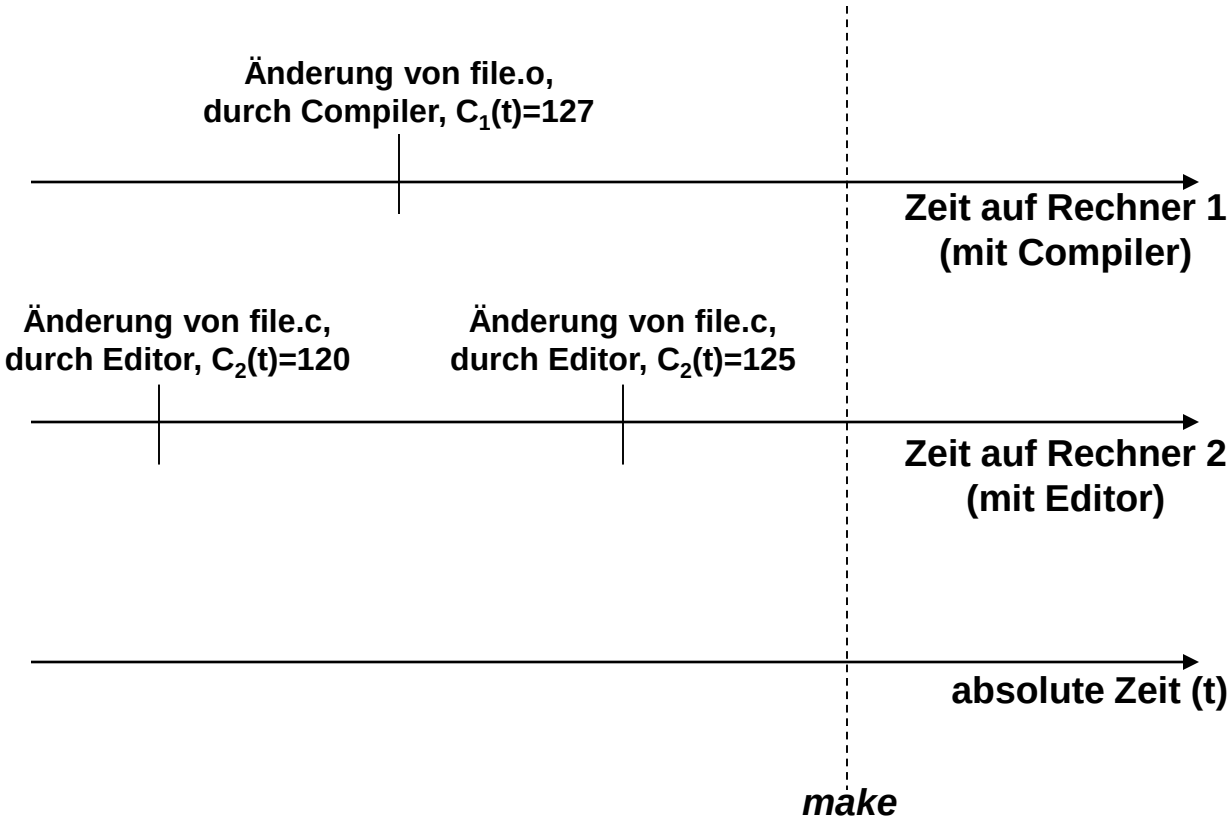
Recovery	**Bei einem Fehlerfall in einem verteilten System kann ein Rücksetzen auf Basis von System-Logs notwendig werden. Ereignisse werden unter Berücksichtigung ihrer Zeitstempel rückgängig gemacht.**
Roboter	**Prozessoren und Sensoren in verschiedenen Roboterkomponenten müssen synchronisiert werden. Beispiel: Fußballroboter**
Sensornetzwerk	**Die Zeitpunkte von gemessenen Ereignissen in Produktionsabläufen in z.B. Industriebetrieben müssen zeitlich korreliert werden können.**
Echtzeit-anwendungen	**Die Zeitpunkte von Ereignissen müssen zeitlich genau festgelegt werden können.**
verteilte Online-Auktion	**Die zeitlichen Reihenfolgen der Angebote müssen nachvollziehbar festgelegt werden.**
make	**Bei Ablauf eines makefiles müssen die richtigen Software-Komponenten aktualisiert werden.**

```
# Teil einer Makefile
...
file.o : file.c
        cc -c file.c
...
```

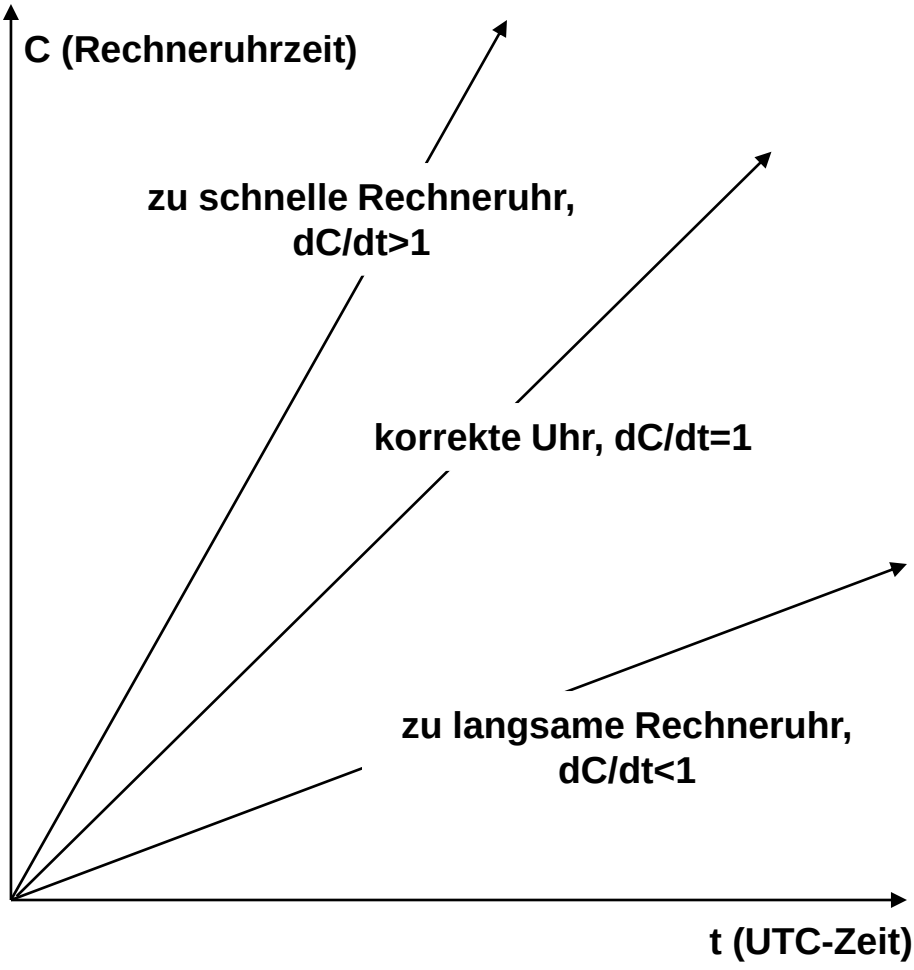


$C_i(t)$: Zeitwert der Systemuhr (Clock) auf Rechner i zum absoluten Zeitpunkt t.

***make* aktualisiert file.o fälschlicherweise nicht.**

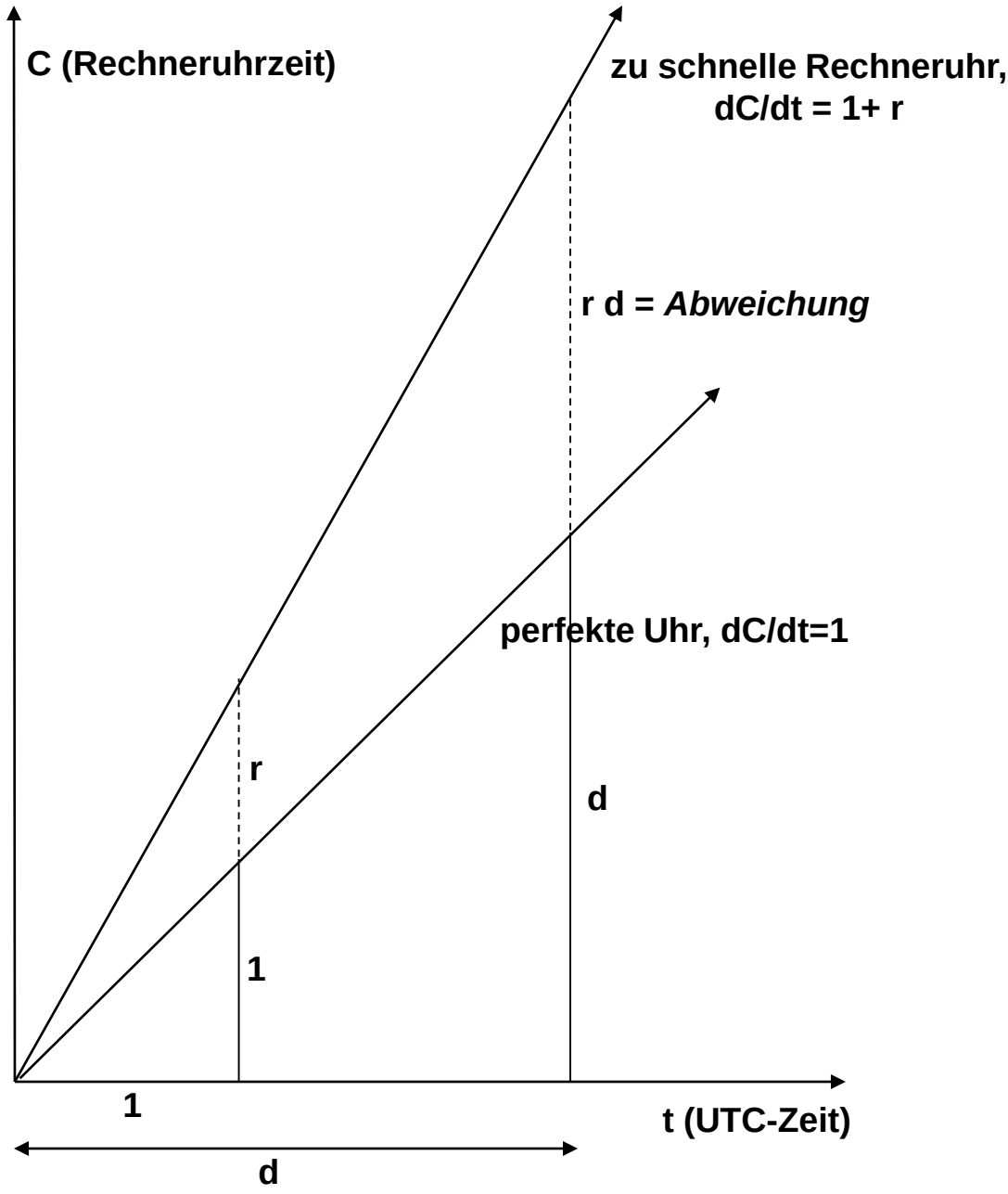
Bei Änderung von file.o durch den Compiler zum Zeitpunkt $C_1(t)=122$ wäre die Aktualisierung von file.o erfolgt.

Zeitprobleme mit *make* im verteilten System



nach [Tanenbaum]

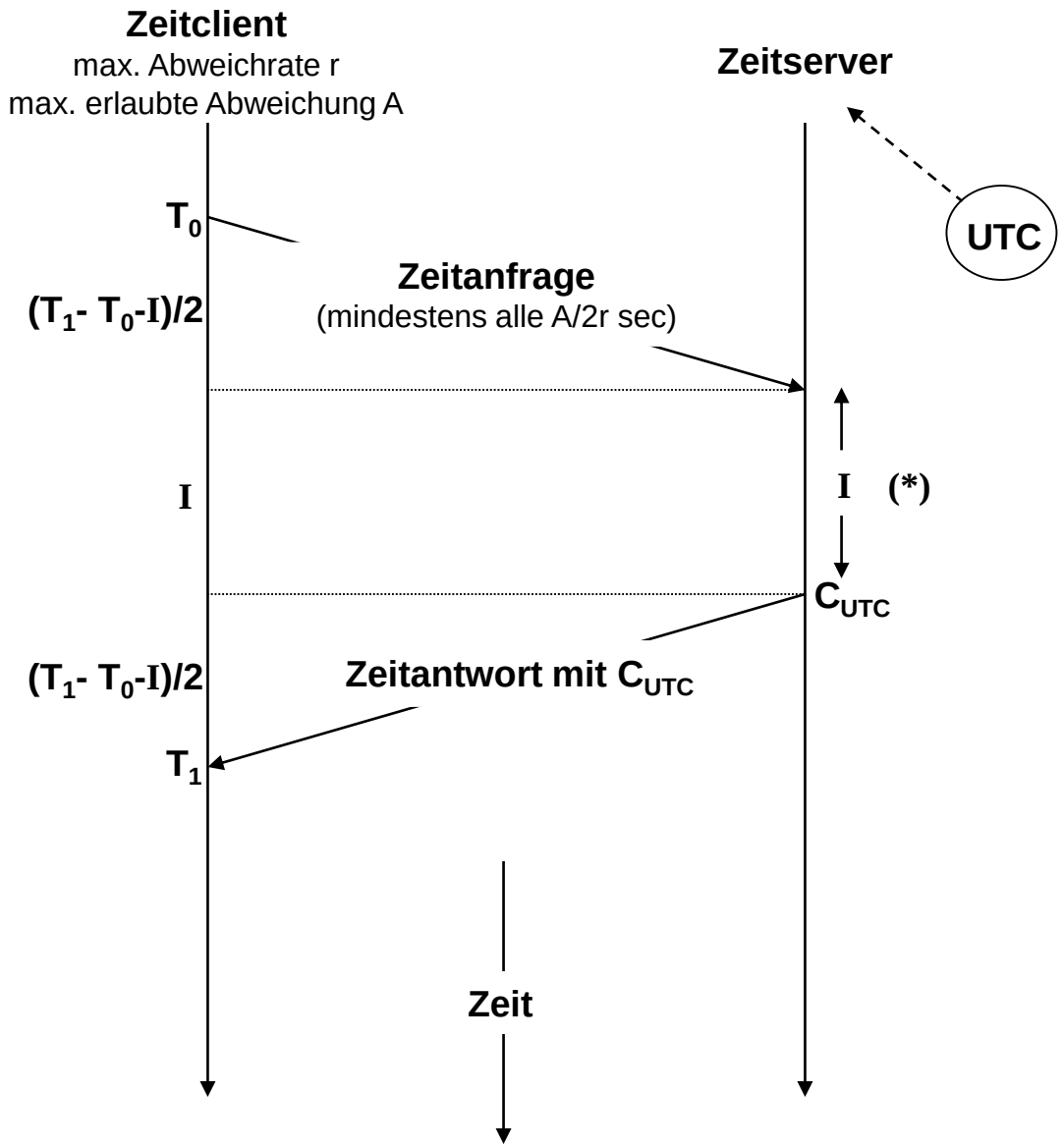
physikalische Uhren im verteilten System



⇒ **Abweichung A zwischen zwei Rechneruhren**

mit dC/dt=1+r und dC/dt=1-r nach einer Zeitspanne d: A = 2 r d

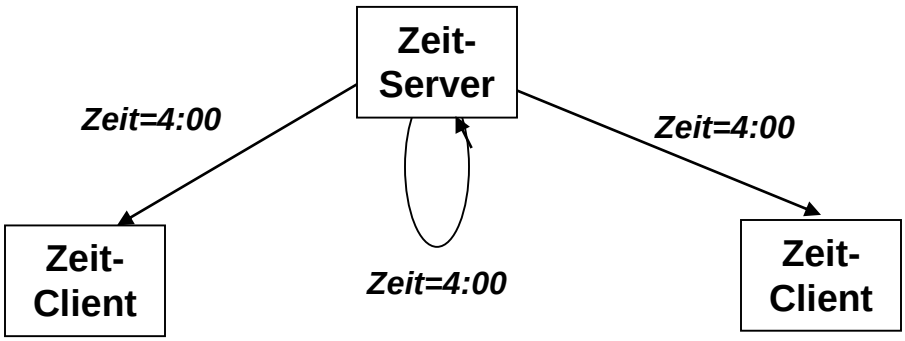
Abweichung physikalischer Uhren



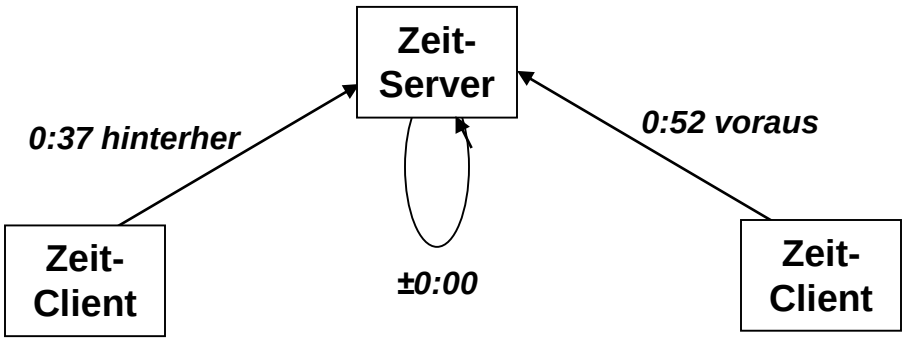
(*) I umfasst Interrupterkennung und –verarbeitung am Zeitserver

T_0: Zeit des Zeitclients beim Aussenden der Zeitanfrage
T_1: Zeit des Zeitclients beim Empfang der Zeitantwort, Zeitclient stellt seine Uhr auf $C_{UTC} + (T_1- T_0-I)/2$ ein

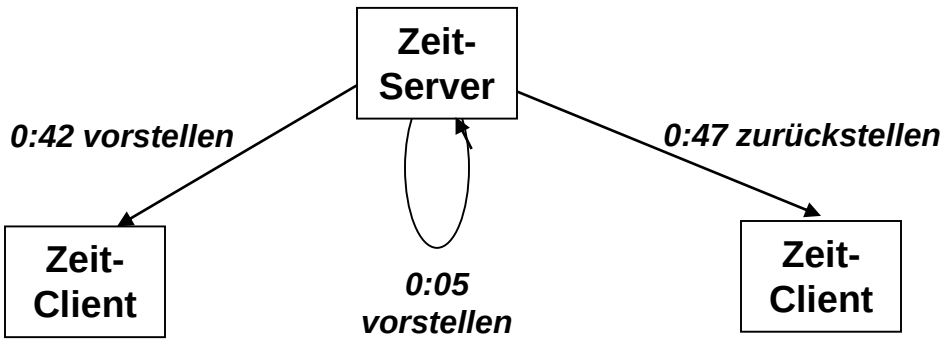
Synchronisierung mit passivem Zeitserver, zentraler Algorithmus



Zeitrequest des Zeitservers



Zeitreplies von Zeitclients und -Server
⇒ Durchschnitt = +0:05



Zeitkorrekturen für Zeitclients und -Server

Synchronisierung mit aktivem Zeitserver, zentraler Algorithmus

Anforderungen:

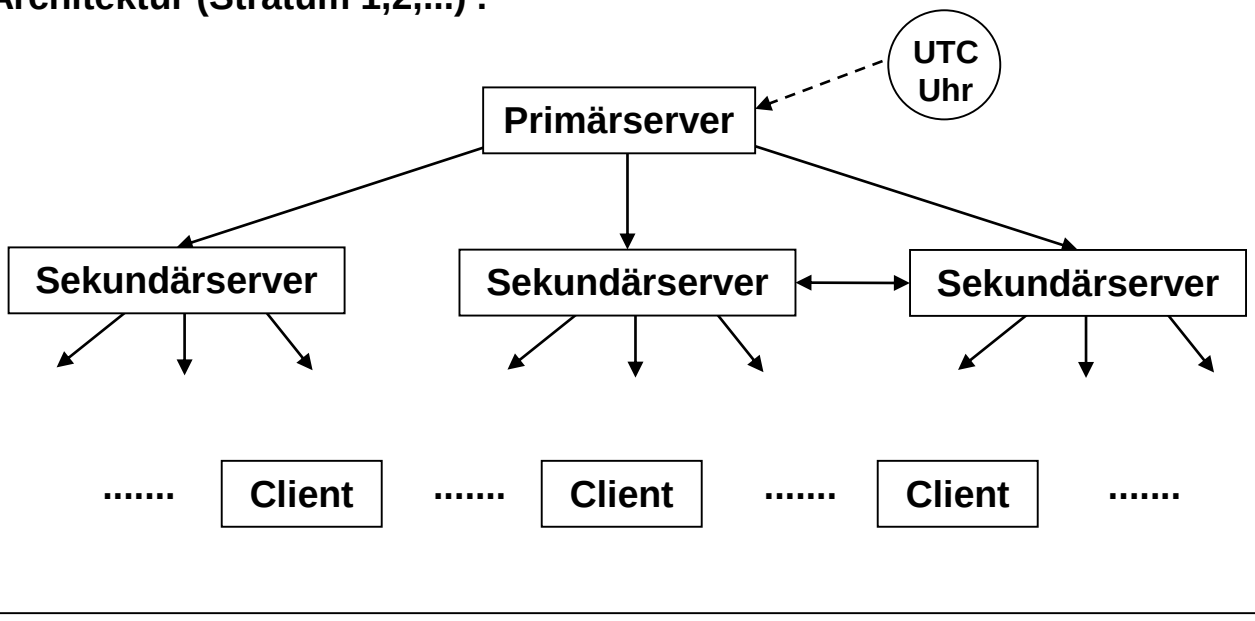
Die Uhrensynchronisierung auf UTC soll über das Internet möglich sein.

Die Synchronisierung soll auch bei längeren Verbindungs-unterbrechungen zuverlässig möglich sein.

Es soll die Möglichkeit häufiger Synchronisierung durch Clients auch bei großen Anzahlen von Servern und Clients möglich sein.

Sicherheit: Clients müssen von Servern Authentisierung anfordern können.

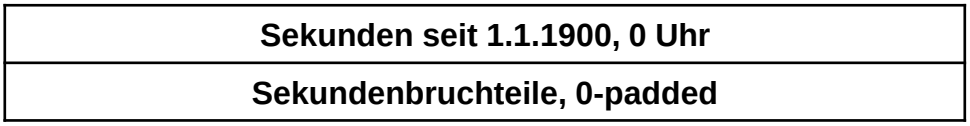
Architektur (Stratum 1,2,...) :



Synchronisierungsrichtung

Network Time Protocol (NTP): Anforderungen und Architektur

Zeitstempel:



Operationsmodi:

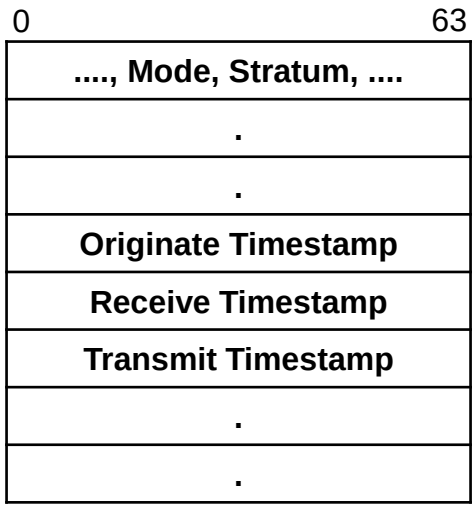
Symmetrischer Modus:
Austausch sehr genauer Zeitinformationen zwischen einem Paar von Zeitservern.

Client/Server-Modus:
Ähnlich zum zentralen passiven Algorithmus. Ein Zeitserver nimmt Anforderungen von Zeitclients oder untergeordneten Zeitservern entgegen, die er mit einem Zeitstempel beantwortet.

Broadcast-Modus:
Ein Zeitserver sendet periodische Zeitnachrichten an andere Zeitserver. Diese setzen ihre Uhren unter Berücksichtigung einer kleinen Verzögerung, sie antworten nicht.

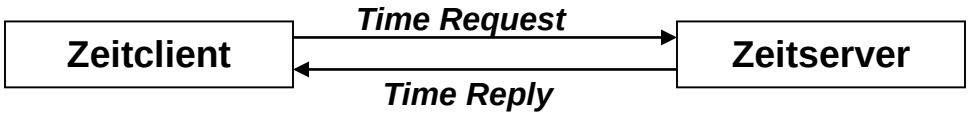
Genauigkeit (Broadcast-Modus)
< Genauigkeit (Client/Server-Modus)
< Genauigkeit (Symmetrischer Modus)

Nachrichten:

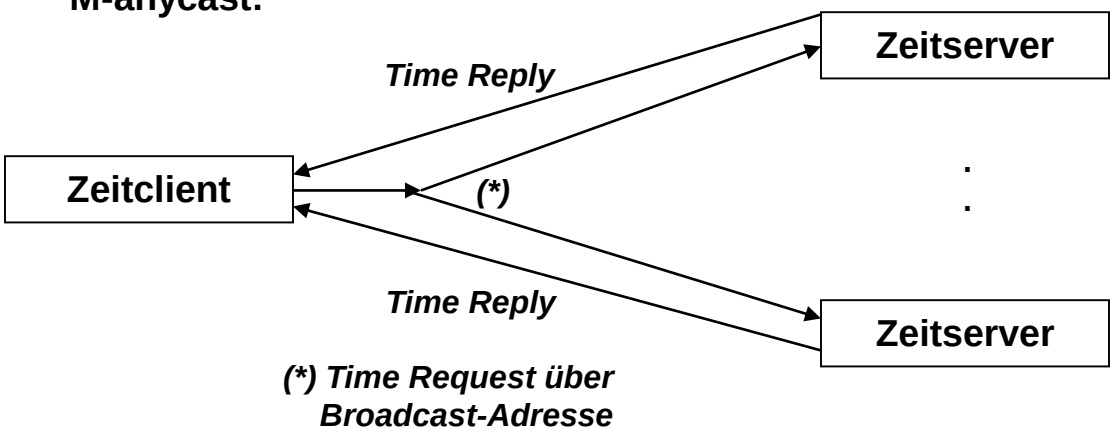


Network Time Protocol (NTP): Zeitstempel, Operationsmodi, Nachrichten

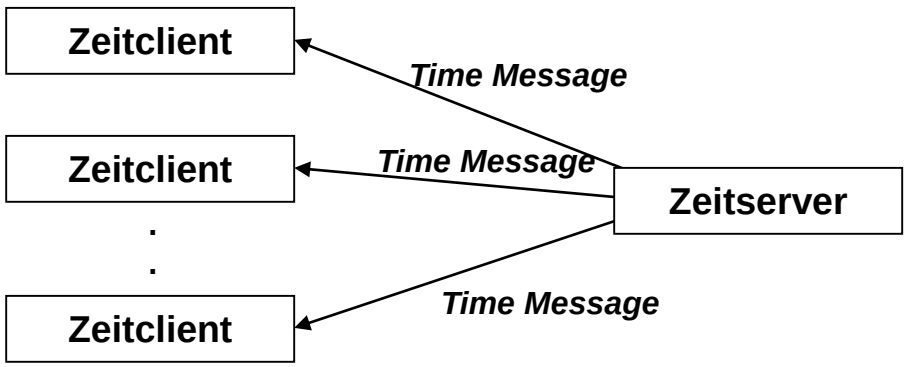
Unicast:



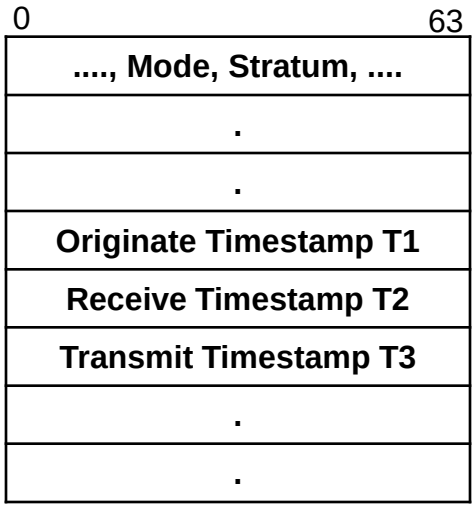
M-anycast:

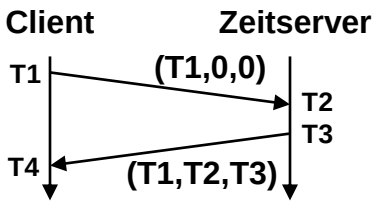


Broadcast:



Simple Network Time Protocol (SNTP): Operating Modes





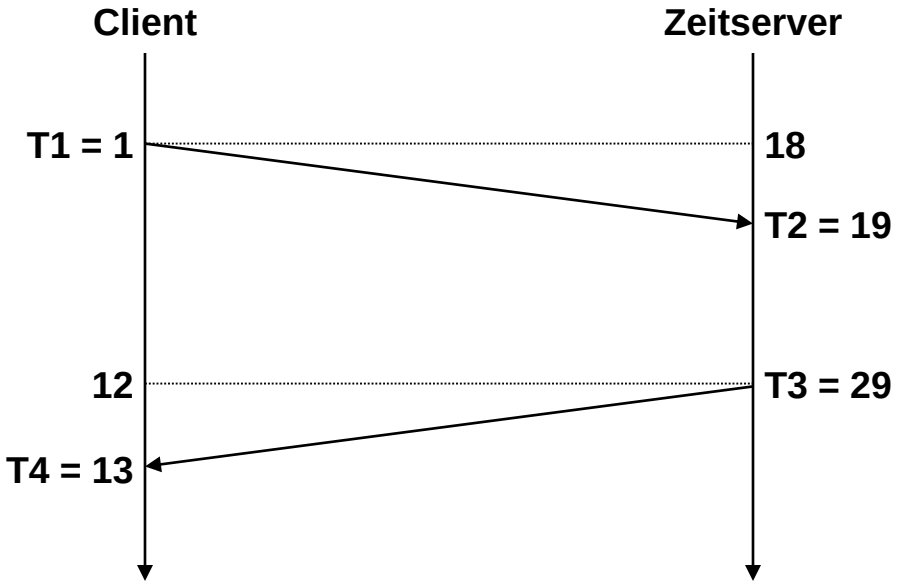
T1: Client sendet Request

T2: Server empfängt Request

T3: Server sendet Reply

T4: Client empfängt Reply
(nicht in Message enthalten)

Client-Uhr geht 17 Sekunden nach: Clock Offset = 17



Offset t = ((T2-T1) + (T3-T4)) / 2 = ((19-1) + (29-13)) / 2 = (18+16) / 2 = 17

Round Trip Delay d = (T4-T1) - (T3-T2) = (13-1) - (29-19) = 2

Clock Offset = Zeitdifferenz zwischen Uhren
Round Trip Delay = Übertragungszeit von Zeitanfrage und Zeitantwort

Simple Network Time Protocol (SNTP): Message, Ablauf und Beispiel

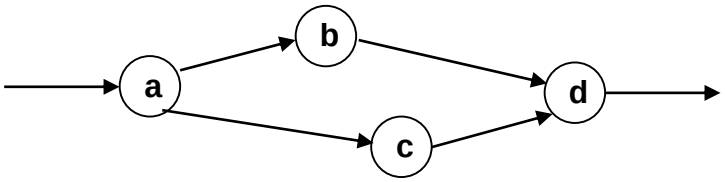
Partielle Ordnung $\leq$ für Ereignisse a, b, c, ...

$a \leq a$	Reflexivität
$a \leq b \wedge b \leq a \Rightarrow a = b$	Antisymmetrie
$a \leq b \wedge b \leq c \Rightarrow a \leq c$	Transitivität

Strikte partielle Ordnung $\leq$ für Ereignisse a, b, c, ...

Für kein a gilt: $a \leq a$	Irreflexivität
$a \leq b \wedge b \leq a \Rightarrow a = b$	Antisymmetrie
$a \leq b \wedge b \leq c \Rightarrow a \leq c$	Transitivität

Eine (strikte) partielle Ordnung lässt nebenläufige Ereignisse zu:

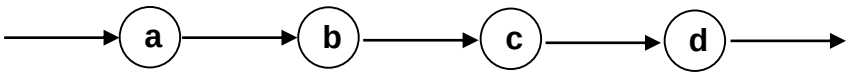


Totale Ordnung $\leq$ für Ereignisse a, b, c, ...

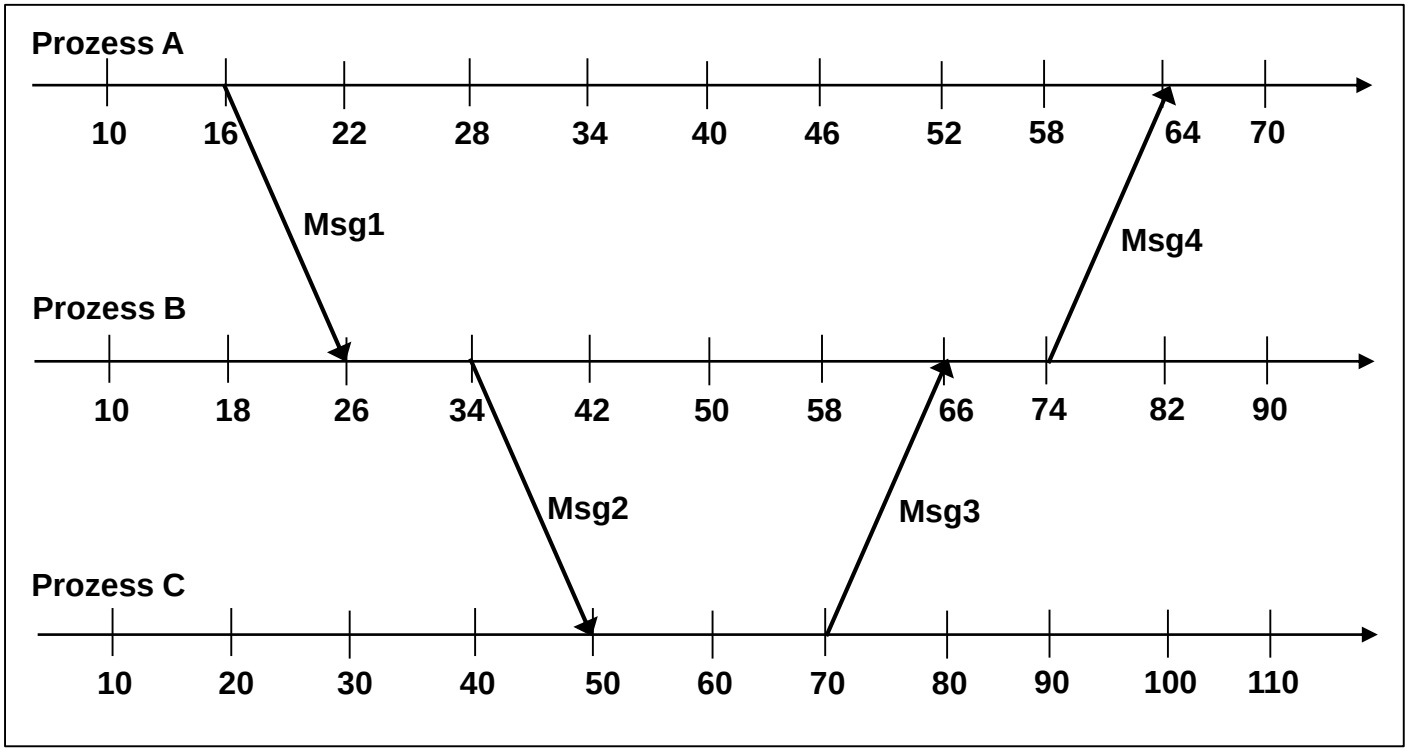
$a \leq a$	Reflexivität
$a \leq b \wedge b \leq a \Rightarrow a = b$	Antisymmetrie
$a \leq b \wedge b \leq c \Rightarrow a \leq c$	Transitivität
$a \leq b \vee b \leq a$	Totalität
Beispiel: $\leq$ auf den ganzen Zahlen	

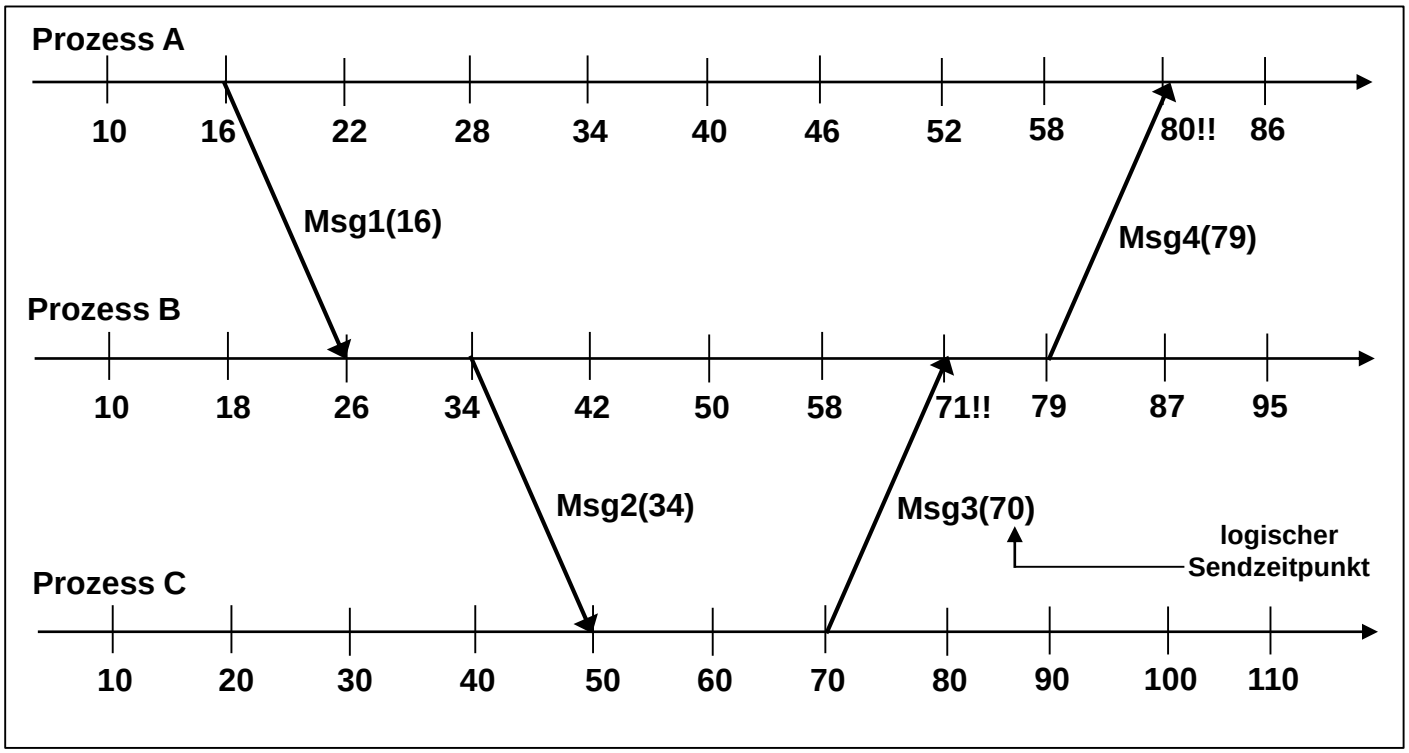
Für die Definition einer strikten totalen Ordnung wird wiederum die Reflexivität durch die Irreflexivität ersetzt.
Beispiel: < auf den ganzen Zahlen.

Eine (strikte) totale Ordnung lässt nur "Ketten" von Ereignissen zu:



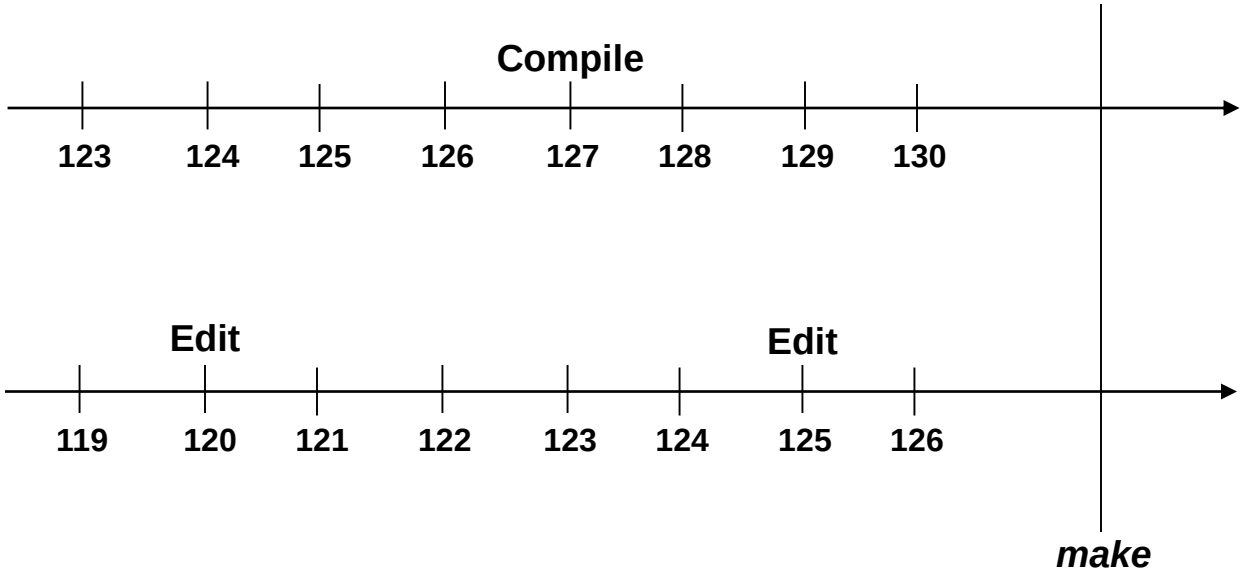
Partielle, totale Ordnungen



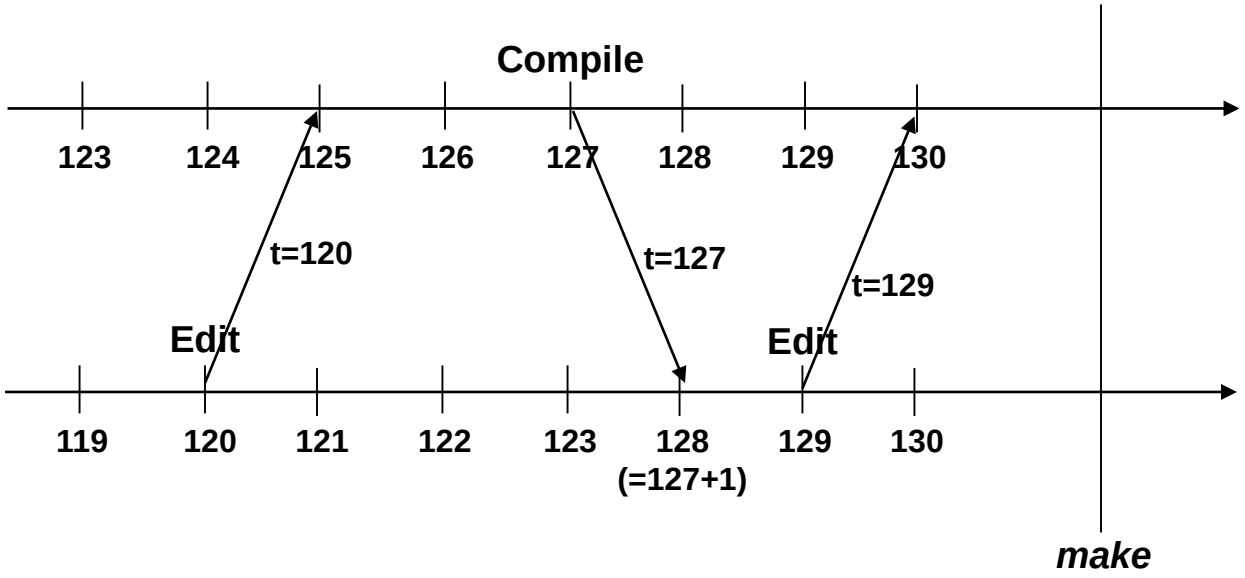


Lamport-Algorithmus für logische Zeiten VS 3.12

```
# Teil einer Makefile
file.o : file.c
        cc -c file.c
```



make **aktualisiert file.o fälschlicherweise nicht.**

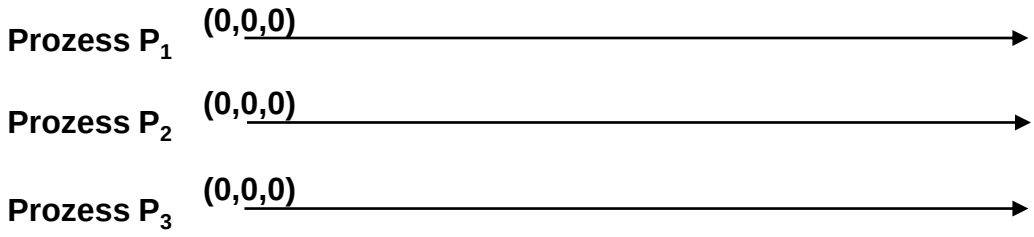


Partner-Prozess wird von Edit- bzw. Compile-Ereignissen benachrichtigt.

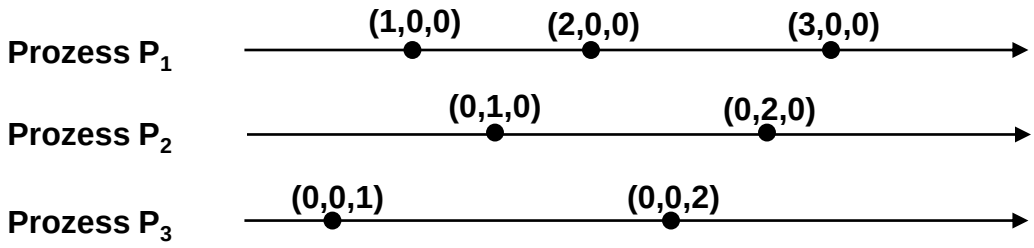
make **aktualisiert file.o (letztes Ereignis ist Edit-Ereignis).**

Lamport-Zeitstempel und make

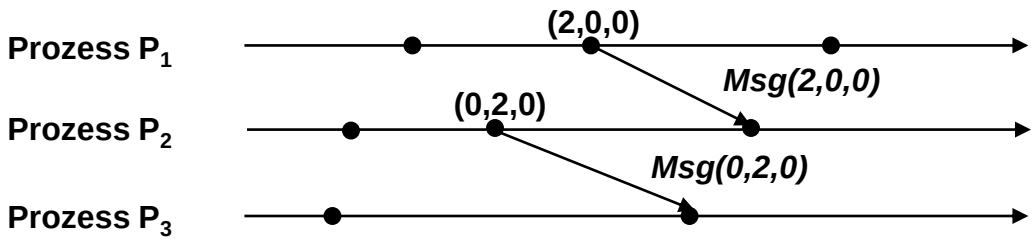
VC1: Am Anfang setzt $P_{i:}$ $V_i[j] := 0$ für $\forall j$



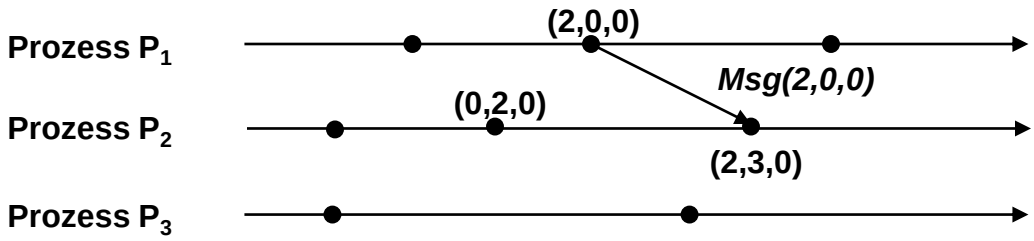
VC2: Unmittelbar vor Zuweisung eines Zeitstempels an ein eigenes Ereignis setzt P_i: $V_i[i] := V_i[i] + 1$, $V_i[j] := V_i[j]$ mit $j \neq i$ sonst



VC3: P_i gibt jeder versendeten Nachricht den momentanen Wert seiner Vektor-Uhr V_i als Zeitstempel $T_i[1:N]$ mit.



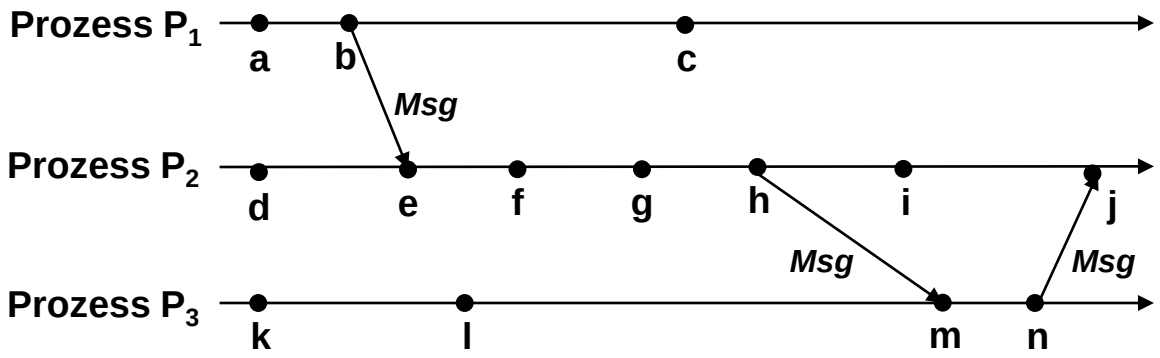
VC4: Empfängt P_i den Zeitstempel $T_k[1:N]$ in einer Nachricht von P_k, setzt er $V_i[j] := \max \{ V_i[j], T_k[j] \}$ $\forall j$ und danach $V_i[i] := V_i[i] + 1$.



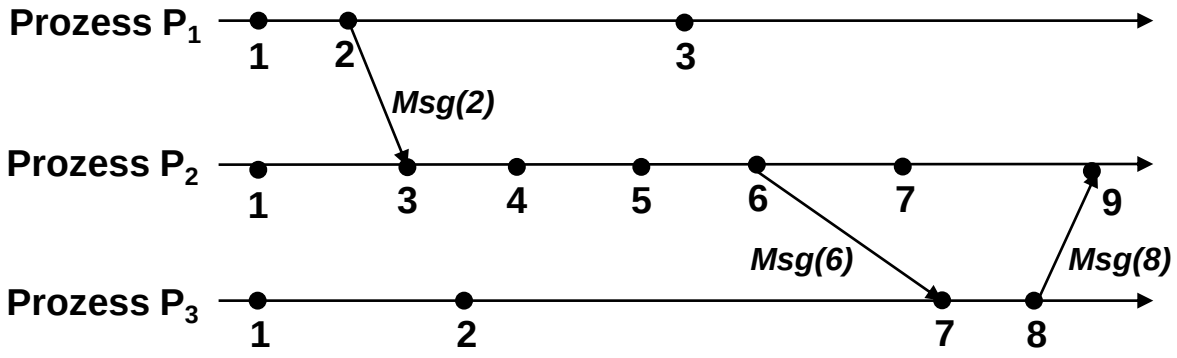
Vektor-Zeitstempel: Aktualisierungsregeln

Msg = Message *Zeit*

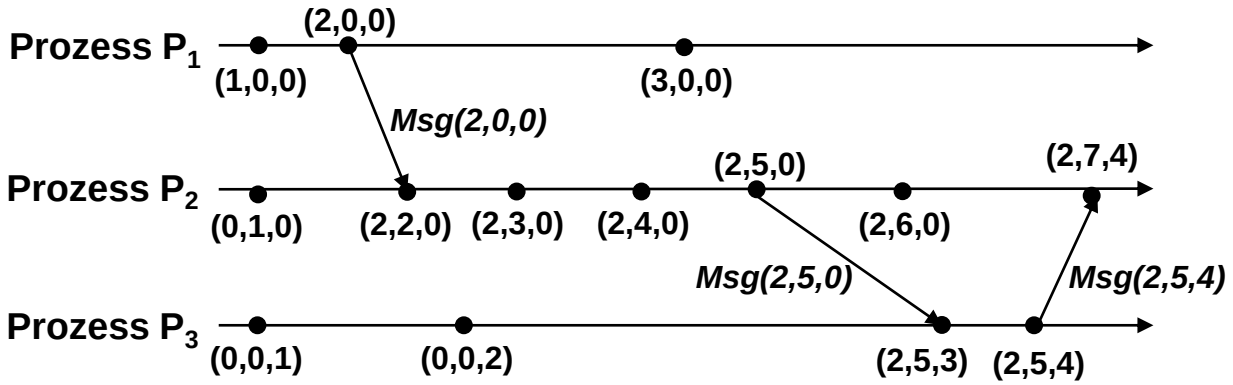
Ereignisse:



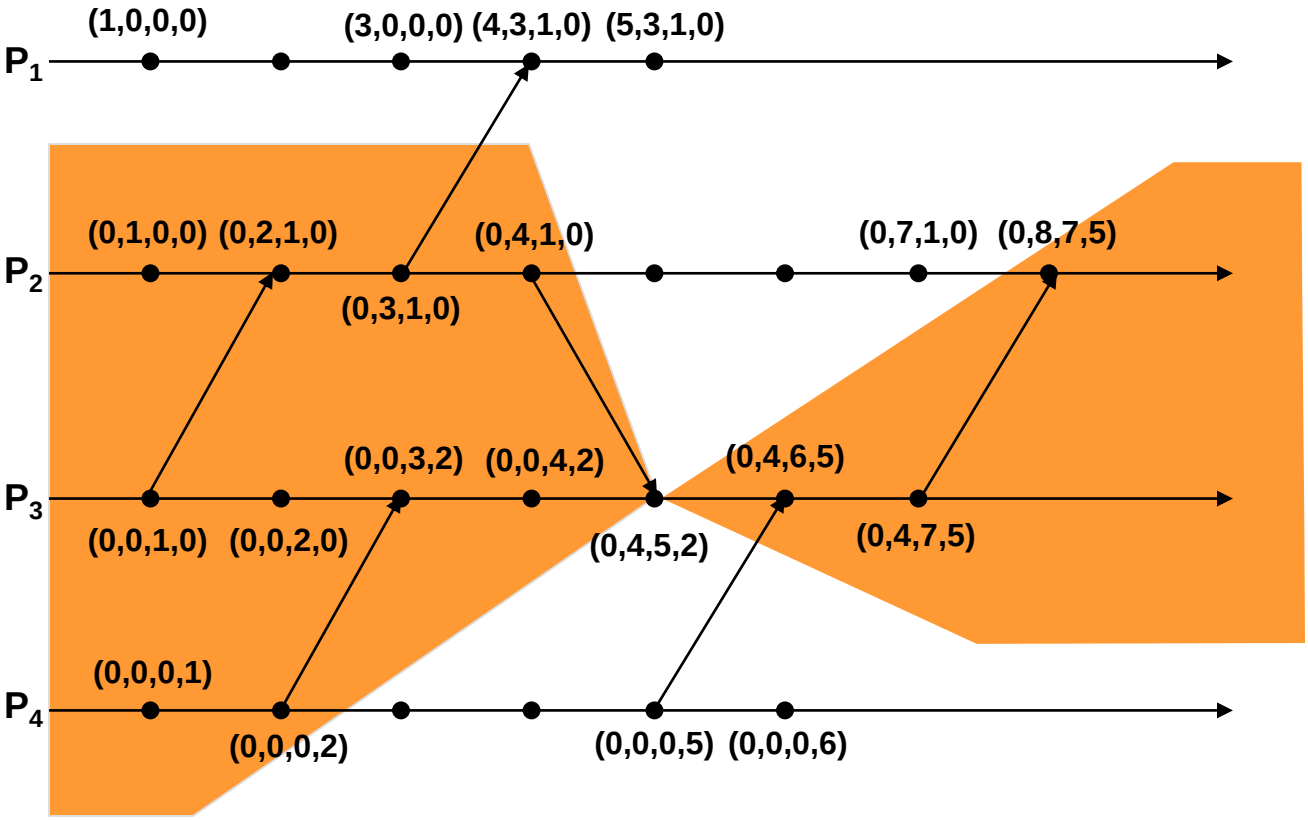
Lamport-Stempel:



Vektor-Stempel:



Lamport- und Vektor-Zeitstempel



Für das Ereignis mit Vektorzeitstempel (0,4,5,2):

Potentiell beeinflussende bzw. potentiell beeinflusste Ereignisse

Vektor-Zeitstempel: Beispiel