

Verteilte Systeme

Theoretische Übungen

Bemerkungen

Diese Übungen dienen zur Vorbereitung auf die schriftliche Prüfung. Sie sind freiwillig und werden nicht korrigiert. Wenn Sie Fragen oder Bemerkungen zu den schriftlichen Übungen haben, wenden Sie sich bitte an einen der verantwortlichen Assistenten.

Matthias Kovatsch (kovatsch@inf.ethz.ch)

1 Verteilte Systeme

Beschreiben Sie ein konzeptionelles Problem, das nur in verteilten Systemen auftreten kann.

2 Fehlermodelle

Im Abschnitt “Kommunikation” der Vorlesung werden verschiedene Fehlermodelle beschrieben (fehlerhaftes Senden, Empfangen, Übertragen, Crash, Fail-Stop, Zeitfehler, Byzantinische Fehler).

Durch welche Fehlermodelle werden die folgenden Anwendungsfälle am besten charakterisiert? Geben Sie auch jeweils an, wen oder was Sie unter einer Nachricht und dem Empfänger bzw. Sender einer Nachricht verstehen.

1. Bei der Anfrage an einen Webserver wird ein Dokument (HTML-Seite) nicht gefunden.
2. Die Batterie eines GSM-Telefons ist leer.
3. Auf einem Rechner, der an einem Peer-to-Peer-Netz teilnimmt, hat sich ein spezialisierter Virus eingenistet, der den P2P-Verkehr beobachtet und bestimmte Zugriffe sperrt.
4. Die WLAN-Verbindung eines Laptops ist instabil und bricht immer wieder für kurze Zeit ab.
5. Wegen Überlastung eines Mailservers kommen wichtige E-Mails verspätet beim Empfänger an.
6. Der Spam-Filter eines E-Mail-Clients verschiebt wichtige E-Mails in einen Spam-Ordner, wo sie der Benutzer übersieht.
7. Ein Drucker druckt den Text von Postscript-Dateien aus, statt den Postscript-Code zu interpretieren.

3 Kommunikation

1. Wie kann es bei synchroner Kommunikation zwischen zwei Prozessen zu einem Deadlock kommen?
2. Bei welchem Kommunikationsmechanismus besteht nur eine geringe Gefahr für Deadlocks? Begründen Sie Ihre Antwort.
3. Warum bevorzugen Programmierer trotzdem RPC?
4. Was ist der Unterschied zwischen synchroner, mitteilungsbasierter und synchroner, auftragsorientierter Kommunikation ohne Rückgabewert?

4 RPC

1. Ist es möglich, bei RPC-Aufrufen einen Zeiger als Eingabeparameter zu verwenden (“call by reference”)? Als Ausgabeparameter? Begründung!
2. Wenn ein Client zu seiner Anfrage nach einem gewissen Timeout keine Bestätigung erhält, wird er die Anfrage wiederholen. Es könnte aber nur die Bestätigungsnachricht verlorengegangen sein, obwohl der Server die Anfrage bearbeitet hat. Welche Gefahr besteht hierbei und wie könnte man ihr begegnen?
3. Mit welcher RPC-Fehlersemantik-Klasse würden Sie das Verhalten eines Paares Websurfer/Webserver beschreiben, wenn der Websurfer eine GET-Anfrage (Lesen einer Webseite) stellt und, wenn nichts angezeigt wird, den “Reload”-Knopf des Browsers drückt, bis die gewünschte HTML-Seite erscheint?

5 Synchrone Kommunikation mit einem Mars-Rover?

Ein Gefährt wird zum Mars geschickt, das mit einer Kamera ausgestattet ist und von der Erde aus gesteuert werden kann. Die Entfernung zwischen Erde und Mars betrage 55.7 Mio km (kleinster Abstand zwischen Erde und Mars). Es steht eine Uplink-Verbindung (von der Erde zum Mars) mit einer Bandbreite von 100 bit/s zur Verfügung, sowie eine Downlink-Verbindung (vom Mars zur Erde), die pro Sekunde ein Bild liefert. Mit Steuerbefehlen der Länge 10 Bit soll das Gefährt so gelenkt werden, dass es Hindernissen ausweicht.

Ein Hindernis ist auf den Bildern als solches zu erkennen, wenn es maximal 10 m entfernt ist. Die Reaktionszeit des Steuermanns auf der Erde beträgt 4 s. Auf einen Steuerbefehl reagiert das Vehikel sofort, sobald es ihn empfangen hat.

Wie schnell darf das Gefährt maximal fahren, damit einem Hinderniss zuverlässig ausgewichen werden kann?

6 Modellierung von Web-Schnittstellen

In der Vorlesung wurden zwei verschiedene Paradigmen vorgestellt, nach denen man Web-Schnittstellen modellieren kann: serviceorientiert mittels WS-* und ressourcenorientiert direkt mittels HTTP 1.1. In dieser Aufgabe sollen Sie für beide Paradigmen die Web-Schnittstelle eines internetfähigen digitalen Bilderrahmens modellieren. Der digitale Bilderrahmen kann auf seinem Display sowohl Text als auch Bilder darstellen. Ausserdem unterstützt er die Darstellung von Diashows im Atom¹- und RSS²-Format. Die Darstellung von Slideshows kann weiter konfiguriert werden. So kann man die Darstellungsdauer eines Bildes festlegen, ebenso wie die Darstellungsreihenfolge der Bilder (zufällig/ursprüngliche Reihenfolge). Der Rahmen verfügt ausserdem über drei verschiedene Übergangsmodi zwischen den einzelnen Bildern. Das Display selbst kann in 16 verschiedenen Helligkeitsstufen gedimmt werden. Weiterhin sind im Bilderrahmen noch ein binärer Bewegungsmelder, welcher die Anwesenheit von Personen feststellen kann, und ein Helligkeitssensor, welcher in 256 verschiedenen Stufen auflöst, verbaut.

Modellieren Sie nun für beide Paradigmen die entsprechenden Schnittstellen, welche die nachfolgend beschriebene Funktionalität zur Verfügung stellen:

1. Darstellung eines Textes auf dem Bildschirm (unformatierter Text)
2. Darstellung eines JPEG-Bildes auf dem Bildschirm
3. Darstellung einer Diashow, welche im Atom- bzw. RSS-Format übertragen wird
4. Abfrage der aktuellen Darstellungskonfiguration
5. Abfrage des momentanen Bildschirminhalts (z.B. gerenderter Text oder aktuelles Bild des Feeds)
6. Setzen/Auslesen der Konfiguration der Diashow
 - Anzeigedauer pro Bild in Sekunden
 - Übergangsanimation zwischen den Bildern (keine, überblenden, umblättern)
 - Reihenfolge der Bilder (wie im Feed spezifiziert oder zufällig)
7. Setzen/Auslesen der Bildschirmhelligkeit
8. Auslesen des Bewegungssensors
9. Auslesen des Helligkeitssensors

Beispiel: Nehmen wir an, dass es eine Funktion gibt, mittels derer man den Bildschirm des Bilderrahmens ein- und ausschalten und dessen aktuellen Zustand (ein- oder ausgeschaltet) abfragen kann. Nachfolgend ein Modellierungsvorschlag für beide Paradigmen.

¹[http://de.wikipedia.org/wiki/Atom_\(Format\)](http://de.wikipedia.org/wiki/Atom_(Format))

²<http://de.wikipedia.org/wiki/RSS>

- Serviceorientiert: (Pseudocode ausreichend)
 - `setDisplayPower([xsd:boolean])`
(`[xsd:boolean]`, `[xsd:integer]` und `[xsd:string]` sind verwendbare Basistypen; wird `[xsd:complexType]` benötigt, muss dieser nicht weiter spezifiziert werden)
 - `[xsd:boolean] getDisplayPower()`
- Ressourcenorientiert:
 - Pfad: `/display/power`
 - Methoden: GET, PUT
 - Repräsentation: `text/plain`
 - Erlaubte Werte:
 - * Display ist eingeschaltet: `true`
 - * Display ist ausgeschaltet: `false`

7 REST

Ist das ressourcenorientierte Beispiel (und Ihre Lösungen) aus Aufgabe 6 bereits RESTful? Geben Sie jeweils ein Beispiel mit Erklärung, welche Eigenschaften von REST von dieser Web-Schnittstelle erfüllt werden und welche nicht.

8 Client/Server

1. Bei Web-basierten Diensten wird oft ein Bezeichner in Links codiert (“URL rewriting”), um die aktuelle Transaktion zu identifizieren. Wie könnte ein Unbefugter eine laufende Transaktion “übernehmen” und was kann man gegen diese Gefahr tun?
2. Welches Problem entsteht bei einem zustandsbehafteten Server, wenn viele Clients abstürzen, bevor sie ihre Transaktionen beendet haben?
3. Erläutern Sie kurz ein paar Vorteile von zustandslosen gegenüber zustandsbehafteten Client/Server-Protokollen und umgekehrt.

9 Jini

1. Erläutern Sie kurz die Funktion von Leases.
2. Eine Besonderheit von Jini ist das Ausnutzen der Mobilität von Java-Code. Welche Code-Teile werden übertragen und welche Möglichkeiten ergeben sich dadurch?

10 Internet Protokolle

Jeder Aufruf eines Web-Services muss mehrere Protokollschichten durchlaufen, damit er dem Server zugestellt werden kann. Bei einem SOAP-Requests mit HTTP-Binding sind das:

- Anwendungsschicht
- WS-*-Transportschicht (HTTP)
- Transportschicht (TCP)
- Vermittlungsschicht (IP)
- (Tiefere Schichten, die hier nicht behandelt werden müssen)

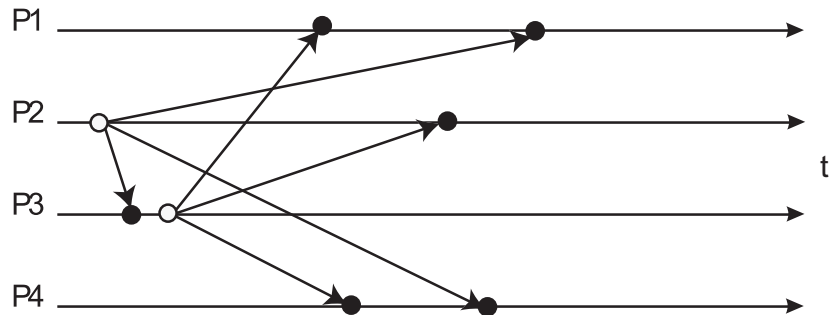
Beschreiben Sie für jede der vier genannten Schichten jeweils

1. in welcher Form der Aufruf verpackt wird,
2. welche Informationen dort jeweils zur Adressierung benötigt werden und
3. woher der Client die jeweilige Information hat.

11 Broadcast

In Abbildung 1 sind zwei Broadcast-Fälle dargestellt.

A



B

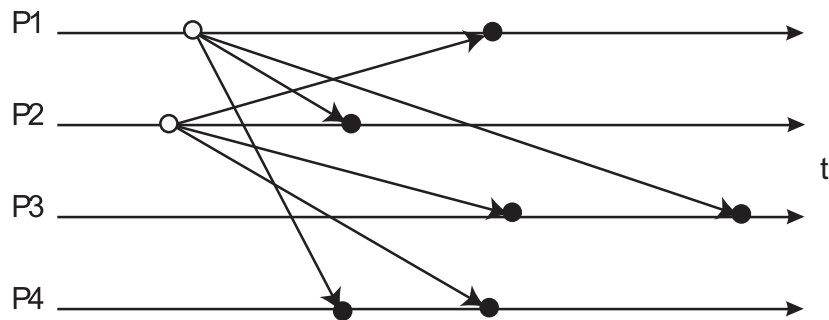


Abbildung 1: Broadcast

1. Welcher der beiden Fälle ist "atomar"?
2. Besteht eine kausale Abhängigkeit zwischen den Broadcasts von P2 und P3 im Fall A?
 Zwischen den Broadcasts von P1 und P2 im Fall B?
3. Sind Broadcasts, die über einen zentralen "Sequencer" gesendet werden, notwendigerweise total geordnet? Welche Voraussetzung muss dazu erfüllt sein?

12 Lamport-Zeit

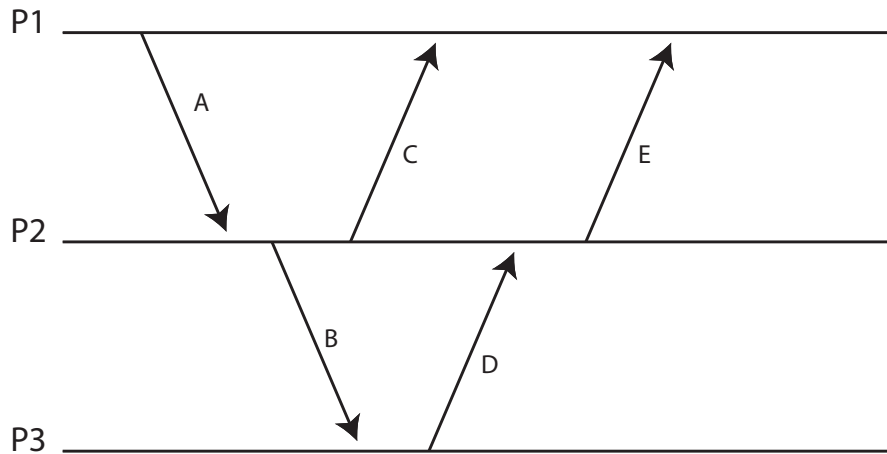


Abbildung 2: Zeitdiagramm

Im folgenden bezeichnet \prec die Kausalrelation auf Ereignissen (“happened before”), C ist die Abbildung von Ereignissen auf Zeitstempel (die durch natürliche Zahlen repräsentiert werden).

1. Geben Sie ein Paar von Ereignissen aus Abb. 2 an, über deren kausale Abhängigkeit keine Aussage getroffen werden kann.
2. Fügen Sie in Abb. 2 eine Nachricht N ein, für die gilt

$$A.\text{receive} \prec N.\text{send} \wedge C(N.\text{receive}) < C(E.\text{send})$$

wobei Sie Absender und Empfänger der Nachricht (die unterschiedlich sein sollen) frei wählen können, sofern die Bedingung erfüllt ist.

3. Fügen Sie auf ähnliche Art eine Nachricht M ein, für die gilt

$$M.\text{send} \prec C.\text{send} \wedge C(B.\text{receive}) < C(M.\text{receive})$$

und die von P2 gesendet und von P3 empfangen wird.

Bemerkung: Die Notation X.send bzw. X.receive bezeichnet das send- bzw. receive-Ereignis der Nachricht X.

13 Lamport-Zeit – Wechselseitiger Ausschluss

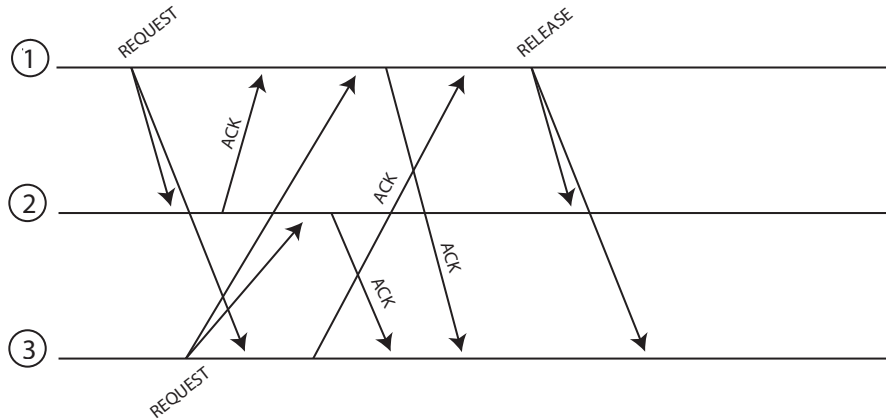


Abbildung 3: Wechselseitiger Ausschluss mit Lamport-Zeit

In Abb. 3 ist ein Zeitdiagramm dargestellt mit Nachrichten von drei Prozessen. Prozesse 1 und 3 bewerben sich um den exklusiven Zugriff auf eine gemeinsame Ressource. Die Prozesse wenden das aus der Vorlesung bekannte Verfahren zum wechselseitigen Ausschluss an, das Lamport-Zeit und verteilte Warteschlangen benutzt.

1. Geben Sie die Sende- und Empfangszeitstempel für jedes Ereignis an.
2. Geben Sie für die Prozesse 1 und 3 an, wie die Warteschlange des jeweiligen Prozesses nach jedem Sende- bzw. Empfangereignis aussieht.
3. Sind beim Einreihen in die Warteschlange der Sende- oder der Empfangszeitstempel zu verwenden? Warum?
4. Welche Bedingung muss erfüllt sein, damit Prozess 1 auf die Ressource zugreifen kann?
5. Markieren Sie den Zeitpunkt im Zeitdiagramm, zu dem Prozess 1 bzw. Prozess 3 auf die Ressource zugreifen kann.

14 Sicherheit

1. Auf welche Herausforderungen trifft man bei der Schlüsselverteilung in verteilten Systemen?
2. Beschreiben Sie zwei mögliche Lösungsansätze zur Schlüsselverteilung.

15 Einwegfunktionen

Mit Einwegfunktionen lassen sich Einmalpasswörter erzeugen und leicht überprüfen. f sei eine Einwegfunktion und x_1 ein initiales Passwort, aus dem eine Passwortkette erzeugt wird:

$$x_1 \xrightarrow{f} x_2 \xrightarrow{f} \dots \xrightarrow{f} x_{n-1} \xrightarrow{f} x_n$$

1. Um die Passwörter zur Authentisierung nutzen zu können, muss x_n zunächst zum Server S übertragen werden. Welche der folgenden Anforderungen müssen erfüllt sein:
 - a) Ein Angreifer darf nichts über x_n erfahren, die Übertragung muss also geheimnisbewahrend erfolgen.
 - b) Es muss sichergestellt sein, dass x_n bei der Übertragung nicht verändert wird.
2. Wir nehmen an, es sei $n = 100$. Dem Server S wird x_{100} bekannt gemacht. Ein Client C schreibt die Werte x_1, x_2, \dots, x_{99} in eine Liste. Bei der ersten Anmeldung an S verwendet er x_{99} und streicht diesen Wert von der Liste. Beim zweiten Mal verwendet C aus Versehen x_{89} (statt x_{98}). Welche Gefahr besteht, wenn dieser Wert von einem Angreifer abgehört wird und S den Anmeldeversuch einfach ignoriert, weil $f(x_{89}) \neq x_{99}$?

16 One-Time-Pads

Wenn One-Time-Pads ein perfektes Verschlüsselungssystem darstellen, warum werden diese dann heutzutage nicht global eingesetzt?

17 Diffie-Hellman

In der Vorlesung wurde der Diffie-Hellman-Algorithmus besprochen.

1. Für was wird er verwendet?
2. Beschreiben Sie kurz das Verfahren.
3. Was ist ein möglicher Angriff und wie könnte man sich dagegen verteidigen?

18 Zertifikate

Wie funktioniert zertifikatsbasierte Authentifizierung? Von welchem weitverbreiteten Protokoll wird sie verwendet?