Betriebssysteme 1

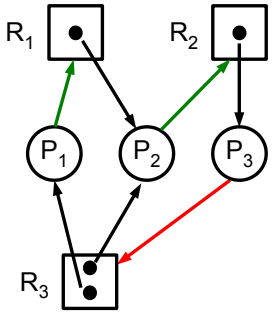
SS 2019

Prof. Dr.-Ing. Hans-Georg Eßer
Fachhochschule Südwestfalen

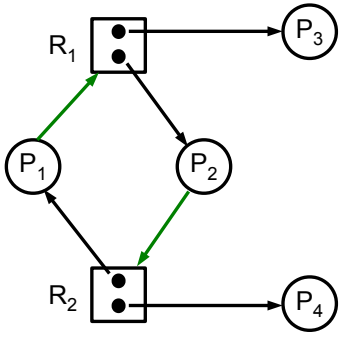
Foliensatz G: v1.0, 2019/07/04

- Deadlocks (Ergänzung zu Foliensatz E)

Graph bei Mehrfach-Ressourcen



Mit roter Kante ($P_3 \rightarrow R_3$) gibt es einen Deadlock (ohne nicht)



Kreis, aber kein Deadlock – Bedingung ist nur **notwendig**, nicht hinreichend!

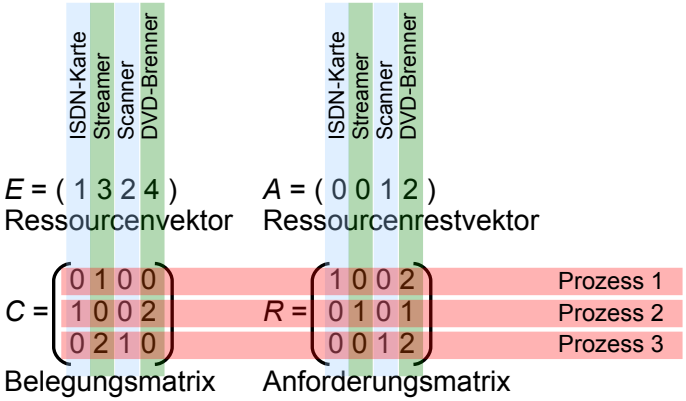
Matrix-Algorithmus (1)

- Alternative zu Graph-Verfahren nötig
- Simuliere
  - vollständige Zuteilung der angeforderten Ressourcen und
  - anschließend komplette Rückgabe
- Nutzt drei Datenstrukturen:
  - Belegungsmatrix
  - Ressourcenrestvektor
  - Anforderungsmatrix

Matrix-Algorithmus (2)

- n Prozesse P_1, ... , P_n
- m Ressourcentypen R_1, ... , R_m
  Vom Typ R_i gibt es E_i Ressourcen-Instanzen (i=1,...,m)
  → **Ressourcenvektor** $E = (E_1\ E_2\ ...\ E_m)$
- **Ressourcenrestvektor** A (wie viele sind noch frei?)
- **Belegungsmatrix** C
  C_{ij} = Anzahl Ressourcen vom Typ j, die von Prozess i belegt sind
- **Anforderungsmatrix** R
  R_{ij} = Anzahl Ressourcen vom Typ j, die Prozess i noch benötigt

Matrix-Algorithmus (3) – Beispiel



Matrix-Algorithmus (4)

Algorithmus

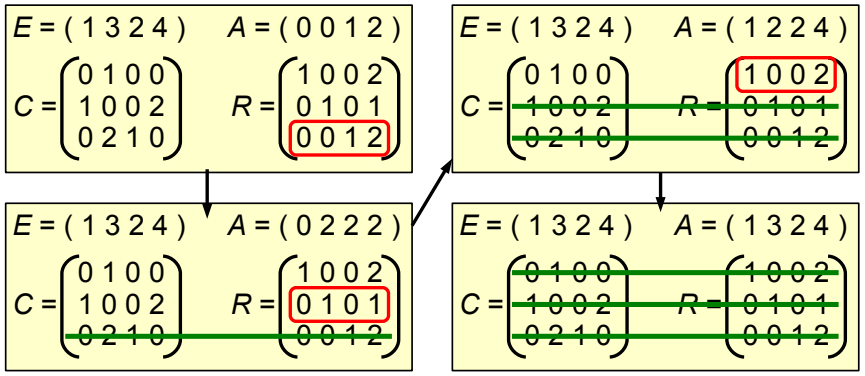
1. Suche einen unmarkierten Prozess P_i, dessen verbleibende Anforderungen vollständig erfüllbar sind, also $R_{ij} \leq A_j$ für alle j
2. Gibt es keinen solchen Prozess, beende Algorithmus
3. Ein solcher Prozess könnte erfolgreich abgearbeitet werden. Simuliere die Rückgabe aller belegten Ressourcen:

   $A := A + C_i$ (i-te Zeile von $\boldsymbol{C}$)

   Markiere den Prozess – er ist nicht Teil eines Deadlocks
4. Weiter mit Schritt 1

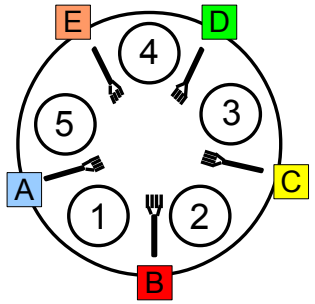
Matrix-Algorithmus (5)

Alle Prozesse, die nach diesem Algorithmus nicht markiert sind, sind an einem Deadlock beteiligt



Matrix-Algorithmus (6)

Beispiel: 5 Philosophen



A B C D E A B C D E

$E = (1\,1\,1\,1\,1)$ $A = (0\,0\,0\,0\,0)$

$$C = \begin{pmatrix} 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \end{pmatrix} \quad R = \begin{pmatrix} 0&0&0&0&1 \\ 1&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \end{pmatrix}$$

- Algorithmus bricht direkt ab
- alle Prozesse sind Teil eines Deadlocks