

Bitte beachten Sie, dass *Präsenzaufgaben* zwar nicht schriftlich bearbeitet werden müssen und nicht bewertet werden, jedoch als Vorbereitung für den Übungstermin und für die Klausur notwendig sind.

Die Abgabe der *Pflichtaufgaben* erfolgt in Teams von 3–5 Studierenden online als exakt eine PDF-Datei spätestens bis zum oben genannten Termin. Eine spätere Abgabe ist nicht möglich. Bitte verwenden Sie dabei den Zugriffscode einer vorherigen Abgabe, wenn die Zusammensetzung Ihrer Kleingruppe unverändert geblieben ist. Achten Sie in jedem Fall auf die gleiche Schreibweise der Namen!

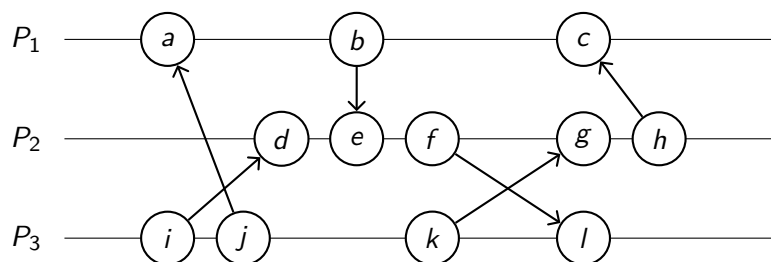
<https://svs.informatik.uni-hamburg.de/submission/for/vis17-6>

Die Übungen zu diesem Blatt finden vom 11.12.2017 bis 14.12.2017 statt.

Aufgabe 1: Logische Uhren

(12 Punkte)

- a) (Pflichtaufgabe) In einem verteilten System treten die Ereignisse a bis l auf. Die kausalen Abhängigkeiten sind durch das nachfolgende *Happened-Before*-Diagramm gegeben. Geben Sie für alle im Happend-Before-Diagramm vorhandenen Ereignisse jeweils die Lamportzeit und die Vektorzeit an. (9 Punkte)



- b) (Pflichtaufgabe) Geben Sie an, welche der folgenden Behauptungen für die Lamportzeit zutrifft und welche der Behauptungen für die Vektorzeit zutrifft. Erläutern Sie weiterhin ihre Wahl. (3 Punkte)
- I) Wenn die Zeit von Ereignis A vor der Zeit von Ereignis B liegt, dann muss eine kausale Abhängigkeit von Ereignis A zu Ereignis B vorliegen.
 - II) Wenn ein Ereignis A kausal vor Ereignis B stattfindet, dann muss auch die Zeit von A vor B liegen.
 - III) Nebenläufige Ereignisse können durch den Vergleich ihrer Zeiten erkannt werden.



Aufgabe 2: Anpassung von Zeit

(Präsenzaufgabe) Um 10:59:20 stellen Sie fest, dass die Uhr um 4 Sekunden vor geht. In einem Computersystem ist es nicht sinnvoll die Uhr direkt um 4 Sekunden zurückzustellen. Erläutern Sie diesen Sachverhalt anhand eines Beispiels. Zeigen Sie dafür, warum ein Zurückstellen der Uhr in Ihrem Beispiel Probleme bereitet. Welche andere Möglichkeit haben Sie, die Uhr bis spätestens 11 Uhr wieder auf die richtige Zeit einzustellen?

Aufgabe 3: Zeitsynchronisation

(3 Punkte)

Stellen Sie sich folgendes Szenario einer Zeitsynchronisation vor: ein Client, welcher seine Uhr synchronisieren möchte, sendet zum Zeitpunkt 10:14:00 seiner lokalen Zeit eine Anfrage an einen Zeit-Server. Um 10:14:06 seiner lokalen Zeit bekommt er die Rückantwort des Zeit-Servers. Diese Antwort beinhaltet den Zeitstempel 10:14:05.

- a) (Pflichtaufgabe) Berechnen Sie nach der Cristians-Methode eine geeignete Zeit T_{sync} , die der Client fortan als lokale Zeit verwenden sollte. Geben Sie die Zeit T_{sync} an und zeigen Sie Ihren Rechenweg. (2 Punkte)
- b) (Pflichtaufgabe) Die berechnete Zeit steht immer in Abhängigkeit zu den Laufzeiten der Nachrichten im verteilten System. Erläutern Sie kurz, auf welcher Annahme bezüglich der zwei Nachrichtenlaufzeiten die Berechnung nach Christian basiert? (1 Punkt)
- c) (Präsenzaufgabe) Diese Annahme über die Nachrichtenlaufzeiten ist allerdings nicht immer gegeben. Überlegen Sie nun, was bezüglich der Annahme im schlechtesten Fall passieren könnte. Geben Sie weiterhin an, zu welchem maximalen Fehler es bezüglich der Abweichung zwischen der berechneten synchronisierten Zeit T_{sync} und der wirklichen Zeit des Zeit-Servers in diesem schlechtesten Fall kommen kann.

Aufgabe 4: Bully-Algorithmus

(5 Punkte)

- a) (Pflichtaufgabe) Erläutern Sie die Ziele des Bully-Algorithmus und erklären Sie kurz, wie der Algorithmus bei der Erfüllung des Ziels vorgeht. Beschreiben Sie dann den Ablauf für den folgenden Fall: Das Netzwerk besitzt vier aktive Knoten mit den IDs (3, 5, 9, 11). Der Knoten mit der ID 5 startet den Bully-Algorithmus. (5 Punkte)
- b) (Präsenzaufgabe) Erläutern Sie, wie viele Nachrichten im schlechtesten und im besten Fall durch den Algorithmus verschickt werden und wann diese Fälle eintreten würden.
- c) (Präsenzaufgabe) Mit welcher Funktionalität müssten Sie den Bully-Algorithmus erweitern, damit der Algorithmus effizienter mit temporären Netzwerkpartitionen umgehen kann. Überlegen Sie dafür folgende zwei Fälle: (1) Die Trennung des ursprünglichen Netzwerkes in zwei Partitionen. (2) Die Zusammenführung des Netzwerkes aus den zwei Partitionen. Überlegen Sie jeweils, ob und mit welcher Funktionalität die Fälle effizienter bezüglich des Ziels des Algorithmus gelöst werden könnten.