Übung 6

Abgabe: 21.06.2018 8:30

Aufgabe 6.1: Semaphoren in C (2 Punkte)

Im L2P finden Sie das Programm erzeuger_verbraucher.c. Es handelt sich um eine Implementierung des Erzeuger-Verbraucher-Problems mit zwei Verbrauchern. Die Synchronisation der Verbraucherprozesse erfolgt über UNIX-Semaphoren. Erweitern Sie das Programm um einen weiteren Erzeugerprozess. Stellen Sie bei Ihrer Lösung durch eine zweite Semaphore sicher, dass alles richtig funktioniert.

Aufgabe 6.2: Deadlocks (2 + 4 = 6 Punkte)

Betrachten Sie in dieser Aufgabe das Problem der Erkennung von Deadlocks.

a) Gegeben sei ein System mit drei Prozessen $P_1, \ldots, P_3$ und vier Betriebsmitteln $R_A, \ldots, R_D$, von denen jeweils nur ein Exemplar zur Verfügung steht und die nur von einem Prozess gleichzeitig verwendet werden können. Die Anforderungen von Betriebsmitteln durch die Prozesse ist in der folgenden Tabelle gegeben.

Zeit	P_1	P_2	P_3
1	(A;5)		
2		(B;5)	
3			(D;3)
4			(C;5)
5	(D;1)		(B;3)
6		(A;2)	
7			

Ein Eintrag der Form $(X;a)$ in der Spalte P_i bedeutet, dass das Betriebsmittel R_X für a Zeiteinheiten vom Prozess P_i angefordert wird. Ein Prozess kann seine Arbeit erst dann fortsetzen, wenn ihm das jeweils angeforderte Betriebsmittel zugewiesen wurde.

Sind einem Prozess die angeforderten Betriebsmittel zum Zeitpunkt t zugewiesen worden, werden sie in der Zeit von $t+1$ bis $t+a$ verwendet und stehen zum Zeitpunkt $t+a+1$ wieder zur Verfügung.

Stellen Sie die Situation aus der obigen Tabelle für den Zeitpunkt $t = 7$ anhand eines Resource Allocation Graph dar. Liegt ein Deadlock vor? Begründen Sie Ihre Antwort.

b) Zwei Prozesse A und B benötigen zu ihrer Ausführung je eine gewisse Zeitdauer t_A bzw. t_B und greifen auf die drei Betriebsmittel X, Y und Z zu. t_{IJ} bezeichne die Dauer, die das Betriebsmittel I vom Prozess J irgendwann während der Ausführungszeit benötigt wird. Ein Experte macht nun folgende Aussagen:

i) Ein Deadlock kann nicht auftreten, wenn $t_{XA} \cdot t_{XB} + t_{YA} \cdot t_{YB} + t_{ZA} \cdot t_{ZB} < t_A \cdot t_B$.
ii) Ein Deadlock liegt automatisch vor, wenn $t_{XA} + t_{YA} + t_{ZA} > t_A$ und $t_{XB} + t_{YB} + t_{ZB} > t_B$.
iii) Ein Deadlock liegt automatisch vor, wenn $t_{XA} \cdot t_{XB} + t_{YA} \cdot t_{YB} + t_{ZA} \cdot t_{ZB} > t_A \cdot t_B$.
iv) Ein Deadlock kann nicht auftreten, wenn $t_{XA} + t_{YA} + t_{ZA} < t_A$ und $t_{XB} + t_{YB} + t_{ZB} < t_B$.

Welche dieser Aussagen sind richtig und welche Aussagen stimmen nicht? Begründen Sie zu jeder der einzelnen Aussagen, warum diese gültig sind, oder geben Sie zu jeder Aussage, die nicht gültig ist, ein Gegenbeispiel an (z.B. durch ein Prozessfortschrittsdiagramm).

Aufgabe 6.3: Resource Allocation Graph (3 Punkte)

Betrachten Sie in dieser Aufgabe das Problem der Erkennung von Deadlocks. Gegeben sei ein System mit drei Prozessen $P_1, \ldots, P_3$ und vier Betriebsmitteln $A, \ldots, D$, von denen jeweils nur ein Exemplar zur Verfügung steht und die nur von einem Prozess gleichzeitig verwendet werden können. Die Anforderungen von Betriebsmitteln durch die Prozesse ist in der folgenden Tabelle gegeben.

Zeit	P_1	P_2	P_3
1	(D;2)	(B;4)	
2			(A;5)
3	(C;3)	(D;3)	
4		(C;2)	(B;3)
5	(A;5)		
6			

Ein Eintrag der Form $(X; a)$ in der Spalte P_i bedeutet, dass das Betriebsmittel X für a Zeiteinheiten von Prozess P_i angefordert wird. Ein Prozess kann seine Ausführung erst dann fortsetzen, wenn ihm das jeweils angeforderte Betriebsmittel zugewiesen wurde.

Fordert ein Prozess ein Betriebsmittel zum Zeitpunkt t an und es kann zugewiesen werden, wird es in der Zeit von $t+1$ bis $t+a$ verwendet und steht zum Zeitpunkt $t+a+1$ wieder zur Verfügung. Während ein Prozess auf die Zuordnung eines angeforderten Betriebsmittels wartet, ist er blockiert und die Freigabe der bereits zugeordneten Betriebsmittel verzögert sich entsprechend. Wird ein Betriebsmittel zum Zeitpunkt t von einem Prozess freigegeben und es wartet ein anderer Prozess auf die Zuteilung dieses Betriebsmittels, kann er es ab $t+1$ nutzen.

Es werde angenommen, dass die Prozesse simultan ausgeführt werden (z.B. auf mehreren CPU-Kernen), so dass die jeweils in einer Zeile stehenden Anweisungen zeitgleich ausgeführt werden, solange kein Prozess blockiert ist.

Stellen Sie die Situation aus der obigen Tabelle für den Zeitpunkt $t = 6$ anhand eines Resource Allocation Graph dar. Liegt ein Deadlock vor? Begründen Sie Ihre Antwort.

Aufgabe 6.4: Banker-Algorithmus (2 + 1 = 3 Punkte)

Gegeben seien drei Prozesse P_1, P_2 und P_3, die drei Betriebsmittel BM_1, BM_2 und BM_3 benutzen wollen. Es existieren 13 Exemplare von BM_1, 8 von BM_2 und 9 von BM_3; jedes Exemplar kann nur exklusiv genutzt werden. Zum Zeitpunkt 0 sei der Vektor freier Betriebsmittel (BM_1, BM_2, BM_3) gegeben mit: $V(0) = (V_1(0), V_2(0), V_3(0)) = (9, 5, 6)$.

Weiterhin gilt:

- $Q_1^{max}(0) = (Q_{11}^{max}, Q_{12}^{max}, Q_{13}^{max}) = (11, 3, 3)$,
  $H_1(0) = (H_{11}(0), H_{12}(0), H_{13}(0)) = (0, 2, 1)$
- $Q_2^{max}(0) = (4, 6, 5)$, $H_2(0) = (1, 0, 1)$
- $Q_3^{max}(0) = (5, 1, 1)$, $H_3(0) = (3, 1, 1)$

a) Prüfen Sie mit dem in der Vorlesung vorgestellten Banker-Algorithmus zur Deadlockvermeidung, ob sich das System in einem sicheren Zustand befindet.

b) Ein System verwende den Banker-Algorithmus, um Deadlocks zu vermeiden. Zu einem betrachteten Zeitpunkt fordere ein Prozess P ein Betriebsmittel BM an; die Anforderung wird allerdings vom System abgelehnt. Kann man daraus sicher schließen, dass die Zuteilung von BM an P einen Deadlock verursachen würde? Warum bzw. warum nicht?

Aufgabe 6.5: Prozessfortschrittsdiagramme (1,5 + 3,5 + 1 = 6 Punkte)

Wollen zwei Prozesse während ihrer Laufzeit verschiedene Betriebsmittel exklusiv benutzen, könnten sie dies dem System im Voraus mitteilen, um Deadlocks zu vermeiden. Das Betriebssystem könnte dann ihren zeitlichen Ablauf so koordinieren, dass ein erfolgreiches Beenden beider Prozesse gesichert ist (falls dies möglich ist).

Zur Veranschaulichung des Ablaufs kann man Prozessfortschrittsdiagramme verwenden. Die Achsen markieren dabei die Laufzeiten der beiden Prozesse, die um Betriebsmittel konkurrieren. Schraffiert oder farblich hervorgehoben werden die Zeiträume, in denen beide Prozesse gleichzeitig ein exklusives Betriebsmittel beanspruchen. Ein Schedule gibt an, welcher Prozess in welcher Zeiteinheit ausgeführt wird (z.B. für die Zeiteinheiten 1 bis 10 und zwei Prozesse A und B: $AABABBBAAA$). Diese Art der Darstellung beschränkt sich auf eine einzelne CPU (bzw. einen Kern); würde man mehrere CPUs/Kerne verwenden, würde die Darstellung komplexer.

a) Betrachtet werde ein System mit einer CPU (und einem Kern). Gegeben seien die Prozesse A und B, die jeweils zehn Zeiteinheiten laufen, und fünf exklusiv zu nutzende Betriebsmittel BM_1 bis BM_5. Die Prozesse versuchen in den folgenden Zeiteinheiten ihrer Ausführung auf die Betriebsmittel zuzugreifen:

	BM_1	BM_2	BM_3	BM_4	BM_5
Prozess A	6-7	5-9	6-7	3-5	7-8
Prozess B	1-3	3-5	7-9	6-8	5-7

Tragen Sie diese Situation in ein Prozessfortschrittsdiagramm ein (Prozess A: y-Achse, Prozess B: x-Achse).

b) Zu jedem Zeitpunkt befindet sich das System in einem Zustand (x, y), d.h. Prozess B wurde bereits für x Zeiteinheiten ausgeführt, Prozess A für y Zeiteinheiten. Ein Zustand, in dem eine erfolgreiche Beendigung der beiden Prozesse möglich ist, wird sicher genannt. Ein Zustand, in dem zwar noch nicht unbedingt ein Deadlock vorliegt, von dem aus allerdings ein Deadlock unvermeidlich ist, wird unsicher genannt. Ein Zustand im Diagramm, der durch keine Ausführungsreihenfolge der Prozesse jemals erreicht werden kann, wird unmöglich genannt. Heben Sie in Ihrem Diagramm aus Teil a) hervor, welche Bereiche unsicher und welche unmöglich sind. Geben Sie darüber hinaus an, welchen Status (sicher, unsicher, unmöglich) folgende Zustände haben:

$$(7,3); (5,10); (9,8); (5,2)$$

c) Schreiben Sie einen Schedule für die beiden Prozesse auf, so dass beide Prozesse vollständig ausgeführt werden können, und zeichnen Sie ihn in das Diagramm aus Teil a) ein.

Hinweis: Ein sequentieller Schedule (d.h. zunächst die vollständige Ausführung eines Prozesses, dann die des anderen) ist an dieser Stelle verboten.