

5. Übungsblatt

Dienstag, 2. Juli 2019 13:49

2 von 5

Aufgabe 5.2: Handsimulation des Bankieralgorithmus (1,2 Punkte) (Theorie¹)

Der Bankieralgorithmus kontrolliert Ressourcenallokationen, damit keine unsicheren Zustände auftreten.¹
Gegeben ist die folgende verzahnte Ausführung der vier Prozesse P_1, P_2, P_3 und P_4 :

Zeit	Aktion	Zeit	Aktion
P_1 { 1	allocate_r(A, 3);	P_1 { 15	allocate_r(D, 1);
2	allocate_r(D, 2);	16	release_r(C, 3);
P_2 { 3	allocate_r(B, 3);	P_4 { 17	allocate_r(D, 2);
4	allocate_r(D, 1);	18	release_r(B, 1);
P_3 { 5	allocate_r(C, 3);	P_3 { 19	release_r(B, 2);
6	allocate_r(B, 2);	20	exit();
P_1 { 7	release_r(A, 2);	P_1 { 21	release_r(D, 3);
8	allocate_r(C, 3);	22	exit();
P_4 { 9	allocate_r(B, 1);	P_2 { 23	release_r(A, 1);
10	allocate_r(D, 1);	24	release_r(D, 1);
P_3 { 11	allocate_r(C, 1);	25	exit();
12	release_r(C, 4);	P_4 { 26	release_r(D, 3);
P_2 { 13	allocate_r(A, 1);	27	exit();
14	release_r(B, 3);		

Die Funktionen `allocate_r` und `release_r` erhalten hierbei jeweils die Kennung für eines der Betriebsmittel A, B, C oder D und die angeforderte bzw. freizugebende Anzahl als Parameter.
Von jedem Betriebsmittel sind zu Beginn 4 Einheiten vorhanden und nicht belegt.

Führen Sie eine Handsimulation für den gegebenen Ablauf durch. Geben Sie dabei zu jedem Zeitpunkt die aktuellen Belegungen, die Restanforderungen und die freien Betriebsmittel in der aus der Vorlesung bekannten Matrixschreibweise an, prüfen Sie mit dem Bankieralgorithmus, ob die Allokation zu einem unsicheren Zustand führt oder nicht. Tritt ein unsicherer Zustand auf oder sind zum Zeitpunkt der Anfrage nicht genug Betriebsmittel vorhanden, wird der dazugehörige Prozess bis zum Ende der Handsimulation blockiert. Blockierte Prozesse geben ihre Betriebsmittel *nicht* frei.

Aufgabe 5.3: Betriebsmittelverwaltung (Tafelübung)

a) Nennen sie Beispiele für Betriebsmittel. Nach welchen Kriterien lassen sie sich klassifizieren?

Untersuchen Sie die folgende Belegungssituation:

Belegung: $B = \begin{pmatrix} 5 & 1 \\ 1 & 1 \\ 2 & 2 \end{pmatrix}$ Gesamtanforderung: $G = \begin{pmatrix} 9 & 3 \\ 2 & 5 \\ 3 & 3 \end{pmatrix}$ Freie Ressourcen: $f = \begin{pmatrix} 1 & 1 \end{pmatrix}$

¹Dieser Ansatz ist bei Banken inzwischen scheinbar in Vergessenheit geraten, kommt dort heute doch bevorzugt eine andere Strategie zum Einsatz: [http://de.wikipedia.org/wiki/Bail-out_\(Wirtschaft\)](http://de.wikipedia.org/wiki/Bail-out_(Wirtschaft))

Zeitpunkt 3:

$B_3 = \begin{pmatrix} 3 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ $R_3 = \begin{pmatrix} 0 & 0 & 3 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$ $F_3 = (1 \ 1 \ 4 \ 2)$

Zustand sicher, da min. eine Ausführungsreihenfolge existiert.

Zeitpunkt 4:

$B_4 = \begin{pmatrix} 3 & 0 & 0 & 2 \\ 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ $R_4 = \begin{pmatrix} 0 & 0 & 3 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$ $F_4 = (1 \ 1 \ 4 \ 1)$

sicher, da min. eine Ausführungsreihenfolge existiert.

Zeitpunkt 0:

$B_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$ $\begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{matrix}$

$G_0 = \begin{pmatrix} 3 & 0 & 3 & 3 \\ 1 & 3 & 0 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$

$R_0 = G_0 - B_0 = \begin{pmatrix} 3 & 0 & 3 & 3 \\ 1 & 3 & 0 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$

$F_0 = (4 \ 4 \ 4 \ 4)$

Zeitpunkt 1:

$B_1 = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

$F_1 = (1 \ 4 \ 4 \ 4)$

Zustand sicher, da mindestens eine Ausführungsreihenfolge existiert.

$R_1 = \begin{pmatrix} 0 & 0 & 3 & 3 \\ 1 & 3 & 0 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$

Zeitpunkt 2:

$B_2 = \begin{pmatrix} 3 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

$R_2 = \begin{pmatrix} 0 & 0 & 3 & 1 \\ 1 & 3 & 0 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$

$F_2 = (1 \ 4 \ 4 \ 2)$

Zustand sicher, da mindestens eine Ausführungsreihenfolge existiert.