

K Kommunikation

1. Situation: Teilnehmer Meier meier@ee.fhm.edu verschickt an einen Teilnehmer Huber huber@pruefung.com eine Mail. Meier bedient sich auf seiner Workstation eines Mailclients um die Mail zu erstellen. Er schickt sie mittels des Clients dann sofort an seinen Mailserverrechner mail.ee.fhm.edu mit der Bitte um Weiterleitung. Dieser Mailserverrechner nimmt dann zu einem späteren Zeitpunkt Kontakt mit dem Mailserverrechner pruefung.com auf und übergibt ihm die Nachricht für Teilnehmer huber. Huber fragt zu einem beliebigen späteren Zeitpunkt mittels seines Mailclients seinen Mailserverrechner nach einer neuen Nachricht für sich ab und erhält bei dieser Gelegenheit die neue Mail von Teilnehmer meier zugestellt. Die Zustellung einer Mail zwischen Mailrechnern wird im Fehlerfall mehrfach versucht und nach einer maximalen Anzahl von Fehlversuchen vom Senderrechner abgebrochen. Die tatsächliche Internetadresse des Rechners ee.fhm.edu ist 192.168.1.1 und die des Rechners pruefung.com 192.168.1.2
 - a. Zeigen Sie anhand eines Blockdiagramms die beteiligten Rechenprozesse und kennzeichnen Sie deren Kommunikation. Zeigen Sie auf welche zeitlichen Abhängigkeiten existieren.
 - b. Klassifizieren Sie den Vorgang aus Sicht des Anwenders nach Art der Adressierung, Blockierung, Pufferung und Kommunikationsform.
 - c. Welche Art der Transparenz aus Sicht des Anwenders ist hier realisiert? Was ist wenn der Empfängerrechner ausgetauscht wird oder seinen Ort wechselt? Was ist wenn Huber oder Meier ihren Mailserverrechner wechseln?
 - d. Sie wollen weltweit unter einem Namen erreichbar sein. Was wäre zu tun?
 - e. Wie ist hier die Fehlersemantik aus Sicht des Mail-Senders bezüglich des Empfangens und Lesens der Mail durch den Empfänger zu beurteilen?
 - f. Als Kennung einer jeden Mail dient eine einzigartige Signaturnummer. Wie könnte eine Implementation der exactly-once Fehlersemantik aussehen? Beschreiben Sie die Aktionen der einzelnen Teilnehmer anhand eine Time-Sequenz-Diagramms.
2. Was ist der Unterschied zwischen einer Mailbox-Kommunikation und einer Portbasierenden Kommunikation?
3.
 - a. Nennen Sie ein praktisches Beispiel einer auftragsorientierten Kommunikation.
 - b. Wenn mehrere Client-Tasks existieren, die von nur einer Servertask bedient werden, was passiert mit quasi gleichzeitig eintreffenden Aufträgen verschiedener Clients? Zeigen Sie anhand eines Time-Sequenzdiagramms mit zwei Clienttasks und einer Servertask den Fall quasi gleichzeitig eintreffender Aufträge beim Server. Welche Randbedingung ergibt sich hier für die Serverimplementation damit hier möglichst keine Probleme auftreten?
 - c. Es sei nun die Kommunikation zwischen den Clients und den Servern portbasiert und es gibt die Möglichkeit innerhalb von Tasks Threads zu erzeugen. Skizzieren Sie grob eine mögliche Strategie die Servertask so zu implementieren, daß quasi die parallele Bearbeitung von Clientaufträgen möglich ist.
4. Es soll ein elektronisches Wahlsystem zur Vertiefungsfächerwahl fuer alle 24 Masterstudenten eingerichtet werden. Die Wahl soll von den Masterstudenten an 24 Rechnern zeitgleich ausgeführt werden und ist nur gültig wenn alle nach Aufforderung durch den Wahlkoordinatorrechner abgestimmt haben. Das individuelle Abstimmungsergebnis soll bei Gültigkeit der Wahl auf allen Rechnern sichtbar sein.
 - a. Welche Art der Kommunikation wählen Sie zwischen dem Wahlkoordinatorrechner und den einzelnen Rechnern?
 - b. Welcher Zuverlässigkeitsgrad und welcher Ordnungsgrad ist gefordert?

S Synchronisation

1. Kompensation der Zeitdrift. Sie stellen fest, daß eine Uhr 4 Sekunden vorgeht. Die Uhr soll in 8 Sekunden justiert sein. Geben Sie die lineare Funktion an die dies bewerkstelligt.
2. Geben Sie zu untenstehendem Ablaufplan für jedes Ereignis die jeweilige Vektorzeit an. Markieren Sie, welche Ereignisse in einer *happened-before* Relation stehen und wo keine Angaben möglich sind. Dabei bezeichnen:

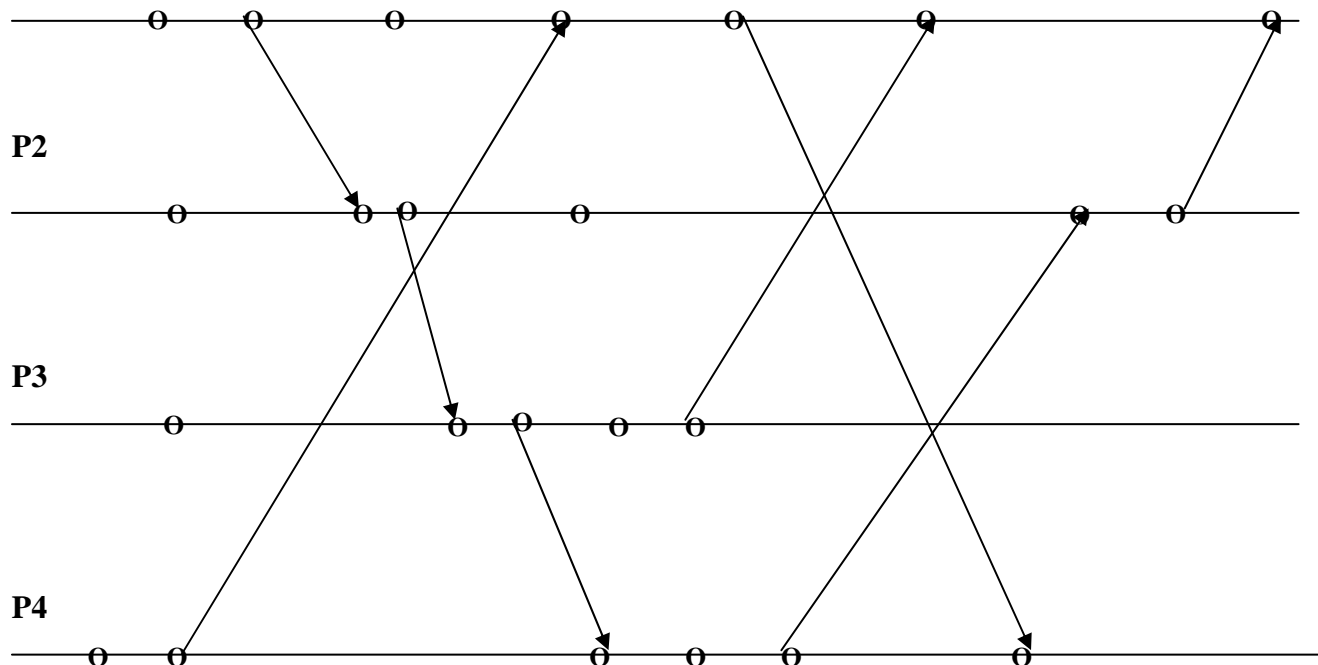
I - die Initialisierung der Zeit auf 0 bzw. den entsprechenden Vektor.

L - ein lokales Ereignis

S - ein Sendeereignis

E - ein Empfangsereignis

P1



Lösungen

K 1a:

K 1b: Es handelt sich hierbei um eine indirekte Adressierung. Der Sender adressiert nicht direkt den Empfangsprozess sondern läßt die Nachricht in eine Mailbox, die vom Mailserver gehalten wird, hinterlegen. Eine direkte Adressierung liegt hier nicht vor, da der lesende Mailclient als Stellvertreter für Huber nicht unmittelbar adressiert wird.

Unter dem Gesichtspunkt der Blockierung liegt eine asynchrone Kommunikation vor. Weder Teilnehmer meier noch huber warten im Normalfall synchron auf Mailaustausch.

Außerdem ergibt sich dadurch, daß die beiden Mailclients nicht immer laufen die Notwendigkeit der Pufferung (sogar Mehrfachpufferung). Die Kommunikation zwischen den Partnerprozessen ist meldungsorientiert. Dabei ist die Kommunikation zwischen den einzelnen Prozessen rendezvous-orientiert. Aus Sicht des Sender Meier/Empfänger bei der Bedienung seines Mailclients ist es auch rendezvous-orientiert wenn man davon ausgeht, daß Meier die Quittung abwartet daß sein Mailserver die Nachricht entgegengenommen hat und daß Huber pollt ob eine Nachricht da ist oder nicht.

K 1c: Die symbolische Mailadresse wird in der realen Interprozeßkommunikation ersetzt werden durch die realen Rechneradressen. Diese Umsetzung geschieht **ortstransparent** für den Maeier/Huber z.B. durch die Mailserver. Die reale Rechneradresse stellt eine Ortsangabe dar. Wenn der Rechner ausgetauscht wird, so passiert keine Veränderung wenn der neue Rechner die Identität des Gewechselten übernimmt. Ansonsten ist im Umsetzer (Namensdienst) Symbolische/Reale-Adressen der Teil der realen Adresse durch die neue zu ersetzen. Dieser Austausch geschieht **migrationstransparent** für die beteiligten Benutzer meier/huber . Das Wechseln der Mailserver bedingt auch eine Änderung der symbolischen Adressen, diese Änderung ist nicht transparent für alle Beteiligten.

K 1d: Eine Idee von vielen: Es gibt eine eindeutige symbolische Adresse für jeden Teilnehmer in einem möglicherweise weltweiten verteilten Verzeichnis. Dort wird dann eine Umsetzung -ähnlich einer Rufumleitung- in eine herkömmliche symbolische Mailadresse erzeugt. Diese wird dann in einem weiteren Umsetzungsvorgang in eine reale Rechneradresse umgesetzt.

K 1e: Da zunächst nichts gesagt wurde, kann man eine **may-be** Semantik vermuten, da der Sender zunächst keine Quittung erhält hat ob die Mail tatsächlich gelesen wurde. Für den Mail-Empfang am Mailrechner ist **zumindest** eine **at-least-once** oder **at-most-once** Fehlersemantik anzunehmen. Es ist nicht ersichtlich ob Auftragsduplikate entfernt werden und ob auch nach Ausfall im Wiederanlaufsfall die Mail-Aufträge noch eindeutig existieren.

K 1f: Der Empfangsmailrechner führt ein persistentes Transaktionsbuch über die bereits korrekt empfangenen Mails mithilfe der Signaturnummern.

Sobald eine Mail doppelt erscheint wird sie z. B. zwar entgegengenommen aber weggeworfen.

K 2. Bei Mailbox-Kommunikation wird von einem Dritten der Mailboxpufferplatz vorgehalten. Damit ist auch über die Laufzeit der Kommunikationspartnerprozesse hinweg die Nachricht gespeichert. Bei Port-Kommunikation müssen beide Partner gleichzeitig laufen.

K 3.a Im Praktikum haben Sie RMI eingesetzt als Beispiel einer auftragsorientierten Kommunikation.

K 3.b Ein Client muß warten. Unklar für den Client nach einem Timeout ist ob die Kommunikation zum Server zusammengebrochen ist oder der Server belegt ist. Der Server darf also für die Auftragsbearbeitung nur soviel Zeit in Anspruch nehmen, so daß die Verzögerung bei gleichzeitiger Clientanfrage für den Client transparent ist und von einer Kommunikationsstörung unterscheidbar ist. Eine andere Lösung ist die Auftragsbearbeitung im Server so zu gestalten, daß der Server trotzdem jederzeit empfangsbereit ist und einen parallel zur Bearbeitung eintreffenden Auftrag mit einer 'busy' Nachricht an den Sender abweist. Der kann dann ggf. später seinen Auftrag wiederholen.

K 3.c Der Server besteht aus i.d.R. aus mehreren Threads. Ein immer laufender Thread ist der Auftragskoordinator. Der lauert auf einem fest vereinbarten Port auf eingehende Clientverbindungen. Sobald ein Verbindungswunsch hereinkommt legt der Auftragskoordinator einen neuen Port an, teilt dem Client mit, den eigentlichen Auftrag an diesen neuen Port zu richten und startet gleichzeitig einen Arbeitsthread. Der Arbeitsthread nimmt an diesem neuen Port den Auftrag vom Client entgegen und bearbeitet ihn. Währenddessen ist der Auftragskoordinatorthread wieder frei und lauert auf weitere Clientanfragen. Ist der Arbeitsthread fertig so schließt er den Port und beendet sich selbst.

K 4.a Gruppenkommunikation, geschlossene Gruppe

K 4.b atomar, denn es sollen alle Teilnehmer wählen oder keiner, entweder haben alle das Ergebnis eines Nachbarn oder keiner und entweder haben alle die Nachricht Wahl gültig oder keiner. Kausale Ordnung reicht, denn der Wahlkoordinator startet die Wahl. Wenn jeder die Wahlaufforderung erhalten hat, dann darf gewählt werden. Aber die Ergebnisse dürfen bei einem Teilnehmer erst sichtbar werden wenn der Wahlkoordinator die Wahl per Nachricht als gültig erklärt hat nachdem er vorher **alle** Einzelabstimmungen erhalten hat.

S 1.

S 2.

P1

