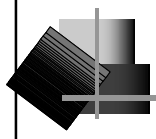
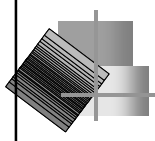


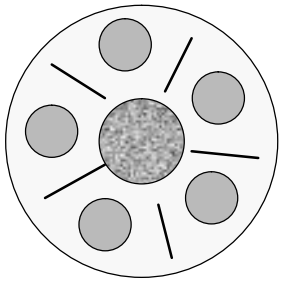
Kontrollflüsse

6. Verklemmungen

1



Dining Philosophers



- Philosophen wechseln zwischen zwei Zuständen
  - Denken
  - Essen
- Hauptgericht “Reis Pur”
  - n Sitzplätze
  - Links 1 Stäbchen
  - Philosoph braucht rechtes und linkes Stäbchen zum Essen
  - Rechts 1 Stäbchen
- Ziel: Kein Pilosoph darf verhungern
- Modell:
  - Philosoph = Prozeß
  - Synchronisation über Semaphore

2

Eine Lösung?

- Datenstrukturen

```
struct Sitzplatz {
   Boolean Frei;
   Bsemaphor Stäbchen;
   }

Sitzplatz Tisch[n];
Semaphor Warten := 0;
```

- Jeder Platz initial frei und Stäbchen vorhanden (1)
- Philosoph

```
Forever {
   Sleep(Random());
   Essen();
   }
```

```
Essen () {
   int i,k;

   k := freier_Platz();
   while (k == -1) {
      P(Warten);
      k := freier_Platz();
      }
   Tisch[k].Frei := False;
   P(Tisch[k].Stäbchen);
   P(Tisch[(k+1) mod n].Stäbchen);

   Essen ...

   V(Tisch[(k+1) mod n].Stäbchen);
   V(Tisch[k].Stäbchen);
   Tisch[k].Frei := True;
   V(Warten);
   }
```

3

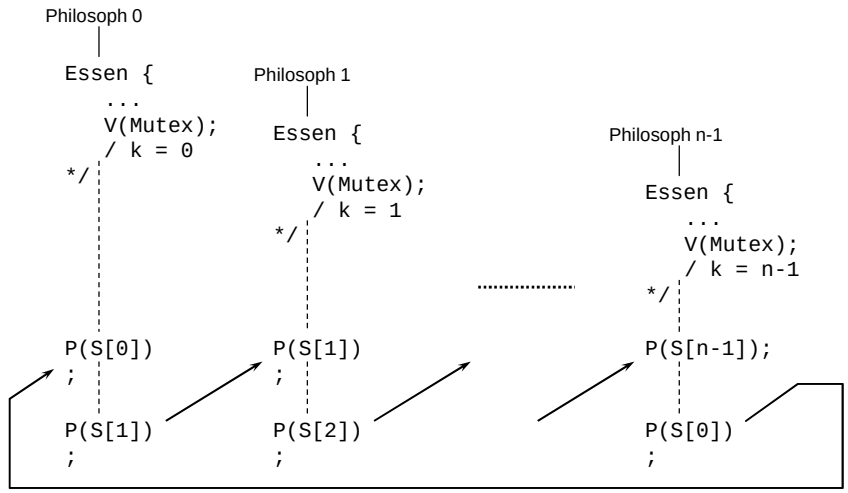
Zweiter Anlauf

- Fehler 1
  - Potentiell setzen sich mehrere Philosophen auf einen Stuhl
  - Wechselseitiger Ausschluß bei der Suche nach freiem Stuhl
- Fehler 2
  - Potentiell Verklemmung

```
Bsemaphor Mutex := 1;

Essen () {
   int k;

   P(Mutex);
   k := freier_Platz();
   while (k == -1) {
      P(Warten);
      k := freier_Platz();
      }
   Tisch[k].Frei := False;
   V(Mutex);
   P(Tisch[k].Stäbchen);
   P(Tisch[(k+1) mod n].Stäbchen);

   Essen ...

   V(Tisch[(k+1) mod n].Stäbchen);
   V(Tisch[k].Stäbchen);
   Tisch[k].Frei := True;
   V(Warten);
   }
```

4

Die Katastrophe



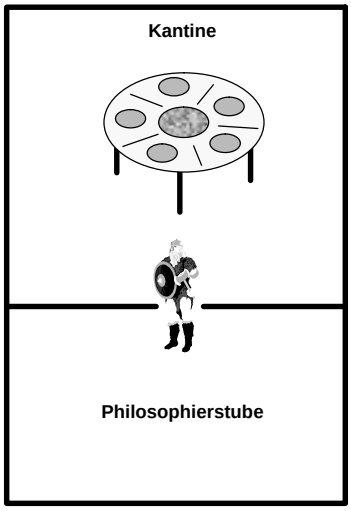
5

Verklemmungsfreie Lösungen

- Bei n Plätzen maximal n-1 Philosophen gleichzeitig am Tisch
  - Mindestens 1 Philosoph hat 2 Stäbchen
- Zyklus aufbrechen
  - Philosophen auf “geraden” Stühlen nehmen zuerst linkes Stäbchen
  - Philosophen auf “ungeraden” Stühlen nehmen zuerst rechtes Stäbchen
- Philosoph nimmt Stäbchen nur wenn beide frei sind
  - Testen und Aufnehmen in einem kritischen Abschnitt
  - Gesonderte Warteliste bei Fehlschlag
- …

6

Türsteher-Variante



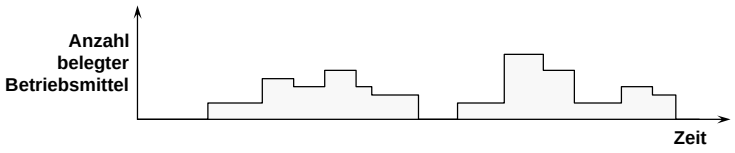

```
Semaphor Türsteher := n-1;

Essen () {
   int k;

   P(Türsteher);
   P(Mutex);
   k := freier_Platz();
   Tisch[k].Frei := False;
   V(Mutex);

   Essen ...

   Tisch[k].Frei := True;
   V(Warten);
   V(Türsteher);
   }
```

- Braucht man noch das Semaphor mutex?

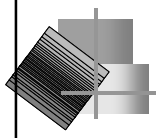
7

System-Modell

- Mehrere Betriebsmittelarten
  - z.B. Prozessor, Speicherseite, Datei, ...
  - Pro Betriebsmittelart bestimmte Anzahl an Instanzen
  - Instanzen einer Art völlig identisch
    - Keine Belegung einer bestimmten Instanz
- Typische Betriebsmittelverwendung durch Prozesse
  - Prozesse fordern Betriebsmittel an (Request)
    - Blockade, wenn keine freie Ressource belegt werden kann
  - Nutzung des Betriebsmittels (Use)
  - Freigabe des belegten Betriebsmittels (Release)



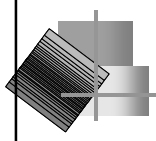
8



Formale Modelle

- Beschreibung von Verklemmungszuständen
  - Eigenschaften
  - Auswirkungen
  - Erkennen von Verklemmungen
  - Beheben von Verklemmungen
- Zwei Modelle
- Resource Allocation Graph
  - Strukturorientiert
  - Schnappschuß
  - Prozesse, Betriebsmittel
  - Belegt, Besitzt
- Modell von Holt
  - Zustandsorientiert
  - Vollständig
  - Systemzustände
  - Zustandsübergänge

9



Resource Allocation Graphs

- Formale Beschreibung von Verklemmungszuständen
- Graphen G = (V,E)
- Knoten

$$V = P \cup R$$

$$P = \{P_1, \ldots, P_n\}$$

$$R = \{R_1, \ldots, R_m\}$$

  - P = Menge von Prozessen
  - R = Menge von Betriebsmittelarten
- Kanten
  - Prozeß fordert Betriebsmittel an (Anforderungskante)

$$p \in P, r \in R: \quad p \rightarrow r \in E$$

  - Prozeß besitzt Betriebsmittel (Besitzt-Kante)

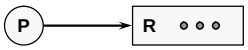
$$p \in P, r \in R: \quad r \rightarrow p \in E$$

10

Graphische Darstellung

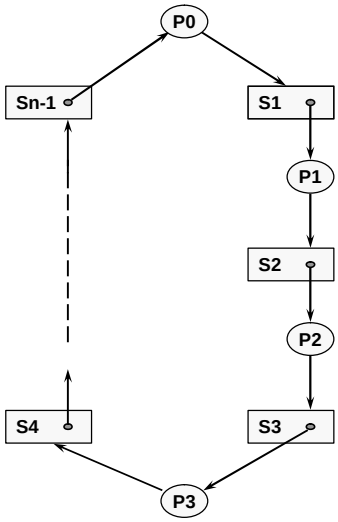
- Prozesse

- Betriebsmittelarten
  - Punkt = 1 Betriebsmittel



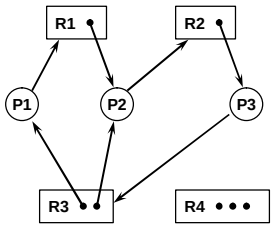
- Anforderungskante

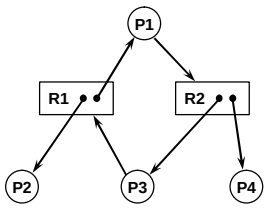
- Besitzt-Kante



11

Verklemmungen in RAG´s





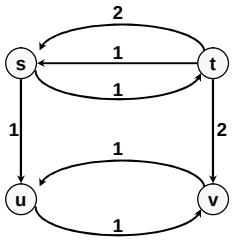
- Prozeßsystem verklemmt ⇒ RAG enthält Zyklus
- RAG enthält Zyklus ⇒ Prozeßsystem verklemmt?
  - Ja, bei nur einer Instanz pro Betriebsmittelart
  - Ja, bei nur einer Instanz der am Zyklus beteiligten Betriebsmittelarten
  - Möglicherweise, bei mehreren Instanzen einer Betriebsmittelart
- Verklemmung trotz mehrerer Betriebsmittel pro Art erkennen?

12

Formales Modell nach Holt (1972)

- Prozeßsystem S = (Z,P)
  - Z = Menge von Systemzuständen
  - P = Prozeßmenge
- Jeder Prozeß führt Zustandsübergänge aus

$$s \xrightarrow{p} t \quad s \in Z, t \in Z, p \in P$$



- Folgen von Zustandsübergängen

$$s \xrightarrow{i} t, t \xrightarrow{j} u, \ldots, v \xrightarrow{x} w \quad = \quad s \xrightarrow{*} w$$

13

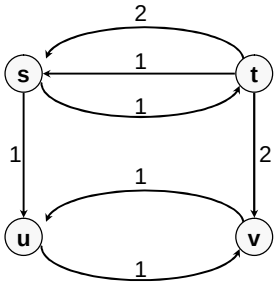
Ausgezeichnete Systemzustände

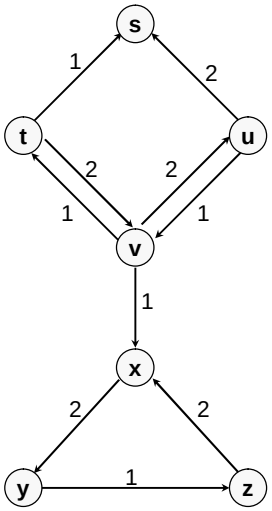
- Blockierter Prozeß
  - Ein Prozeß P_k ist im Zustand s blockiert, wenn kein Zustand t mit
    $s \xrightarrow{k} t$
    existiert
- Verklemmter Zustand
  - Ein Prozeß P_k ist im Zustand s verklemmt, wenn er für alle Übergänge
    $s \xrightarrow{*} t$
    im Zustand t blockiert ist (P_k kann nicht mehr befreit werden)
- Sicherer Zustand
  - Zustand s ist sicher, wenn für alle Zustände t mit
    $s \xrightarrow{*} t$
    t in keinem Fall ein Verklemmungszustand ist

14

Beispiele

- Blockierte Zustände?
- Verklemmte Zustände?
- Sichere Zustände?





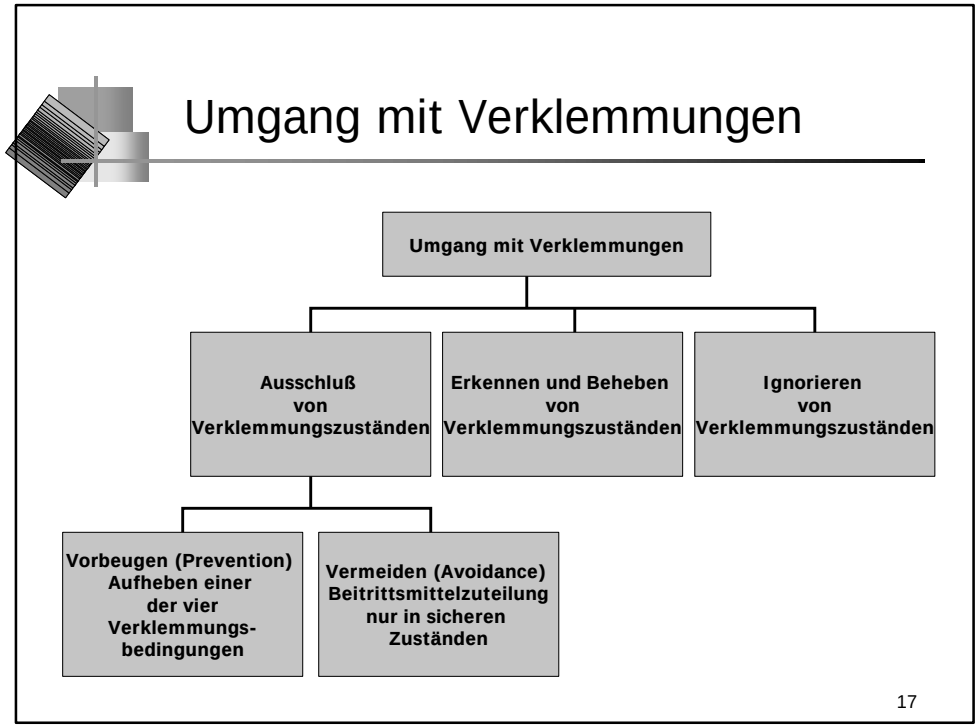
15

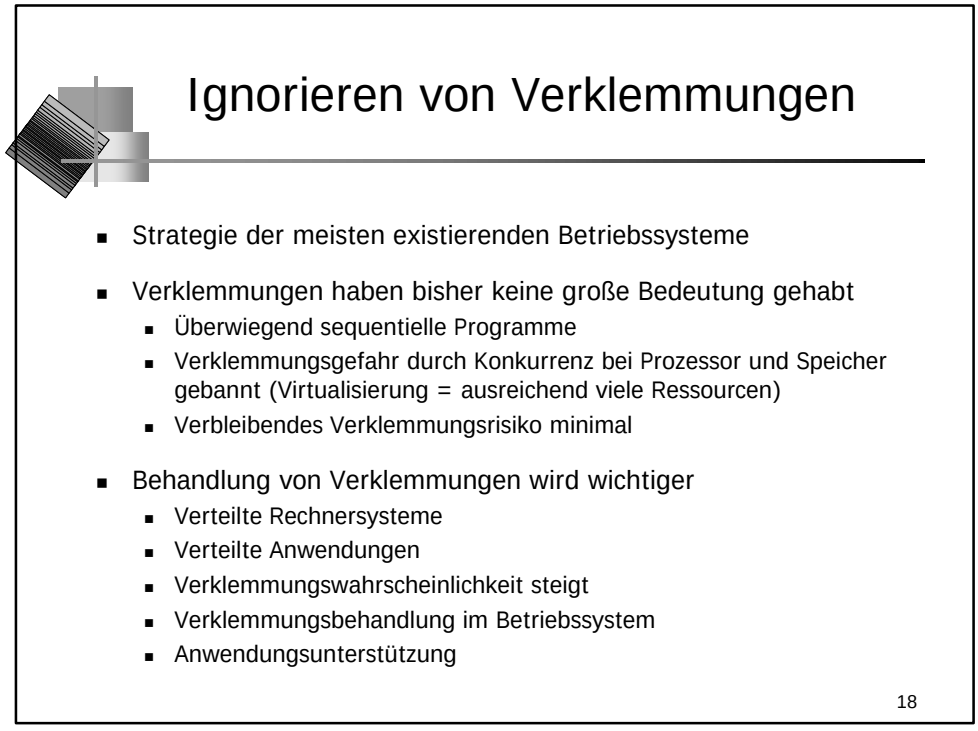
Bedingungen für eine Verklemmung

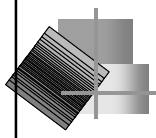
- WA:
  Wechselseitiger Ausschluß bei der Nutzung von Betriebsmitteln (exklusive Nutzung)
- BF:
  Prozesse besitzen Betriebsmittel und fordern weitere an
- KE:
  Betriebsmittel können nicht zwangsweise entzogen werden
- ZK:
  Es existiert eine zirkuläre Prozeßkette, in der jeder Prozeß ein oder mehrere Betriebsmittel besitzt, die vom nächsten Prozeß im Zyklus angefordert werden

Alle 4 Bedingungen müssen gleichzeitig gelten!

16


Umgang mit Verklemmungen
Umgang mit Verklemmungen
Ausschluß von Verklemmungszuständen
Erkennen und Beheben von Verklemmungszuständen
Ignorieren von Verklemmungszuständen
Vorbeugen (Prevention) Aufheben einer der vier Verklemmungs-bedingungen
Vermeiden (Avoidance) Beitrittsmittelzuteilung nur in sicheren Zuständen
17


Ignorieren von Verklemmungen
Strategie der meisten existierenden Betriebssysteme
Verklemmungen haben bisher keine große Bedeutung gehabt
Überwiegend sequentielle Programme
Verklemmungsgefahr durch Konkurrenz bei Prozessor und Speicher gebannt (Virtualisierung = ausreichend viele Ressourcen)
Verbleibendes Verklemmungsrisiko minimal
Behandlung von Verklemmungen wird wichtiger
Verteilte Rechnersysteme
Verteilte Anwendungen
Verklemmungswahrscheinlichkeit steigt
Verklemmungsbehandlung im Betriebssystem
Anwendungsunterstützung
18

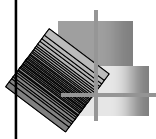


Vorbeugen (Deadlock Prevention)

Aufheben einer Grundbedingung

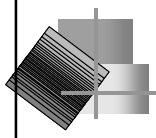
- Wechselseitiger Ausschluß (WA):
  - Bereitstellen ausreichend vieler Betriebsmittelinstanzen
  - Physisch oder Virtuell
- Besitzen und Fordern (BF):
  - Anfordern aller benötigten Betriebsmittel bei Prozeßstart
  - Anfordern zusätzlicher Betriebsmittel nur bei vorheriger Freigabe der zugeteilten Betriebsmittel
  - Nachteile
    - Reduzierte Betriebsmittelauslastung
    - Aushungerung einzelner Prozesse

19



- Kein zwangsweiser Entzug (KE):
  - Wenn möglich, Entzug von Betriebsmitteln, z.B. Prozessor, Speicherseiten
- Zirkuläre Prozeßkette (ZK):
  - Totale Ordnung auf allen Betriebsmitteln definieren
  - Betriebsmittelanforderungen nur aufsteigend bzgl. Ordnung
  - Warum funktioniert das?

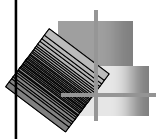
20



Erkennen von Verklemmungen

- Voraussetzungen
  - Betriebsmittelarten und jeweilige Betriebsmittelanzahl bekannt
  - Aktuell belegte Betriebsmittel pro Prozeß
  - Betriebsmittelanforderungen pro Prozeß
- Existiert eine Prozeßteilmenge, die sich in einer Verklemmung befindet?
- Algorithmus
  - Ermittlung einer Aktivierungsreihenfolge aller Prozesse, so daß deren Anforderungen jeweils erfüllt werden können?
    - Ja: Keine Verklemmung
    - Nein: Verklemmung

21



Datenstrukturen

- Betriebsmittelvorrat:

$$B = \left(bm_1, bm_2, \ldots, bm_k\right)$$

- Belegte Betriebsmittel pro Prozeß:

$$P_v = \left(p_{v,1}, \ldots, p_{v,i}, \ldots, p_{v,k}\right) \quad P = \begin{pmatrix} P_1 \\ \vdots \\ P_n \end{pmatrix}$$

- Angeforderte Betriebsmittel pro Prozeß:

$$Q_v = \left(q_{v,1}, \ldots, q_{v,i}, \ldots, q_{v,k}\right) \qquad Q = \begin{pmatrix} Q_1 \\ \vdots \\ Q_n \end{pmatrix}$$

- Verfügbare Betriebsmittel:

$$V = \left(v_1, v_2, \ldots, v_k\right), \quad v_j := bm_j - \sum_{i=1}^{n} p_{i,j}$$

22

Erkennungsalgorithmus

```
Boolean Erkennen (P,Q,V) {
   W := V;
   Stop := False;
   for i:=1 to n {
      if (P[i]=0)
         Markiere P[i];
      }
   do {
      if (Unmarkierte Zeile u mit Q[u]<W) {
         /* Anforderungen von Prozeß u erfüllbar */
         Markiere P[u];
         W := W+P[u];
         }
      else
         Stop := True;
      } until Stop;
   if (Unmarkierte Zeilen in P) return True else return False;
   }
```

- Verklemmung ⇔ Unmarkierte Zeilen in P bei Terminierung
  ⇔ Erkennen(P,Q,V)=False

23

Beispiel

- Betriebsmittelarten B und verfügbarer Vorrat

$$B = (12 \quad 8 \quad 40), \quad V = (2 \quad 1 \quad 9)$$

- Aktuelle Belegungs- (P) und Anforderungsmatrix (Q)

$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} \qquad Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 0 & 2 \\ 5 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

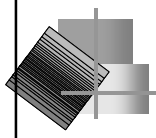
- Anforderungen erfüllbar, z.B. Prozeß 4

$$P_4 = (5 \quad 0 \quad 9)$$

$$Q_4 = (5 \quad 6 \quad 7)$$

$$P_4 + Q_4 = (10 \quad 6 \quad 16) \leq (12 \quad 8 \quad 40)$$

24



Ablaufprotokoll (1)

Iteration 1

$$R = (12 \quad 8 \quad 40)$$
$$V = (2 \quad 1 \quad 9)$$
$$W = (2 \quad 1 \quad 9)$$
$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix}$$
$$Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 0 & 2 \\ 5 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

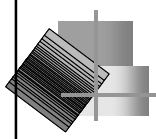
Iteration 2

$$R = (12 \quad 8 \quad 40)$$
$$V = (2 \quad 1 \quad 9)$$
$$W = (3 \quad 2 \quad 14)$$
$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} *$$
$$Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 0 & 2 \\ 5 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

Iteration 3

$$R = (12 \quad 8 \quad 40)$$
$$V = (2 \quad 1 \quad 9)$$
$$W = (5 \quad 4 \quad 21)$$
$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} **$$
$$Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 0 & 2 \\ 5 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

25



Ablaufprotokoll (2)

Iteration 4

$$R = (12 \quad 8 \quad 40)$$
$$V = (2 \quad 1 \quad 9)$$
$$W = (5 \quad 7 \quad 24)$$
$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} ***$$
$$Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 0 & 2 \\ 5 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

Iteration 5

$$R = (12 \quad 8 \quad 40)$$
$$V = (2 \quad 1 \quad 9)$$
$$W = (10 \quad 7 \quad 33)$$
$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} ***$$
$$Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 0 & 2 \\ 5 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

Iteration 6

$$R = (12 \quad 8 \quad 40)$$
$$V = (2 \quad 1 \quad 9)$$
$$W = (12 \quad 8 \quad 40)$$
$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} *****$$
$$Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 0 & 2 \\ 5 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

26

Beispiel 2

- Betriebsmittelarten B und verfügbarer Vorrat

$$B = (12 \quad 8 \quad 40), \quad V = (2 \quad 1 \quad 9)$$

- Aktuelle Belegungs- (P) und Anforderungsmatrix (Q)

$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} \qquad Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 1 & 3 \\ 6 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

- Anforderungen erfüllbar, z.B. Prozeß 4

$$P_4 = (5 \quad 0 \quad 9)$$
$$Q_4 = (6 \quad 6 \quad 7)$$
$$P_4 + Q_4 = (11 \quad 6 \quad 16) \leq (12 \quad 8 \quad 40)$$

27

Ablaufprotokoll (1)

$R = (12 \quad 8 \quad 40)$
$V = (2 \quad 1 \quad 9)$
$W = (2 \quad 1 \quad 9)$

$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} \qquad Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 1 & 3 \\ 6 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

Iteration 1

$R = (12 \quad 8 \quad 40)$
$V = (2 \quad 1 \quad 9)$
$W = (3 \quad 2 \quad 14)$

$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} * \qquad Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 1 & 3 \\ 6 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

Iteration 2

$R = (12 \quad 8 \quad 40)$
$V = (2 \quad 1 \quad 9)$
$W = (5 \quad 4 \quad 21)$

$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} * \qquad Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 1 & 3 \\ 6 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

Iteration 3

28

Ablaufprotokoll (2)

$$R = (12 \quad 8 \quad 40)$$
$$V = (2 \quad 1 \quad 9)$$
$$W = (5 \quad 7 \quad 24)$$
$$P = \begin{pmatrix} 2 & 1 & 7 \\ 0 & 3 & 3 \\ 1 & 1 & 5 \\ 5 & 0 & 9 \\ 2 & 2 & 7 \end{pmatrix} \begin{matrix} \| \\ * \\ * \\ \| \\ * \end{matrix}$$
$$Q = \begin{pmatrix} 7 & 3 & 10 \\ 4 & 4 & 3 \\ 1 & 1 & 3 \\ 6 & 6 & 7 \\ 3 & 2 & 2 \end{pmatrix}$$

Iteration 4

- Anforderungen der Prozesse 1 und 4 nicht erfüllbar
- Verklemmung

29

Vermeiden von Verklemmungen (Avoidance)

- Voraussetzung
  - Maximalanforderung an Betriebsmitteln pro Prozeß bekannt (Matrix A)
  - Alle Betriebsmittelanforderungen werden überprüft
- Kann aktuelle Betriebsmittelanforderung potentiell zu einer Verklemmung führen?
  - Ja: Anforderung wird zurückgewiesen
  - Nein: Anforderung wird erlaubt
- Algorithmus
  - Aktuelle Betriebsmittelanforderung “scheinbar” durchführen
  - Resultierenden Zustand mit Erkennungsalgorithmus überprüfen
    - Annahme: Alle Prozesse fordern bis zu ihrem Maximalbedarf
  - Bei Deadlock Zuteilung zurückweisen
- Algorithmus ist pessimistisch

30

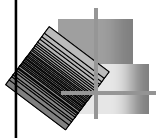
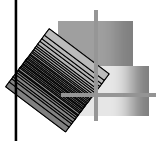


Abbildung auf Erkennungsalgorithmus

```
Boolean Vermeiden (P,A,V,qx) {
   for i:=1 to n {
      if (i=x)
         P'[x]:= P[x]+qx;
      else
         P'[x]:= P[x];
      }
   Q':= A-P';
   V':= V-qx;
   Deadlock := Erkennen(P',Q',V');
   if (Deadlock)
      return False;
   else
      return True;
   }
```

- Anforderung von Prozeß x ablehnen ⇔ Vermeiden(P,A,V,x)=False

31



Beispiel

- Betriebsmittelarten B und verfügbarer Vorrat

$$B = (17 \quad 46 \quad 29), \quad V = (10 \quad 10 \quad 16)$$

- Aktuelle Belegungs- (P) und Maximalanforderungen (A)

$$P = \begin{pmatrix} 2 & 2 & 6 \\ 0 & 9 & 1 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} \qquad A = \begin{pmatrix} 9 & 7 & 13 \\ 5 & 23 & 9 \\ 3 & 18 & 11 \\ 12 & 13 & 6 \\ 2 & 30 & 7 \end{pmatrix}$$

- Aktuelle Anforderung von Prozeß 2

$$q = (2 \quad 5 \quad 3)$$

32

Transformation

- P':

$$q = (2 \quad 5 \quad 3) + \begin{pmatrix} 2 & 2 & 6 \\ 0 & 9 & 1 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} \longrightarrow \begin{pmatrix} 2 & 2 & 6 \\ 2 & 14 & 4 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} = P'$$

- Q':

$$Q' := A - P' = \begin{pmatrix} 9 & 7 & 13 \\ 5 & 23 & 9 \\ 3 & 18 & 11 \\ 12 & 13 & 6 \\ 2 & 30 & 7 \end{pmatrix} - \begin{pmatrix} 2 & 2 & 6 \\ 2 & 14 & 4 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} = \begin{pmatrix} 7 & 5 & 7 \\ 3 & 9 & 5 \\ 3 & 16 & 10 \\ 8 & 5 & 4 \\ 1 & 15 & 4 \end{pmatrix}$$

- V':

$$V' := V - q = (10 \quad 10 \quad 16) - (2 \quad 5 \quad 3) = (8 \quad 5 \quad 13)$$

33

Eingabe für Deadlock-Erkennung

- Betriebsmittelarten B und verfügbarer Vorrat

$$B = (17 \quad 46 \quad 29), \quad V = (8 \quad 5 \quad 13)$$

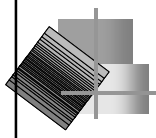
- Aktuelle Belegungs- (P) und Anforderungsmatrix (Q)

$$P = \begin{pmatrix} 2 & 2 & 6 \\ 2 & 14 & 4 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} \qquad Q = \begin{pmatrix} 7 & 5 & 7 \\ 3 & 9 & 5 \\ 3 & 16 & 10 \\ 8 & 5 & 4 \\ 1 & 15 & 4 \end{pmatrix}$$

- Anforderungen erfüllbar, z.B. Prozeß 4

$$P_4 = (4 \quad 8 \quad 2)$$
$$Q_4 = (8 \quad 5 \quad 4)$$
$$P_4 + Q_4 = (12 \quad 13 \quad 6) \leq (17 \quad 46 \quad 29)$$

34



Ablaufprotokoll (1)

Iteration 1

$$R = (17 \quad 46 \quad 29)$$
$$V = (8 \quad 5 \quad 13)$$
$$W = (8 \quad 5 \quad 13)$$
$$P = \begin{pmatrix} 2 & 2 & 6 \\ 2 & 14 & 4 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix}$$
$$Q = \begin{pmatrix} 7 & 5 & 7 \\ 3 & 9 & 5 \\ 3 & 16 & 10 \\ 8 & 5 & 4 \\ 1 & 15 & 4 \end{pmatrix}$$

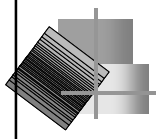
Iteration 2

$$R = (17 \quad 46 \quad 29)$$
$$V = (8 \quad 5 \quad 13)$$
$$W = (10 \quad 7 \quad 19)$$
$$P = \begin{pmatrix} 2 & 2 & 6 \\ 2 & 14 & 4 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} *$$
$$Q = \begin{pmatrix} 7 & 5 & 7 \\ 3 & 9 & 5 \\ 3 & 16 & 10 \\ 8 & 5 & 4 \\ 1 & 15 & 4 \end{pmatrix}$$

Iteration 3

$$R = (17 \quad 46 \quad 29)$$
$$V = (8 \quad 5 \quad 13)$$
$$W = (14 \quad 15 \quad 21)$$
$$P = \begin{pmatrix} 2 & 2 & 6 \\ 2 & 14 & 4 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} \begin{matrix} * \\ \\ \\ * \\ \end{matrix}$$
$$Q = \begin{pmatrix} 7 & 5 & 7 \\ 3 & 9 & 5 \\ 3 & 16 & 10 \\ 8 & 5 & 4 \\ 1 & 15 & 4 \end{pmatrix}$$

35



Ablaufprotokoll (2)

Iteration 4

$$R = (17 \quad 46 \quad 29)$$
$$V = (8 \quad 5 \quad 13)$$
$$W = (16 \quad 29 \quad 25)$$
$$P = \begin{pmatrix} 2 & 2 & 6 \\ 2 & 14 & 4 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} \begin{matrix} * \\ * \\ \\ * \\ \end{matrix}$$
$$Q = \begin{pmatrix} 7 & 5 & 7 \\ 3 & 9 & 5 \\ 3 & 16 & 10 \\ 8 & 5 & 4 \\ 1 & 15 & 4 \end{pmatrix}$$

Iteration 5

$$R = (17 \quad 46 \quad 29)$$
$$V = (8 \quad 5 \quad 13)$$
$$W = (16 \quad 31 \quad 26)$$
$$P = \begin{pmatrix} 2 & 2 & 6 \\ 2 & 14 & 4 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} \begin{matrix} * \\ * \\ * \\ * \\ \end{matrix}$$
$$Q = \begin{pmatrix} 7 & 5 & 7 \\ 3 & 9 & 5 \\ 3 & 16 & 10 \\ 8 & 5 & 4 \\ 1 & 15 & 4 \end{pmatrix}$$

Iteration 6

$$R = (17 \quad 46 \quad 29)$$
$$V = (8 \quad 5 \quad 13)$$
$$W = (17 \quad 46 \quad 29)$$
$$P = \begin{pmatrix} 2 & 2 & 6 \\ 2 & 14 & 4 \\ 0 & 2 & 1 \\ 4 & 8 & 2 \\ 1 & 15 & 3 \end{pmatrix} \begin{matrix} * \\ * \\ * \\ * \\ * \end{matrix}$$
$$Q = \begin{pmatrix} 7 & 5 & 7 \\ 3 & 9 & 5 \\ 3 & 16 & 10 \\ 8 & 5 & 4 \\ 1 & 15 & 4 \end{pmatrix}$$

36