

# VERTEILTE SYSTEME

## Ricart & Agrawala Algorithmus

Referent - Bob Prevos

# Ricart & Agrawala Algorithmus

- Algorithmus zum verteilten, wechselseitigen Ausschluss (1981)
- Voraussetzungen:
  - Totale Ordnung auf allen "Events"
  - Nachrichten kommen immer an

# Ricart & Agrawala Algorithmus

- Algorithmus zum Zugriff auf eine Ressource:
  1. Entwerfe Nachricht mit:  
**Nachricht-ID [PC-ID, Prozess-ID], Zeitstempel**
  2. Sende Anfrage an alle anderen Prozesse in der Gruppe
  3. Warte bis jeder der Prozesse seine Zusage erteilt
  4. Ressource betreten, Zugang genehmigt

# Ricart & Agrawala Algorithmus

- Schritt 2:
  - Sende Anfrage an alle anderen Prozesse in der Gruppe
- Wie reagiert der Empfänger?

# Ricart & Agrawala Algorithmus

- Wie reagiert der Empfänger einer Anfrage?

## **3 mögliche Fälle:**

a.) Empfänger hat momentan kein Interesse an der Ressource -> Sende OK

b.) Empfänger ist bereits in der Ressource

-> Merke Anfrage in Warteschlange(FIFO) &  
antworte erst nach dem Verlassen

c.) Empfänger will Ressource auch betreten!

-> Vergleiche Zeitstempel, der Gewinner darf in die Ressource

# Lamport Zeitstempel

- Verfahren aus der Sicht eines Prozesses p:
- Sei  $Z_x(a)$  der Zeitstempel von Proz. x an Event a
  - Lokales Event a:
    - $Z_x(a) = Z_x(a) + 1$
  - Sendeevent a:
    - $Z_x(a) = Z_x(a) + 1$
    - `send(message,  $Z_x(a)$ )`
  - Empfangsevent a:
    - `receive(message,  $Z_y$ )`
    - $Z_x(a) = \max( Z_x(a), Z_y ) + 1$

# Lamport Zeitstempel

- Verfahren aus der Sicht eines Prozesses p:
- Sei  $Z_x(a)$  der Zeitstempel von Proz. x an Event a

- Lokales Event a:

- $Z_x(a) = Z_x(a) + 1$

- Sendevent a:

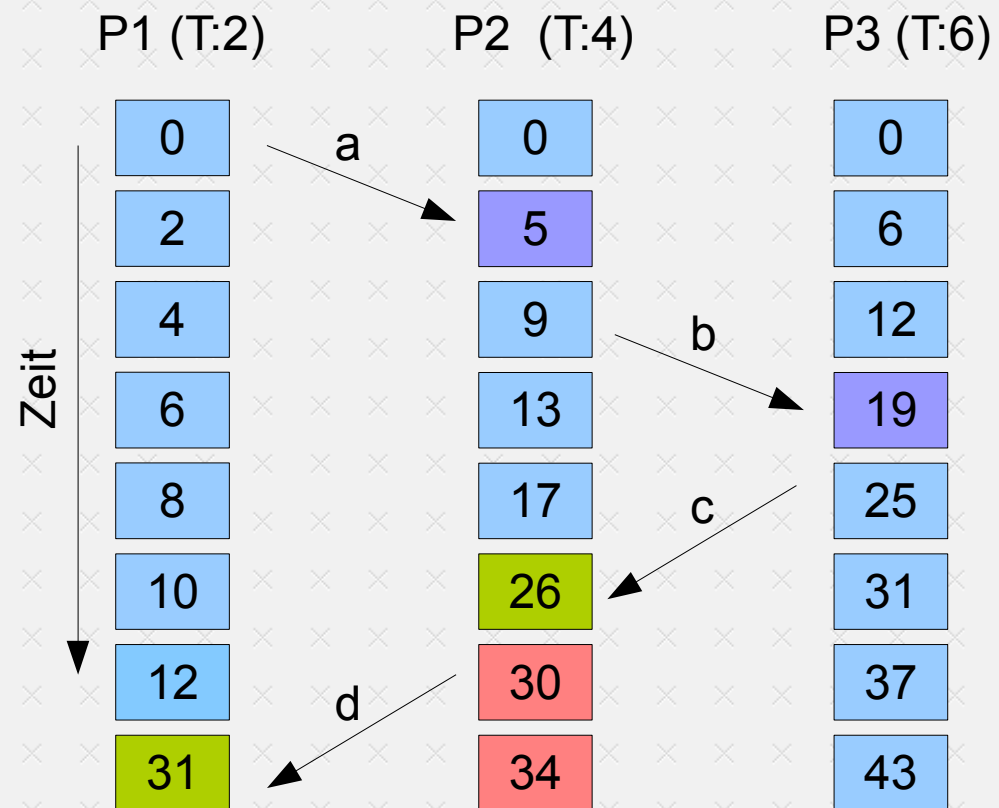
- $Z_x(a) = Z_x(a) + 1$

- $\text{send}(\text{message}, Z_x(a))$

- Empfangsevent a:

- $\text{receive}(\text{message}, Z_y)$

- $Z_x(a) = \max(Z_x(a), Z_y) + 1$



# R&A vs. Lamport Algo.

- Auf Anfragen wird bei Lamport sofort geantwortet, bei R&A erst nach dem Verlassen
  - > keine Freigabebenachrichtigung nötig
  - > anstelle von  $3(n-1)$  Nachrichten,  $2(n-1)$
- Beim Lamport Algo. hat jeder Prozess seine eigene Version der Anfragewarteschlange
  - > der Erste in der Warteschlange gewinnt!
- Beim R&A gewinnt der Prozess, der alle Zusagen erhalten hat
  - (der Gewinner selbst hält seine Zusage für die anderen zurück!)