

Tutorübung Grundlagen: Betriebssysteme und Systemsoftware

Moritz Beckel

München, 16. Dezember 2022

Freitag 10:15-12:00 Uhr Raum ([00.11.038](#))

Zulip-Stream <https://zulip.in.tum.de/#narrow/stream/1295-GBS-Fr-1000-A>

Unterrichtsmaterialien findet ihr hier:

<https://home.in.tum.de/~beckel/gbs>

Lösungen wurden von mir selbst erstellt. Es besteht keine Garantie auf Korrektheit.

Interprozesskommunikation

Bandbreite

= Menge der übertragbaren Informationen

1. **Schmalbandig** (bspw. Signale)
2. **Breitbandig** (bspw. Shared Memory, Sockets)

Explizit vs. Implizit

1. **Implizit** (= Kommunikation über gemeinsames Betriebsmittel, bspw. Shared Memory, keine direkte Unterstützung durch Betriebssystem)
2. **Explizit** (= Versenden und Empfangen von Nachrichten mit dedizierten send, receive Aufrufen)

Interprozesskommunikation

Kopplungsgrad

= Wie stark sind Kommunikationspartner voneinander abhängig

1. **Synchron** (Blockierend)
2. **Asynchron** (Nicht blockierend)

Muster

= Ablauf der Übertragung

1. **Meldung** (erwartet keine Rückmeldung)
2. **Auftrag** (erwartet Rückmeldung)
3. Kommunikationsrichtung (**unidirektional** vs. **bidirektional**)

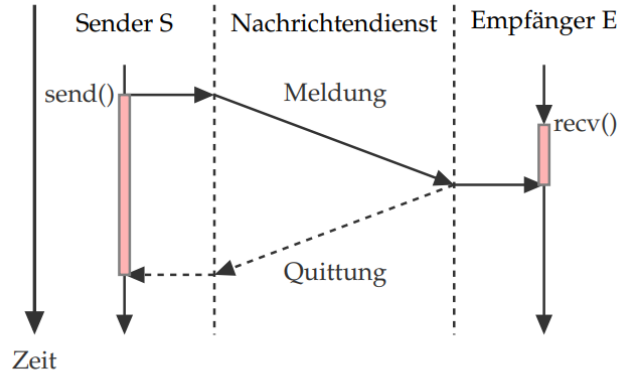


Abbildung 5.2: Synchrone Meldung

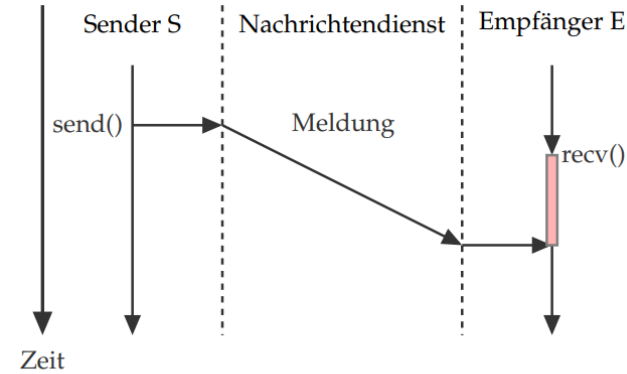


Abbildung 5.1: Asynchrone Meldung

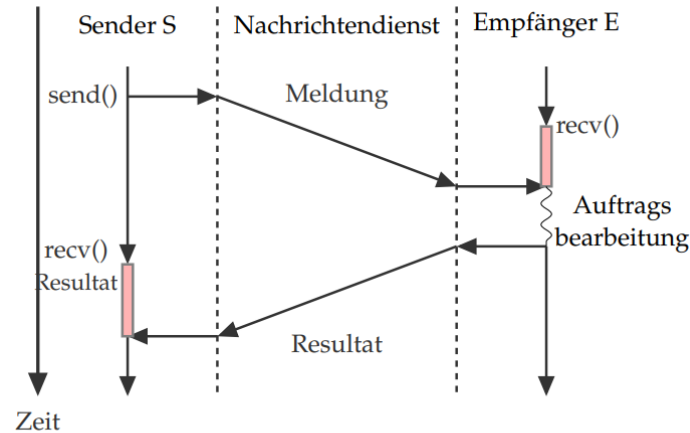


Abbildung 5.3: Asynchroner Auftrag

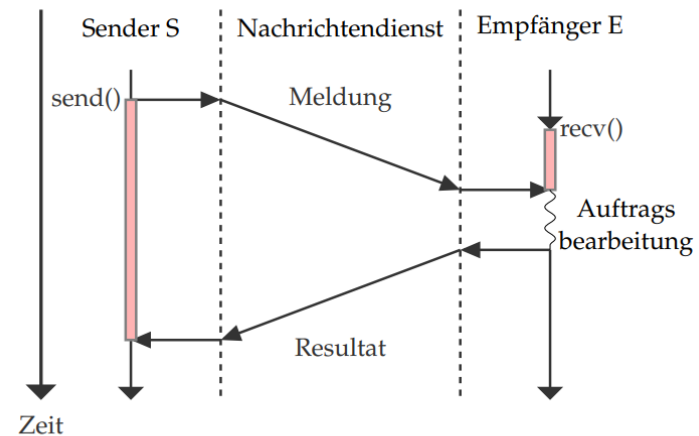


Abbildung 5.4: Synchroner Auftrag

Linux-Sockets

Adressierung

1. Quell-IP-Adresse
2. Ziel-IP-Adresse
3. Quell-Port
4. Ziel-Port
5. Protokoll (TCP, UDP)

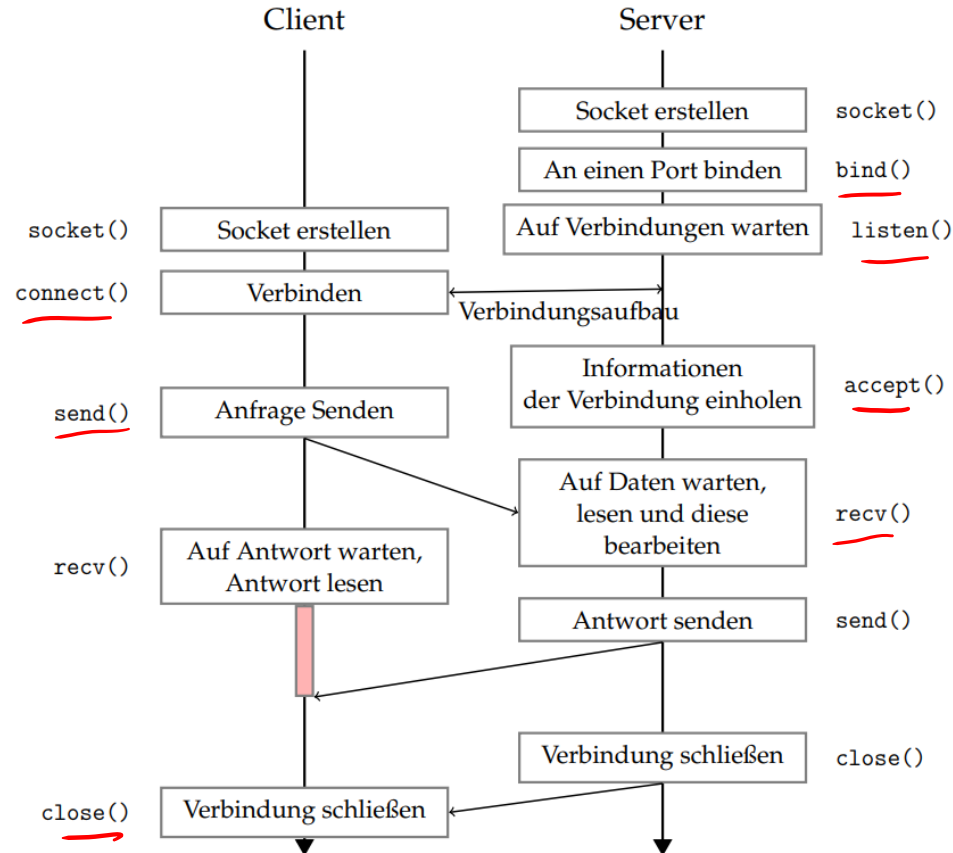
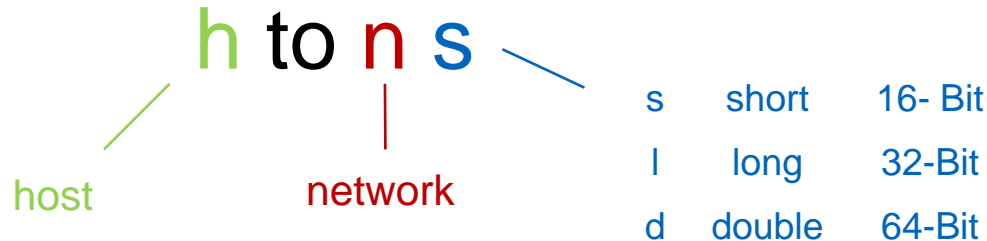



Abbildung 5.9: Sequenzdiagramm Client-Server-Kommunikation.

Interprozesskommunikation

Byte-Order

- Netzwerke verwenden immer Big-Endian
- Die meisten Heimrechner verwenden Little-Endian
- Folgende Funktionen werden zur Umwandlung verwendet (htons, ntohs)

Adresse	Big Endian	Mixed Endian	Little Endian
10000	01	02	04
10001	02	01	03
10002	03	04	02
10003	04	03	01

Aufgabe 2

$A = (0, 4, 2)$ $S = ((P1, 0, P2), (P1, 2, P3), (P2, 5, P1))$ $R = (10, -, 4)$

- P1, gestartet zum Zeitpunkt $t = 0$
- P2, gestartet zum Zeitpunkt $t = 4$
- P3, gestartet zum Zeitpunkt $t = 2$

Die Prozesse tauschen Nachrichten untereinander aus:

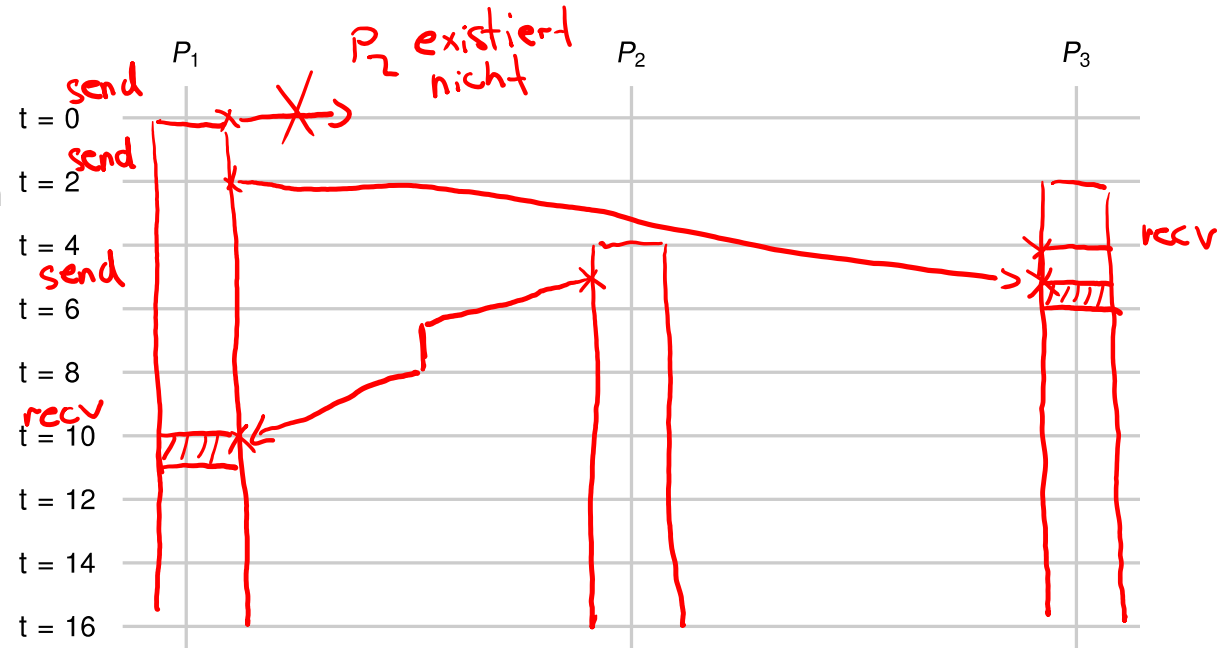
- P1 sendet Nachricht C zum Zeitpunkt $t = 0$ an P2
- P1 sendet Nachricht A zum Zeitpunkt $t = 2$ an P3
- P2 sendet Nachricht B zum Zeitpunkt $t = 5$ an P1

Die **Laufzeit** einer Nachricht beträgt 3 Zeiteinheiten, die **Verarbeitungszeit** genau 1 Zeiteinheit. Eine **receive** Operation zum Empfangen von Nachrichten **wartet maximal 4** Zeiteinheiten. Andere Operationen geschehen overheadfrei. Jeder Prozess kann die Abarbeitung von Nachrichten parallel vornehmen. Zum Empfangen von Nachrichten werden zu folgenden Zeiten **receive**-Operationen gestartet:

- P3 zum Zeitpunkt $t = 4$
- P1 zum Zeitpunkt $t = 10$

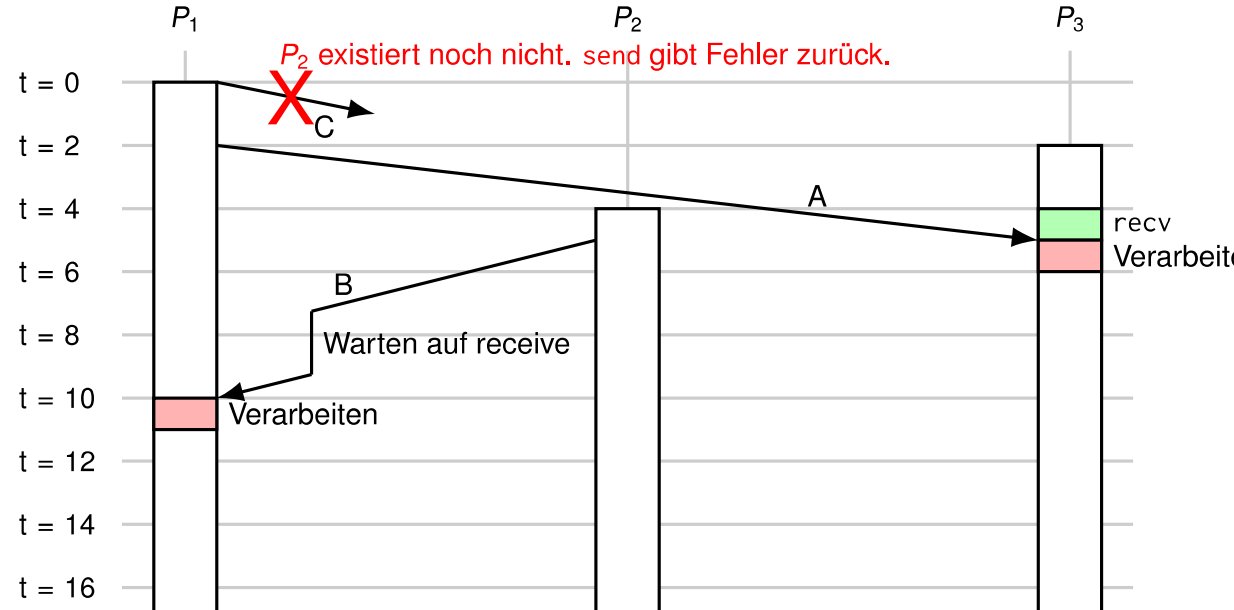
Aufgabe 2 a) $A = (0,4,2)$ $S = ((P_1,0,P_2), (P_1,2, \underline{P_3}), (\underline{P_2},5,\underline{P_1}))$ $R = (10, -, 4)$

Nehmen Sie an, dass die Kommunikation zwischen den Prozessen **asynchron** ist. Modellieren Sie die Kommunikationsvorgänge in einem **Sequenzdiagramm**.



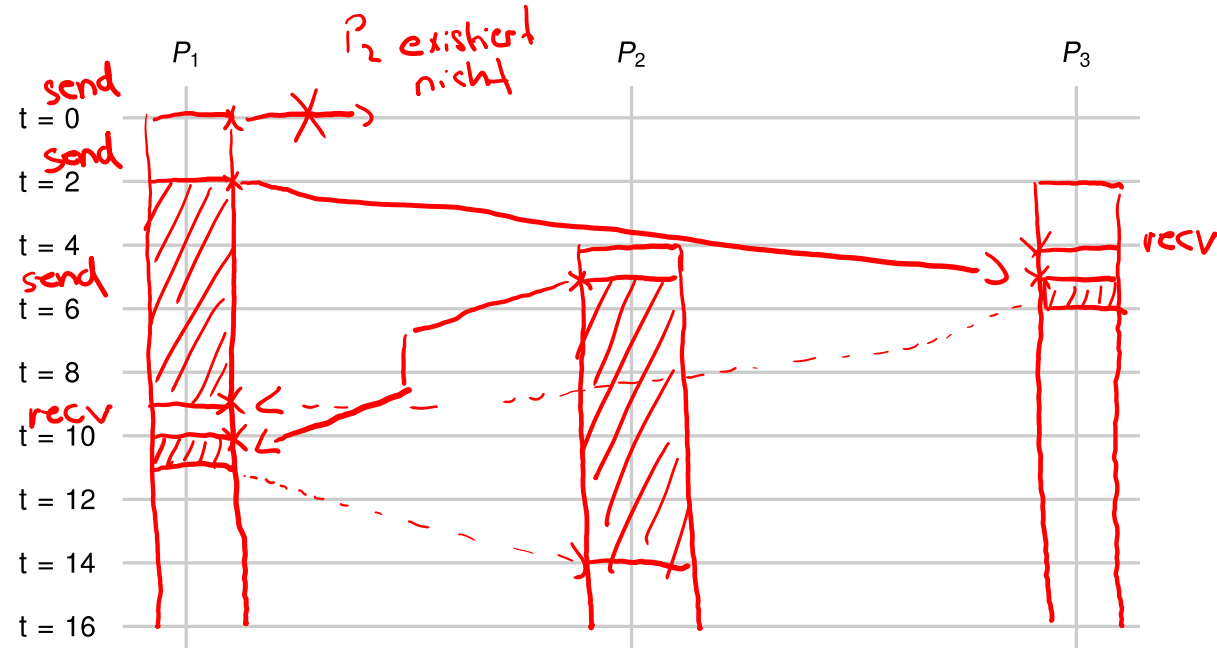
Aufgabe 2 a) $A = (0,4,2)$ $S = ((P_1,0,P_2), (P_1,2, P_3),(P_2,5,P_1))$ $R = (10, -, 4)$

Nehmen Sie an, dass die Kommunikation zwischen den Prozessen **asynchron** ist. Modellieren Sie die Kommunikationsvorgänge in einem **Sequenzdiagramm**.



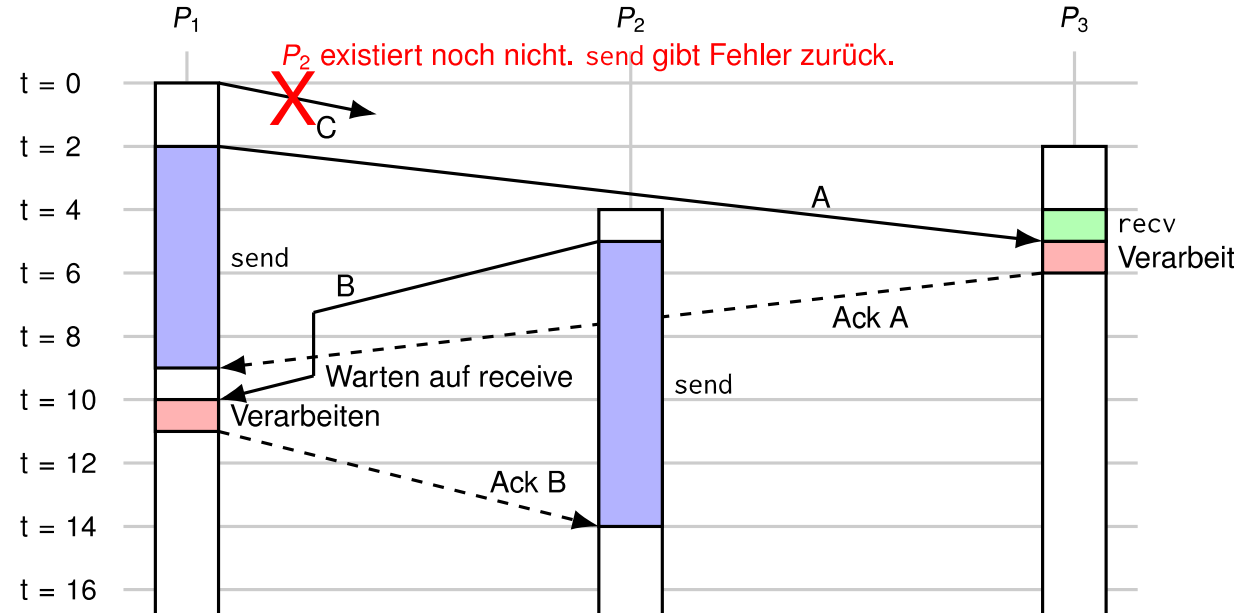
Aufgabe 2 b) $A = (0,4,2)$ $S = ((P_1,0,P_2), (P_1,2, P_3),(P_2,5,P_1))$ $R = (10, -, ,4)$

Wiederholen Sie Aufgabe a) für den Fall von **synchroner** Kommunikation zwischen den Prozessen. Gehen Sie von einer **Bestätigung einer Nachricht** nach deren Verarbeitung aus. Beachten Sie, dass eine Nachricht eventuell wegen eines blockierenden Empfängers erst zum nächstmöglichen Zeitpunkt ankommt. Bestätigungen haben die gleiche Laufzeit wie Nachrichten.



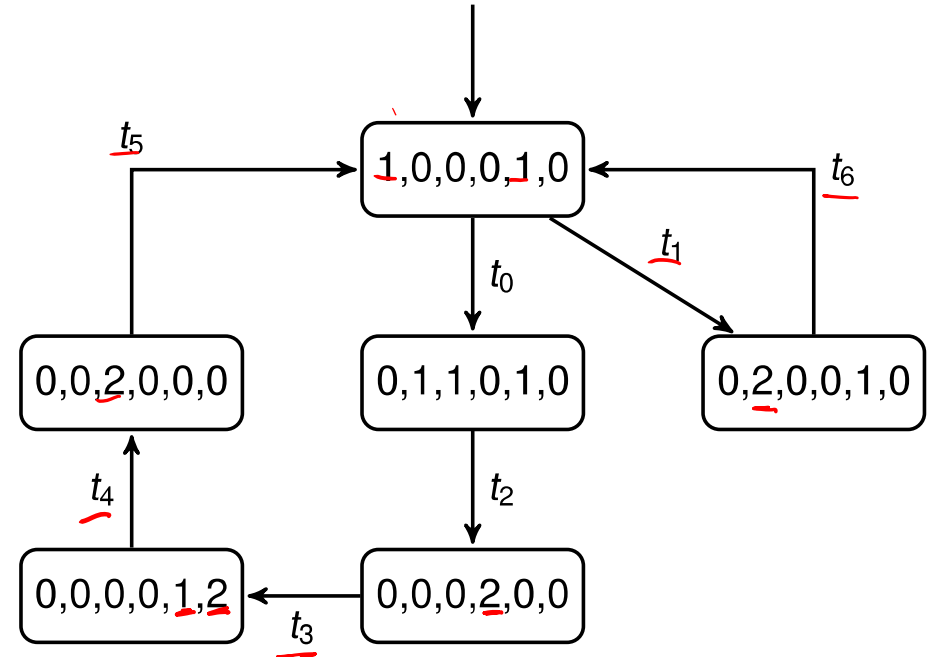
Aufgabe 2 b) $A = (0,4,2)$ $S = ((P_1,0,P_2), (P_1,2, P_3),(P_2,5,P_1))$ $R = (10, - ,4)$

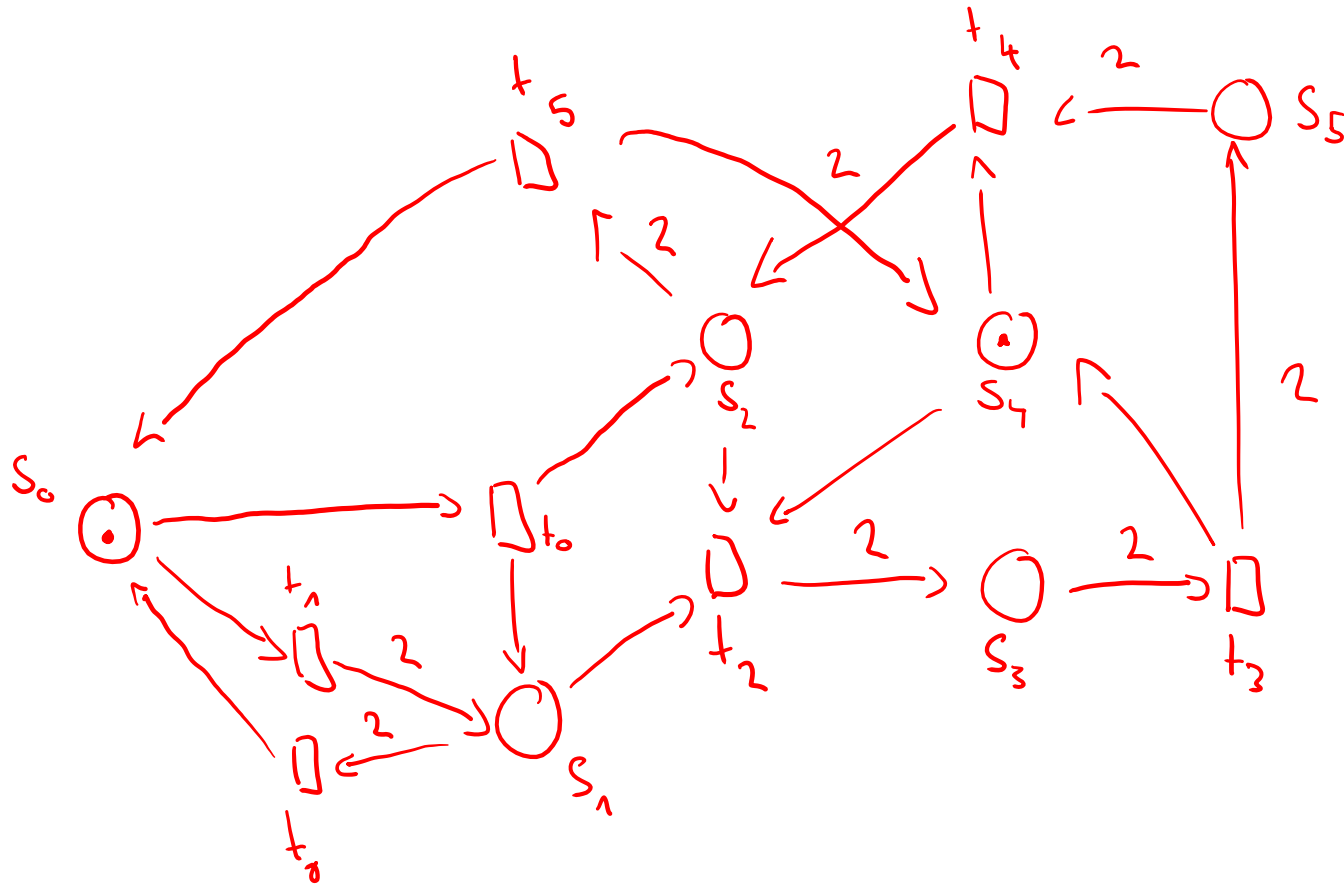
Wiederholen Sie Aufgabe a) für den Fall von **synchroner** Kommunikation zwischen den Prozessen. Gehen Sie von einer **Bestätigung einer Nachricht** nach deren Verarbeitung aus. Beachten Sie, dass eine Nachricht eventuell wegen eines blockierenden Empfängers erst zum nächstmöglichen Zeitpunkt ankommt. Bestätigungen haben die gleiche Laufzeit wie Nachrichten.

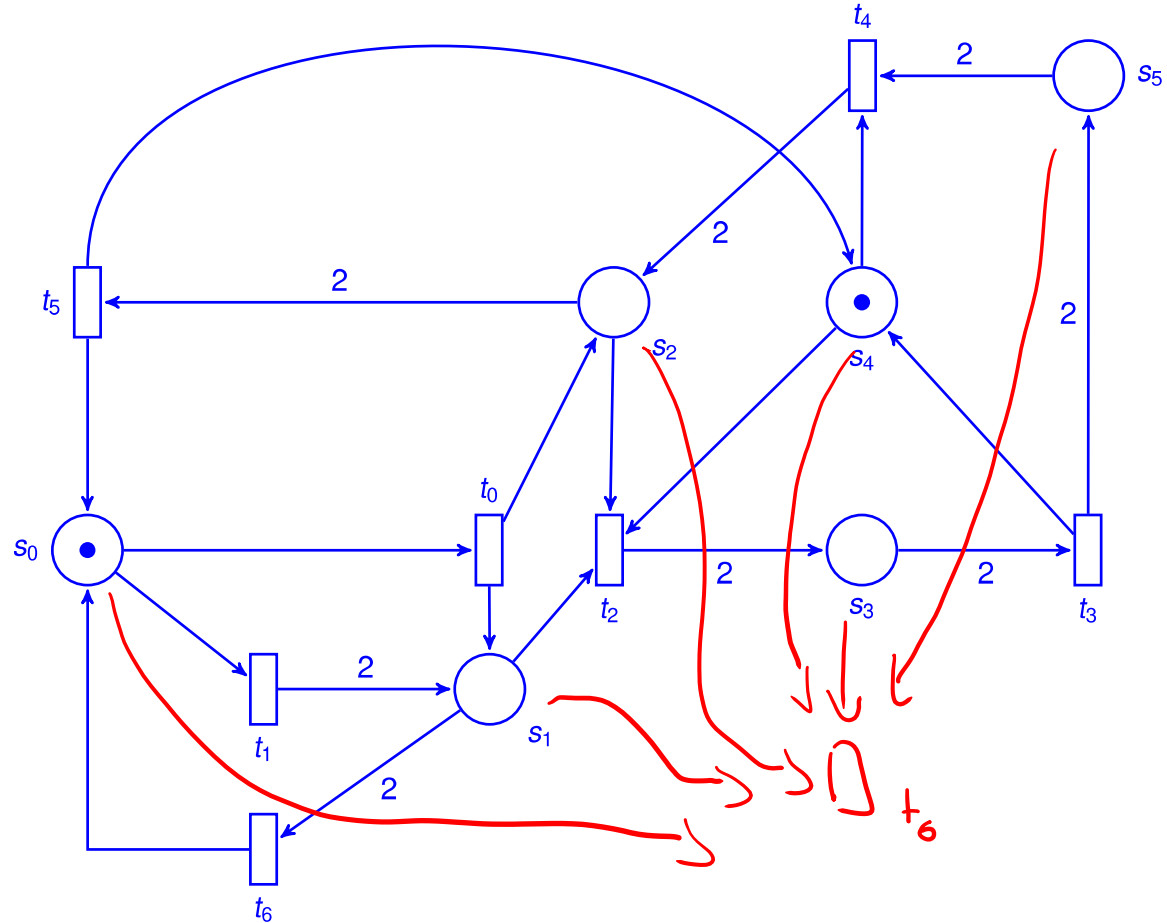


Aufgabe 3

- a) Modellieren Sie das zu dem Erreichbarkeitsgraphen **zugehörige Petrinetz** mit der Startmarkierung $(1, 0, 0, 0, 0, 1, 0)$.







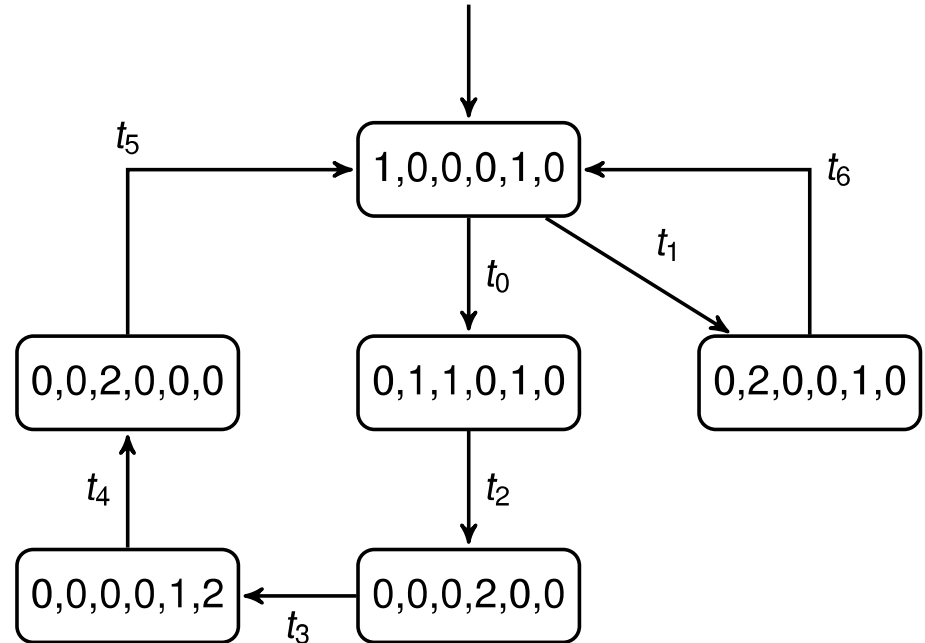
Aufgabe 3

- b) Argumentieren Sie anhand des Erreichbarkeitsgraphen, welche der Ihnen bekannten **Eigenschaften** das dazugehörige Petrinetz besitzt.

Verklemmungsfrei?

Lebendig?

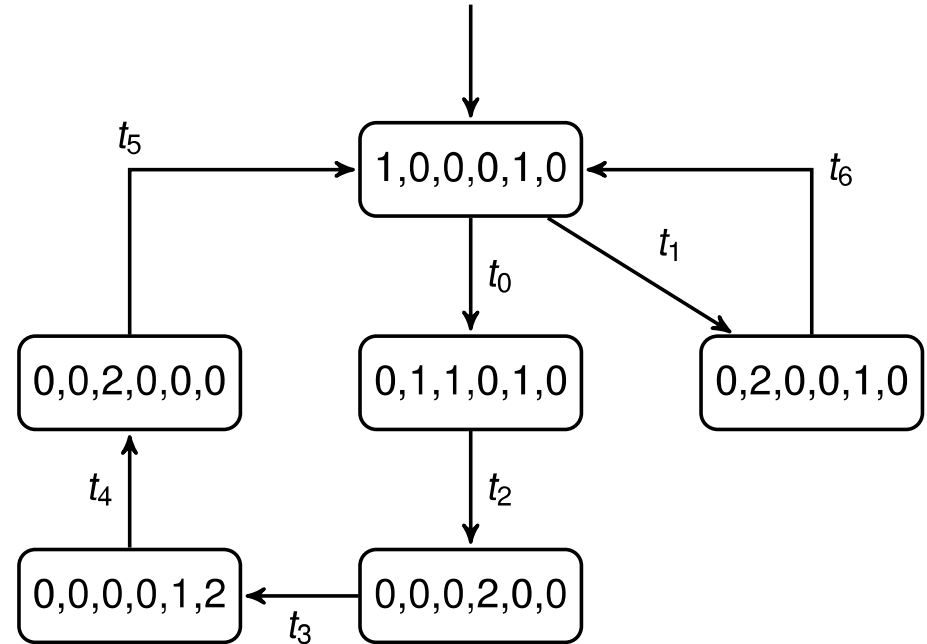
Verhungern?



Aufgabe 3

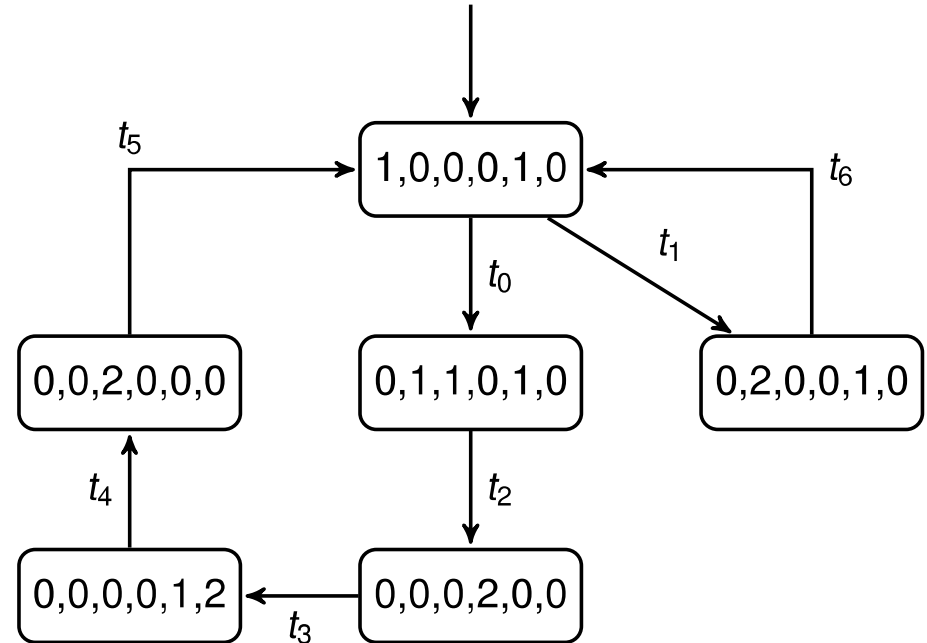
- b) Argumentieren Sie anhand des Erreichbarkeitsgraphen, welche der Ihnen bekannten **Eigenschaften** das dazugehörige Petrinetz besitzt.

Verklemmungsfrei?	Ja
Lebendig?	Ja
Verhungern?	Ja



Aufgabe 3

- c) Ist das zu einem Erreichbarkeitsgraphen zugehörige Petrinetz **eindeutig**?



Aufgabe 3

- c) Ist das zu einem Erreichbarkeitsgraphen zugehörige Petrinetz **eindeutig**?

Nein, es können Transitionen hinzugefügt werden, die niemals schaltbereit sind.
(oder Kapazitätsbegrenzungen)

