client und im Web-Browser.

Hinweise:

- Verwenden Sie das in der Datei `FileServerTemplate.java` vorgegebene Programmgerüst.

P Diese Übungsaufgabe ist Teil eines Programmierprojektes, das dazu dient, das Zusammenspiel verschiedener Webtechnologien zu veranschaulichen. Zu diesem Zweck werden im Rahmen der Übungen verschiedene, aufeinander aufbauende Komponenten implementiert. Bis zum Ende des Semesters soll ein einfacher RSS-Reader mit HTML-Oberfläche entstehen.

Aufgabe 11 : Programmieraufgabe: Chat-Server

Entwickeln Sie einen kleinen Chat-Server mit Java-`Socket`s. Beachten Sie, dass der Chat-Server jede Verbindung in einem eigenen Thread verwalten muss um mehrere Clients parallel bedienen zu können. Der Server soll alle aktiven Client-Verbindungen in einem Set verwalten. Sendet ein Client eine Nachricht, wird sie vom Server an alle Clients im Set weitergeleitet (Broadcast). Achten Sie darauf, dass der Zugriff auf das Set mit Client-Verbindungen *thread-safe* ist. Beendet ein Client die Verbindung, soll dieser Client aus dem Set entfernt werden.