

Tutorat 12

Deadlocks und Bankier-Algorithmus

Vorbereitung

Vorbereitung

Ressourcendiagramm

- **Ressourcenspur** gibt eine **Ausführungsreihenfolge** der beiden **Prozesse** auf einem **pseudoparallelen** Prozessor an
- Die **Ressourcenspur** darf vom Start zum Ende nur nach **rechts** und nach **oben** entlang des Gitters verlaufen
- Eine **Ressourcenspur** kann ein Rechteck nur am **linken** und **unteren** Rand berühren, da die Prozesse ihre jeweiligen **Operationen** erst beim **Überschreiten** eines Zeitpunktes **vollständig** ausführen
- Läuft die **Ressourcenspur** in eine **Ecke** hinein, von der die Spur **weder** nach **rechts** noch nach **oben** fortgesetzt werden kann, ohne ein Rechteck zu betreten, so befinden sich die Prozesse in einem **Deadlock**

Vorbereitung

Belegungs-Anforderungs-Graph

- Wenn eine Ressource von einem Prozess **gesperrt** ist, zeichnet man den Pfeil von der **Ressource** zum **Prozess**
- Wenn ein Prozess eine Ressource anfordert und **blockiert**, weil die Ressource nicht verfügbar ist, dann zeichnet man den Pfeil vom **Prozess** zur **Ressource**.
 - dieser Pfeil wird verwendet, sobald von einer Resource **zwei** Pfeil ausgehen
- Pfeile lassen sich **in Worten** ausdrücken als "**belegt von**" und "**blockiert von**"

Vorbereitung

Bankier-Algorithmus

- Vektor verfügbarer Ressourcen
- Maximalanforderungsmatrix M

$$V = \begin{pmatrix} 4 & 2 & 3 & 1 \end{pmatrix} \begin{matrix} \text{Ress. 1} \\ \text{Ress. 2} \\ \text{Ress. 3} \\ \text{Ress. 4} \end{matrix}$$

$$M = \begin{pmatrix} 2 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 2 & 2 & 2 & 0 \end{pmatrix} \begin{matrix} \text{Prozess 1} \\ \text{Prozess 2} \\ \text{Prozess 3} \end{matrix}$$

- Aktuelle Belegungsmatrix E

$$E = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \\ \color{red}{2} & \color{red}{1} & \color{red}{3} & \color{red}{1} \end{pmatrix} \begin{matrix} \text{Prozess 1} \\ \text{Prozess 2} \\ \text{Prozess 3} \end{matrix}$$

- Aktuelle Restanforderungsmatrix A
- $$A_{ik} = M_{ik} - E_{ik}$$

$$A = \begin{pmatrix} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{pmatrix} \begin{matrix} \text{Prozess 1} \\ \text{Prozess 2} \\ \text{Prozess 3} \end{matrix}$$

- Ressourcenrestvektor

$$F_k = V_k - \sum_{i=1}^n E_{ik}$$

$$F = \begin{pmatrix} 2 & 1 & 0 & 0 \end{pmatrix}$$

Vorbereitung

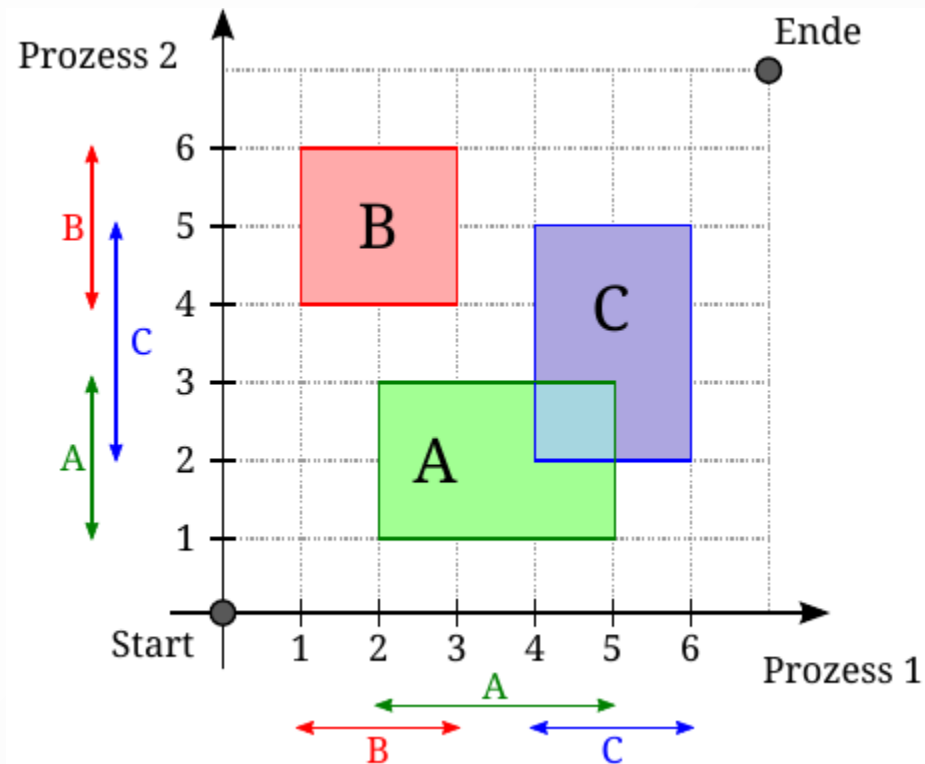
Bankier-Algorithmus

- es geht **nicht** darum einen **Deadlock** zu finden, es geht darum zu sagen, ob man in einem **sicheren** oder **unsicheren** Zustand ist, der **möglicherweise** zu einem **Deadlock** führen könnte

Übungsblatt

Übungsblatt

Aufgabe 1 a)



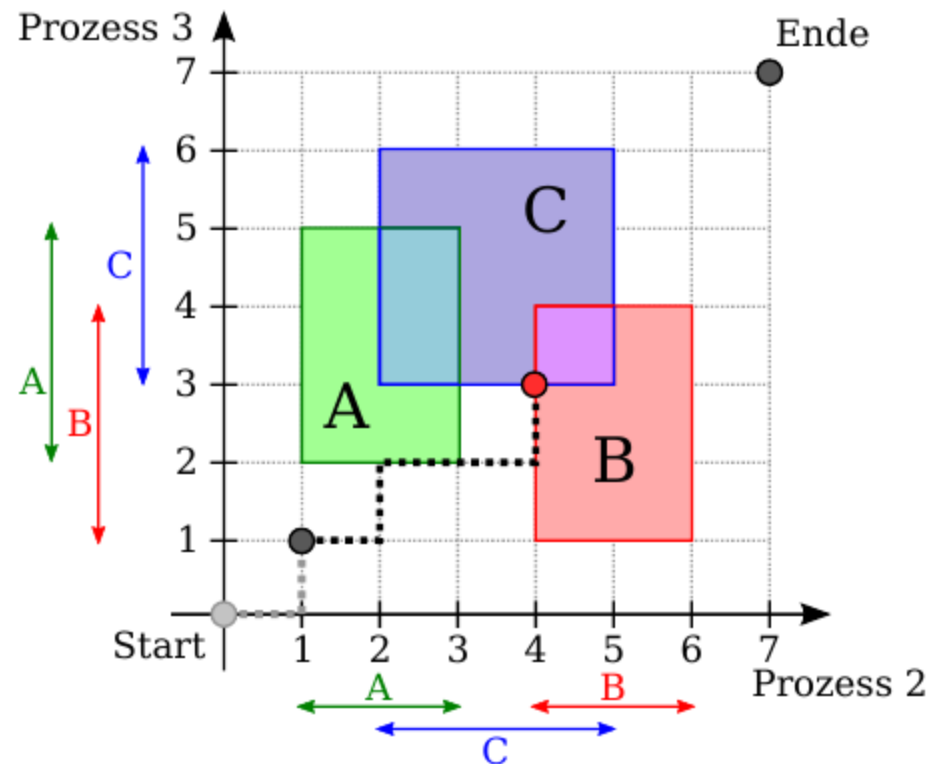
Übungsblatt

Aufgabe 1 b)

- Es gibt keine **konkave** Ecke, in die die Ressourcenspur hineinlaufen kann und nicht mehr nach **rechts** oder **oben** fortgesetzt werden kann:
 - Ressourcenspur kann sich nur nach **oben** oder **rechts** bewegen, da wegen **Pseudoparallelität** immer einer der beiden Prozesse, der gerade die CPU zugewiesen bekommen hat, Instructions ausführt, sich also **vorwärts** in der Zeit bewegt
 - Da es hier zu keinem Zeitpunkt **konkaven** Ecken oder **Sackgassen** gibt, gibt es immer einen Weg, den man nach **oben** oder **rechts** ausweichen kann → es ist zu jedem erreichbaren **Zeitpunkt** einer der beiden **Prozesse ausführbar**
 - Falls die **Instruction** des einen Prozesses **blockiert** ist, kann dafür immer der **andere Prozess** eine **Instruction** ausführen. Es gibt immer einen **Ausweg**, wie sich die beiden Prozesse **nicht** gegenseitig **blockieren**

Übungsblatt

Aufgabe 1 c)



Übungsblatt

Aufgabe 1 d)

- **Befehlsfolge, die zum Deadlock führt:**
 - **1:** *Prozess 2* Zeile 1: Belegt A
 - **2: Prozess 3** Zeile 1: Belegt B
 - **3: Prozess 3** Zeile 2: Fordert A an und blockiert, da belegt von *Prozess 2*
 - **4:** *Prozess 2* Zeile 2: Belegt C
 - **5:** *Prozess 2* Zeile 3: Gibt A frei
 - **6:** *Prozess 2* Zeile 4: Fordert B an und blockiert, da belegt von **Prozess 3**
 - **7: Prozess 3** Zeile 2: bekommt A nun zugeteilt
 - **8: Prozess 3** Zeile 3: Fordert C an und blockiert, da belegt von *Prozess 2*

Übungsblatt

Aufgabe 1 d)

Anmerkungen

- wobei man bei $(1, 1)$ anfängt, bei der **Resourcenspur** die Tupel aus **Prozess** und **Zeilennummer** anzugeben, da es keine 0te Zeile gibt.
 - Prozesse tun ihre jeweiligen **Operationen** erst beim **Überschreiten** eines Zeitpunktes **vollständig** ausführen. Daher kann eine **Resourcenspur** die Rechtecke auch am **linken** und **oberen** Rand berühren
 - der Schritt von 0 zu 1 bedeutet, man ist bei Zeile 1, hat sie aber noch **nicht** ausgeführt
- am **Ende** landet man immer in der **rechten oberen Ecke**, da man ja alle Prozesse ausgeführt haben will

Übungsblatt

Aufgabe 1 e)

Prozess 1

- 1: Anforderung B
- 2: Anforderung A
- 3: Freigabe B
- 4: Anforderung C
- 5: Freigabe A
- 6: Freigabe C

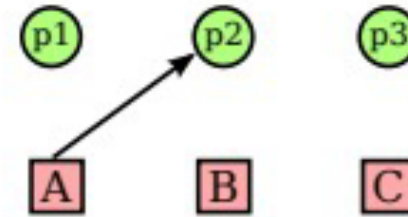
Prozess 2

- 1: Anforderung A
- 2: Anforderung C
- 3: Freigabe A
- 4: Anforderung B
- 5: Freigabe C
- 6: Freigabe B

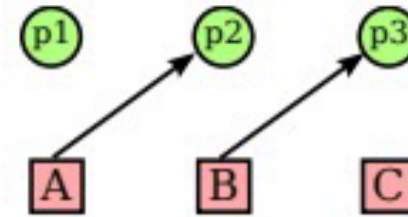
Prozess 3

- 1: Anforderung B
- 2: Anforderung A
- 3: Anforderung C
- 4: Freigabe B
- 5: Freigabe A
- 6: Freigabe C

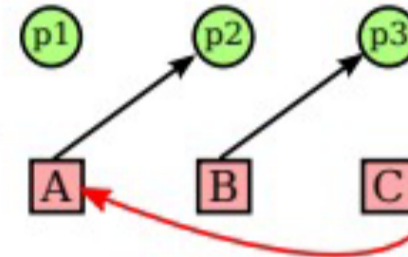
I) p2 belegt A



II) p3 belegt B



III) p3 fordert A an und blockiert



Übungsblatt

Aufgabe 1 e)

Prozess 1

- 1: Anforderung B
- 2: Anforderung A
- 3: Freigabe B
- 4: Anforderung C
- 5: Freigabe A
- 6: Freigabe C

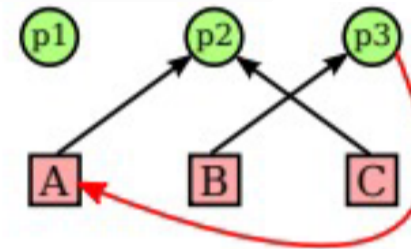
Prozess 2

- 1: Anforderung A
- 2: Anforderung C
- 3: Freigabe A
- 4: Anforderung B
- 5: Freigabe C
- 6: Freigabe B

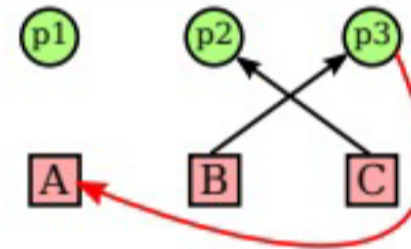
Prozess 3

- 1: Anforderung B
- 2: Anforderung A
- 3: Anforderung C
- 4: Freigabe B
- 5: Freigabe A
- 6: Freigabe C

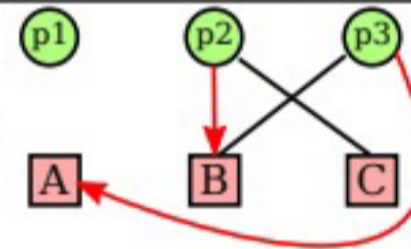
IV) p2 belegt C



V) p2 gibt A frei



VI) p2 fordert B an und blockiert



Übungsblatt

Aufgabe 1 e)

Prozess 1

- 1: Anforderung B
- 2: Anforderung A
- 3: Freigabe B
- 4: Anforderung C
- 5: Freigabe A
- 6: Freigabe C

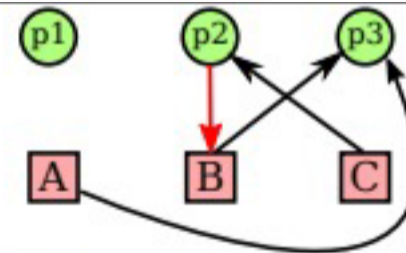
Prozess 2

- 1: Anforderung A
- 2: Anforderung C
- 3: Freigabe A
- 4: Anforderung B
- 5: Freigabe C
- 6: Freigabe B

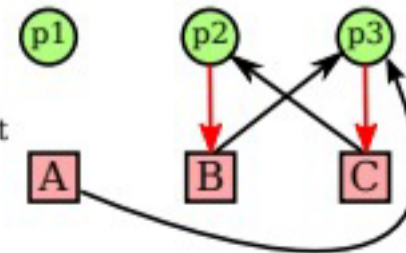
Prozess 3

- 1: Anforderung B
- 2: Anforderung A
- 3: Anforderung C
- 4: Freigabe B
- 5: Freigabe A
- 6: Freigabe C

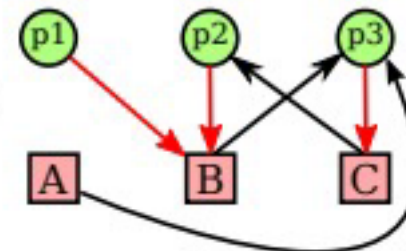
VII) p3 bekommt A nun zugeteilt



VIII) p3 fordert C an und blockiert



IX) p1 fordert B an und blockiert



Übungsblatt

Aufgabe 2 a)

Zustand Z_2

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \left(\begin{array}{ccccc} 0 & \mathbf{3} & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 & 4 \\ 2 & 2 & 0 & 0 & 2 \\ 0 & 2 & 1 & 3 & 2 \end{array} \right) & \begin{array}{l} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \left(\begin{array}{ccccc} 2 & 0 & 4 & 1 & 0 \end{array} \right) \end{array}$$

Nun kann p_3 ausgeführt werden:

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \left(\begin{array}{ccccc} 3 & 3 & 4 & 0 & 3 \\ 1 & 1 & 6 & 4 & 1 \\ 2 & 0 & 4 & 1 & 0 \\ 4 & 3 & 1 & 1 & 2 \end{array} \right) & \begin{array}{l} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \end{array}$$

Übungsblatt

Aufgabe 2 a)

Iteration 1:

Zustand während der maximalen Ressourcenanforderungen des ausgeführten Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 0 & 3 & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 & 4 \\ 4 & 2 & 4 & 1 & 2 \\ 0 & 2 & 1 & 3 & 2 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 3 & 3 & 4 & 0 & 3 \\ 1 & 1 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 1 & 1 & 2 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ (0 & 0 & 0 & 0 & 0) \end{array}$$

Zustand nach Ausführung des Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 0 & 3 & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 & 4 \\ \times & \times & \times & \times & \times \\ 0 & 2 & 1 & 3 & 2 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 3 & 3 & 4 & 0 & 3 \\ 1 & 1 & 6 & 4 & 1 \\ \times & \times & \times & \times & \times \\ 4 & 3 & 1 & 1 & 2 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ (4 & 2 & 4 & 1 & 2) \end{array}$$

Übungsblatt

Aufgabe 2 a)

Die Differenz von F zu den Zeilen der Matrix A ist

$$\begin{pmatrix} 4 & 2 & 4 & 1 & 2 \\ 4 & 2 & 4 & 1 & 2 \\ \times & \times & \times & \times & \times \\ 4 & 2 & 4 & 1 & 2 \end{pmatrix} - \begin{pmatrix} 3 & 3 & 4 & 0 & 3 \\ 1 & 1 & 6 & 4 & 1 \\ \times & \times & \times & \times & \times \\ 4 & 3 & 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 1 & -1 \\ 3 & 1 & -2 & -3 & 1 \\ \times & \times & \times & \times & \times \\ 0 & -1 & 3 & 0 & 0 \end{pmatrix}$$

- Da in jeder der drei verbleibenden relevanten Zeilen **mindestens ein** Eintrag **negativ** ist, kann **kein** weiterer Prozess mehr mit seinen **Maximalanforderungen** ausgeführt werden kann
- **in kurz:** da in jeder Zeile **mindestens ein** Eintrag von A **größer** als der entsprechende Eintrag von F ist → **unsicherer Zustand**

Übungsblatt

Aufgabe 2 b)

Zustand Z_3

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \left(\begin{array}{ccccc} 0 & 2 & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 & 4 \\ 2 & 2 & 0 & 0 & 2 \\ 0 & \mathbf{3} & 1 & 3 & 2 \end{array} \right) & \begin{array}{l} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \left(\begin{array}{ccccc} 2 & 0 & 4 & 1 & 0 \end{array} \right) \end{array}$$

Nun kann p_3 ausgeführt werden:

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \left(\begin{array}{ccccc} 3 & 4 & 4 & 0 & 3 \\ 1 & 1 & 6 & 4 & 1 \\ 2 & 0 & 4 & 1 & 0 \\ 4 & 2 & 1 & 1 & 2 \end{array} \right) & \begin{array}{l} p_1 \\ p_2 \\ p_3 \\ p_4 \end{array} \end{array}$$

Übungsblatt

Aufgabe 2 b)

Iteration 1:

Zustand während der maximalen Ressourcenanforderungen des ausgeführten Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 0 & 2 & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 & 4 \\ 4 & 2 & 4 & 1 & 2 \\ 0 & 1 & 1 & 3 & 2 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 3 & 4 & 4 & 0 & 3 \\ 1 & 1 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 4 & 2 & 1 & 1 & 2 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ (0 & 0 & 0 & 0 & 0) \end{array}$$

Zustand nach Ausführung des Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 0 & 2 & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 & 4 \\ \times & \times & \times & \times & \times \\ 0 & 3 & 1 & 3 & 2 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 3 & 4 & 4 & 0 & 3 \\ 1 & 1 & 6 & 4 & 1 \\ \times & \times & \times & \times & \times \\ 4 & 2 & 1 & 1 & 2 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ (4 & 2 & 4 & 1 & 2) \end{array}$$

Nun kann p_4 ausgeführt werden:

Übungsblatt

Aufgabe 2 b)

Iteration 2:

Zustand während der maximalen Ressourcenanforderungen des ausgeführten Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 0 & 2 & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 & 4 \\ 4 & 5 & 2 & 4 & 4 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 3 & 4 & 4 & 0 & 3 \\ 1 & 1 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ (0 & 0 & 3 & 0 & 0) \end{array}$$

Zustand nach Ausführung des Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 0 & 2 & 1 & 2 & 1 \\ 1 & 0 & 1 & 0 & 4 \\ \times & \times & \times & \times & \times \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 3 & 4 & 4 & 0 & 3 \\ 1 & 1 & 6 & 4 & 1 \\ \times & \times & \times & \times & \times \end{pmatrix} & p_1 & p_2 & p_3 & p_4 \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ (4 & 5 & 5 & 4 & 4) \end{array}$$

Nun kann p_1 ausgeführt werden:

Übungsblatt

Aufgabe 2 b)

Iteration 3:

Zustand während der maximalen Ressourcenanforderungen des ausgeführten Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 3 & 6 & 5 & 2 & 4 \\ 1 & 0 & 1 & 0 & 4 \end{pmatrix} & p_1 & & & & \\ & & & & & p_2 \\ & & & & & p_3 \\ & & & & & p_4 \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 6 & 4 & 1 \end{pmatrix} & p_1 & & & & \\ & & & & & p_2 \\ & & & & & p_3 \\ & & & & & p_4 \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ (1 & 1 & 1 & 4 & 1) \end{array}$$

Zustand nach Ausführung des Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} \times & \times & \times & \times & \times \\ 1 & 0 & 1 & 0 & 4 \end{pmatrix} & p_1 & & & & \\ & & & & & p_2 \\ & & & & & p_3 \\ & & & & & p_4 \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} \times & \times & \times & \times & \times \\ 1 & 1 & 6 & 4 & 1 \end{pmatrix} & p_1 & & & & \\ & & & & & p_2 \\ & & & & & p_3 \\ & & & & & p_4 \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ (4 & 7 & 6 & 6 & 5) \end{array}$$

Nun kann p_2 ausgeführt werden:

Übungsblatt

Aufgabe 2 b)

Iteration 4:

Zustand während der maximalen Ressourcenanforderungen des ausgeführten Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} & \begin{pmatrix} 2 \\ 1 \\ 7 \\ 4 \end{pmatrix} & \begin{pmatrix} 5 \\ 4 \\ 2 \\ 1 \end{pmatrix} & \begin{pmatrix} 3 \\ 2 \\ 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \end{pmatrix} & \begin{pmatrix} 5 \\ 4 \\ 3 \\ 2 \end{pmatrix} \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 3 \\ 6 \\ 0 \\ 2 \\ 4 \end{pmatrix} & \begin{pmatrix} 3 \\ 6 \\ 0 \\ 2 \\ 4 \end{pmatrix} & \begin{pmatrix} 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix} & \begin{pmatrix} 0 \\ 4 \\ 3 \\ 2 \\ 1 \end{pmatrix} & \begin{pmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{pmatrix} \end{array}$$

Zustand nach Ausführung des Prozesses:

$$E = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} \end{array}$$

$$A = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} & \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} \end{array}$$

$$F = \begin{array}{ccccc} & K_1 & K_2 & K_3 & K_4 & K_5 \\ \begin{pmatrix} 5 \\ 7 \\ 7 \\ 6 \\ 9 \end{pmatrix} & \begin{pmatrix} 5 \\ 7 \\ 7 \\ 6 \\ 9 \end{pmatrix} & \begin{pmatrix} 7 \\ 6 \\ 5 \\ 4 \\ 3 \end{pmatrix} & \begin{pmatrix} 7 \\ 6 \\ 5 \\ 4 \\ 3 \end{pmatrix} & \begin{pmatrix} 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix} & \begin{pmatrix} 9 \\ 8 \\ 7 \\ 6 \\ 5 \end{pmatrix} \end{array}$$

Der Zustand Z_3 ist also sicher, da alle Prozesse unter der Annahme des Bankier-Algorithmus terminieren können.

Übungsblatt

Aufgabe 3 a)

- Der **Bankier-Algorithmus** betrachtet nur die **Maximalanforderungen** und nicht den tatsächlichen **Programmablauf**
→ Bankier-Algorithmus verhält sich oft zu **restrektiv**

Beispiel 2

Prozess 1	Prozess 2
1: Anforderung A	1: Anforderung B
2: Freigabe A	2: Freigabe B
3: Anforderung B	3: Anforderung A
4: Freigabe B	4: Freigabe A

- Die **Zeitintervalle**, in denen verschiedene **Ressourcen** für einen Prozess **reserviert** sind, müssen sich **nicht** überlappen

Übungsblatt

Aufgabe 3 a)

Es gilt $V = (1, 1)$ und

$$M = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Wir überprüfen den Zustand nachdem beide Prozesse jeweils die erste Code-Zeile ausgeführt haben. Dann gilt

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

und $F = (0, 0)$. Es handelt sich um einen unsicheren Zustand, da es keinen Prozess mehr gibt, dessen Anforderungen alle mit den verfügbaren Ressourcen erfüllt werden können. Es tritt jedoch kein Deadlock auf, wie auch an einem Ressourcendiagramm abzulesen ist.

Übungsblatt

Aufgabe 3 b)

- Ein Prozess muss **nicht zwingend** die von ihm angegebene "**maximale Anzahl** angeforderter Ressourcen" einer **Klasse** auch wirklich anfordern

Beispiel 2

	Prozess P_0	Prozess P_1
1	<code>x := 4;</code>	<code>Anforderung A;</code>
2	<code>Anforderung A;</code>	<code>x := -3;</code>
3	<code>y := x;</code>	<code>Anforderung A;</code>
4	<code>wenn (y < 0) Freigabe A;</code>	<code>Freigabe A;</code>
5	<code>Anforderung A;</code>	<code>Freigabe A;</code>
6	<code>Freigabe A;</code>	
7	<code>wenn (y ≥ 0) Freigabe A;</code>	

- Variable `x` von beiden Prozessen **gemeinsam** genutzt, Ausführung jeder Zeile **atomare** Operation

Übungsblatt

Aufgabe 3 b)

- Wir nehmen $V = 2$ an
- $M_0 = 2$ (kann im Fall von $y \geq 0$ auftreten) $M_1 = 2$
- Die **Voraussetzungen** für den **Bankier-Algorithmus** sind erfüllt:
 - Die **Maximalanforderungen** sind im Voraus bekannt
 - $M_0 \leq V$ und $M_1 \leq V$, also übersteigen die **Maximalanforderungen** für keinen Prozess die zur Verfügung stehenden **Ressourcen**
- **Unsicherer Zustand nach Ereignissen:**
 - P_0 führt zuerst Zeile 1 und 2 aus, danach führt P_1 die Zeilen 1 und 2 aus:
 - **Unsicherer Zustand:** $E_0 = 1, E_1 = 1, A_0 = M_0 - E_0 = 1, A_1 = M_1 - E_1 = 1$ und $F_A = V_A - E_0 - E_1 = 2 - 1 - 1 = 0$.

Übungsblatt

Aufgabe 3 b)

- Es kann für **keinen** Prozess eine **Restausführung** gewährleistet werden, da der **Bankier-Algorithmus** die **Maximalanforderungen** betrachtet ohne den **tatsächlichen Programmablauf** zu berücksichtigen
- Da $x = -3$ ist, wird von P_0 **nie** mehr als eine **Ressource gleichzeitig** von A verwendet. Deshalb gibt es ausgehend von diesem Zustand **keine Möglichkeit** einen **Deadlock** zu erhalten

Fazit

- In den beiden Beispielen, **kann** (nicht muss) es einen **unsicheren Zustand** geben, der aber **nicht zwingend** zu einem **Deadlock** führt

Quellen

Quellen

Wissenquellen

- 

Quellen

Bildquellen

- 

**Vielen Dank für
eure
Aufmerksamkeit!**

