



SoSe 20

Systemprogrammierung

Aufgabenblatt 4

Abgabe von

Michelle Döring

Jannik Heinze

Iryna Novytska

Charlotte Fritsch

Aufgabe 5.3: Betriebsmittelverwaltung mit Fremdbelegung

In welcher Reihenfolge werden die gegebenen Anforderungen bei Abarbeitung nach den folgenden Strategien erfüllt? Geben Sie für jedes Verfahren außerdem die durchschnittliche Wartezeit an. Reichen Sie für jedes Verfahren eine Lösung in Form der unten dargestellten Tabelle ein. Falls mehrere Aktionen zum selben Zeitpunkt stattfinden, notieren Sie bitte jede in einer eigenen Spalte. Geben Sie für Best Fit mit dynamischem Fenster außerdem in jeder Spalte die aktuelle Fenstergröße mit an.

a) FCFS

Zeitpunkt	0	1	2	3	3	4	5
Freigaben	-	F_1	F_2	F_3		F_4	F_5
Anforderungen	-		A_1	A_2	A_3	A_4	A_5
freie Einheiten	0	3	5		0	1	0

b) First Fit

Zeitpunkt	0	1	2	3	3	4	5
Freigaben	-	F_1	F_2	F_3		F_4	F_5
Anforderungen	-	A_4	A_1	A_2	A_5		A_3
freie Einheiten	0	2	4		1	3	0

c) Best Fit

Zeitpunkt	0	1	2	3	3	4	5
Freigaben	-	F_1	F_2	F_3		F_4	F_5
Anforderungen	-	A_4	A_1	A_3	A_5		A_2
freie Einheiten	0	2	4		0	2	0

d) Best Fit mit dynamischem Fenster von $L_{max} = 3$

Zeitpunkt	0	1	2	3	3	4	5
Freigaben	-	F_1	F_2	F_3		F_4	F_5
Anforderungen	-		A_1	A_3	A_2	A_4	A_5
freie Einheiten	0	3	5		0	1	0
Fenstergröße	3	3	2	1	3	2	1

Aufgabe 5.4: Handsimulation des Bankialgorithmus

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 Bankialgorithmus, 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.

#	frei	Restanforderungen	Belegungen	sichere Reihenfolge
0	(4 4 4 4)	$\begin{pmatrix} 3 & 0 & 3 & 3 \\ 1 & 3 & 0 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	trivial (jede möglich)

Wir prüfen nun ob die Anforderung in Schritt 1 gewährt werden kann. Dazu nehmen wir an, sie würde gewährt, berechnen den Vektor der verbleibenden freien Ressourcen und die Matrizen der Restanforderungen und aktuellen Belegungen. Dabei fällt auf, dass die Prozesse P_2, P_3, P_4 jederzeit terminieren können, wodurch sich auch die Menge der freien Ressourcen nicht ändern würde (Nehmen was sie brauchen und geben direkt wieder ab). Damit ist jede beliebige Ausführungsreihenfolge gültig und der **Zustand ist sicher**. Anforderung 1 kann also gewährt werden und wird (erst) jetzt tatsächlich ausgeführt:

1	(1 4 4 4)	$\begin{pmatrix} 0 & 0 & 3 & 3 \\ 1 & 3 & 0 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	trivial (jede möglich)
---	-----------	--	--	------------------------

Genauso prüfen wir nun das (vorerst nur angenommene) Ergebnis nach Schritt 2. Nun muss P_1 auf jeden Fall vor P_4 terminieren. Es lässt sich aber auch dafür eine sichere Reihenfolge finden. Damit lässt auch die Anforderung 2 das System in einem sicheren Zustand.

2	(1 4 4 2)	$\begin{pmatrix} 0 & 0 & 3 & 1 \\ 1 & 3 & 0 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	z.B. P_1, P_2, P_3, P_4 (P_4 auf jeden Fall nach P_1)
---	-----------	--	--	---

Mit der 3. Anforderung schrumpfen die freien Betriebsmittel so weit, dass P_1 und P_2 vor P_3 und P_4 terminieren *müssen*. Es gibt aber nach wie vor (mindestens) eine sichere Reihenfolge.

3	(1 1 4 2)	$\begin{pmatrix} 0 & 0 & 3 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	P_1, P_2, P_3, P_4 oder P_2, P_1, P_3, P_4
---	-----------	--	--	--

Auch Anforderung 4 versetzt das System nicht in einen unsicheren Zustand.

4	(1 1 4 1)	$\begin{pmatrix} 0 & 0 & 3 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 & 0 & 2 \\ 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	P_1, P_2, P_3, P_4 oder P_2, P_1, P_3, P_4
---	-----------	--	--	--

Eine testweise Anwendung der Anforderung 5 führt zu dem nachfolgenden Zustand. P_2 kann noch terminieren, aber keiner der anderen Prozesse, da nicht genug Ressourcen vorhanden wären. Der resultierende Zustand wäre also **unsicher**. Damit wird der Prozess P_3 blockiert und es bleibt bei dem Ressourcenzustand nach Schritt 4.

(5)	(1 1 1 1)	$\begin{pmatrix} 0 & 0 & 3 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 & 0 & 2 \\ 0 & 3 & 0 & 1 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	KEINE! UNSICHERER ZUSTAND!
-----	-----------	--	--	----------------------------

Da der Prozess P_3 blockiert ist wird die Anforderung übersprungen und der Zustand ändert sich nicht.

(6)	(1 1 4 1)	$\begin{pmatrix} 0 & 0 & 3 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$	$\begin{pmatrix} 3 & 0 & 0 & 2 \\ 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	PROZESS P_3 BLOCKIERT!
-----	-----------	--	--	--------------------------

Ein Freigeben von Ressourcen kann ein System nicht von einem sicheren in einen unsicheren Zustand versetzen. Demnach kann die Anforderung 7 durchgeführt werden:

7	(4 1 4 1)	$\begin{pmatrix} 0 & 0 & 3 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 4 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 2 \\ 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	P_1, P_2, P_4
---	-----------	--	--	-----------------

$$8 \quad (4 \ 1 \ 1 \ 1) \quad \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ \textcolor{red}{0} & \textcolor{red}{2} & \textcolor{red}{4} & \textcolor{red}{0} \\ 0 & 1 & 0 & 3 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 3 & 2 \\ 0 & 3 & 0 & 1 \\ \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad P_1, P_2, P_4$$

$$9 \quad (4 \ 0 \ 1 \ 1) \quad \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ \textcolor{red}{0} & \textcolor{red}{2} & \textcolor{red}{4} & \textcolor{red}{0} \\ 0 & 0 & 0 & 3 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 3 & 2 \\ 0 & 3 & 0 & 1 \\ \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad P_1, P_2, P_4$$

Durch das Allokieren der letzten freien Ressource des Betriebsmittels D , muss P_2 vor P_1 terminieren. Dies ist aber immernoch möglich und der Zustand damit sicher.

$$10 \quad (4 \ 0 \ 1 \ 0) \quad \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ \textcolor{red}{0} & \textcolor{red}{2} & \textcolor{red}{4} & \textcolor{red}{0} \\ 0 & 0 & 0 & 2 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 3 & 2 \\ 0 & 3 & 0 & 1 \\ \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad P_2, P_1, P_4$$

Da der Prozess P_3 blockiert ist werden die Anforderungen 11, 12 übersprungen und der Zustand ändert sich nicht.

Die Anforderung 13 kann P_2 gewährt werden - P_2 hat damit seine Gesamtanforderung erreicht.

$$13 \quad (3 \ 0 \ 1 \ 0) \quad \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ \textcolor{red}{0} & \textcolor{red}{2} & \textcolor{red}{4} & \textcolor{red}{0} \\ 0 & 0 & 0 & 2 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 3 & 2 \\ 1 & 3 & 0 & 1 \\ \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Nun terminiert der Prozess P_2 und seine Betriebsmittel werden frei gegeben. (Und Anforderung 14 ist damit bereits erfüllt.)

$$14 \quad (4 \ 3 \ 1 \ 1) \quad \begin{pmatrix} 0 & 0 & 0 & 1 \\ \textcolor{red}{0} & \textcolor{red}{2} & \textcolor{red}{4} & \textcolor{red}{0} \\ 0 & 0 & 0 & 2 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 3 & 2 \\ 0 & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad P_1, P_4$$

Die Anforderung 15 kann P_1 gewährt werden - P_1 hat damit seine Gesamtanforderung erreicht, terminiert und seine Betriebsmittel werden frei gegeben. (Und Anfrage 16 ist damit bereits erfüllt.)

$$15 \quad (4 \ 3 \ 1 \ 0) \quad \begin{pmatrix} 0 & 0 & 0 & 0 \\ \textcolor{red}{0} & \textcolor{red}{2} & \textcolor{red}{4} & \textcolor{red}{0} \\ 0 & 0 & 0 & 2 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 3 & 3 \\ \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad P_4$$

$$16 \quad (4 \ 3 \ 4 \ 3) \quad \begin{pmatrix} \textcolor{red}{0} & \textcolor{red}{2} & \textcolor{red}{4} & \textcolor{red}{0} \\ 0 & 0 & 0 & 2 \end{pmatrix} \quad \begin{pmatrix} \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad P_4$$

Die Anforderung 17 kann nun P_4 gewährt werden - P_4 hat damit auch seine Gesamtanforderung erreicht, terminiert und seine Betriebsmittel werden frei gegeben.

$$17 \quad (4 \ 3 \ 4 \ 1) \quad \begin{pmatrix} \textcolor{red}{0} & \textcolor{red}{2} & \textcolor{red}{4} & \textcolor{red}{0} \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} \\ 0 & 1 & 0 & 3 \end{pmatrix}$$

$$17 \quad (4 \ 4 \ 4 \ 4) \quad \begin{pmatrix} \textcolor{red}{0} & \textcolor{red}{2} & \textcolor{red}{4} & \textcolor{red}{0} \end{pmatrix} \quad \begin{pmatrix} \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} & \textcolor{red}{0} \end{pmatrix}$$

Alle beteiligten Prozesse sind terminiert oder blockiert, es stehen keine weiteren Anforderungen aus.