

13. Übung

Ausgabe Abgabe
31.01.06 keine

Bei Fragen und Problemen können Sie uns per E-mail unter den folgenden Adressen erreichen:
Christian Gollan, Klaus Macherey, Richard Zens { gollan, k.macherey, zens }@cs.rwth-aachen.de

Aufgabe 1 (10 Punkte): Prozessscheduling

Wir betrachten im folgenden fünf Prozesse, die alle auf einer CPU bearbeitet werden müssen. Die Länge der Prozesse ist in Klammern angegeben: $P_1(5), P_2(7), P_3(3), P_4(8), P_5(2)$. Alle Prozesse sind bereits zum Zeitpunkt $t = 0$ im System vorhanden und sind nach aufsteigendem Index angekommen (also P_1 zuerst).

- a) Geben Sie die entstehenden Warteschlangen für die Strategien FCFS und LCFS an und bestimmen Sie jeweils die mittlere Wartezeit!
- b) Betrachten Sie nun das Round-Robin Verfahren (Zeitquantum 5). Geben Sie zu jedem Prozess an, zu welchen Zeitpunkten er die CPU belegt! Die Umschaltzeiten sollen hierbei vernachlässigt werden. Bestimmen Sie auch hier die mittlere Wartezeit!

Aufgabe 2 (10 Punkte): Prozesssynchronisation

Gegeben seien zwei Prozesse P_0 und P_1 . Es existiert eine kritische Sektion, die nicht von beiden Prozessen gleichzeitig betreten werden kann. Folgender Synchronisationsalgorithmus wird vorgeschlagen:
Den Prozessen P_0 und P_1 werden folgende gemeinsame Variablen zur Verfügung gestellt:

```
bool flag[1];    /* mit FALSE initialisiert                                            */
```

Der Prozess P_i arbeitet nach folgender Programmsequenz, wobei P_j den jeweils konkurrierenden Prozess bezeichnet:

```
01 repeat  
02     flag[j]=TRUE;  
03     while (flag[j]) do noop;  
04     critical_section(Pi);  
05     flag[i]:=FALSE;  
06     remainder_section(Pi);  
07 until FALSE;
```

Arbeitet das Verfahren korrekt? Prüfen Sie dies anhand der Anforderungen an eine korrekte Lösung des wechselseitigen Ausschlussproblems. Skizzieren Sie für *jede* der Bedingungen den Beweis oder geben Sie ein Gegenbeispiel an!

Aufgabe 3 (4+6=10 Punkte): Deadlocks

Gegeben seien vier Prozesse P_0, P_1, P_2 und P_3 , die drei exklusive Betriebsmittel BM_1, BM_2 und BM_3 benutzen wollen. Es existieren 10 Exemplare von $BM_1, 8$ von BM_2 und 4 von BM_3 : Available = (10, 8, 4). Ferner sei der maximale Bedarf der Prozesse bekannt:

- $\text{Max}(0) = (7, 5, 3)$
- $\text{Max}(1) = (7, 2, 2)$
- $\text{Max}(2) = (9, 8, 1)$
- $\text{Max}(3) = (3, 2, 0)$

a) Prüfen Sie für die Zustände A und B, ob sich das System in einem sicheren Zustand befindet! Begründen Sie Ihre Antwort!

$\text{Allocation_A}(0) = (0, 2, 2),$	$\text{Allocation_B}(0) = (1, 2, 1),$	$\text{Allocation_C}(0) = (1, 2, 0)$
$\text{Allocation_A}(1) = (3, 1, 1),$	$\text{Allocation_B}(1) = (3, 1, 2),$	$\text{Allocation_C}(1) = (1, 2, 1)$
$\text{Allocation_A}(2) = (3, 2, 1),$	$\text{Allocation_B}(2) = (2, 2, 0),$	$\text{Allocation_C}(2) = (2, 2, 1)$
$\text{Allocation_A}(3) = (1, 1, 0),$	$\text{Allocation_B}(3) = (1, 1, 0),$	$\text{Allocation_C}(3) = (1, 1, 0)$

b) Gehen Sie von dem sicheren Zustand C aus. Welche der drei einzeln gegebenen Anforderungen führen in einen sicheren Zustand, d.h. welche Zuteilungen dürfen erlaubt werden und welche nicht? Begründen Sie Ihre Antwort!

1. $\text{Request}(3) = (0, 2, 0)$
2. $\text{Request}(2) = (1, 1, 0)$
3. $\text{Request}(0) = (0, 1, 1)$

Aufgabe 4 (2+3+4=9 Punkte): Bediensysteme

Ein von Ihnen neu erworbener Rechner besitze 4 Prozessoren. Da Sie die Lastverteilung des Rechners quantitativ analysieren wollen, erinnern Sie sich an die Theorie der Bediensysteme. Der Rechner verarbeitet die zufällig exponentialverteilt ankommenden Jobs nach folgender Strategie: ist ein Prozessor frei, so wird der Job dem Prozessor zugewiesen und verarbeitet. Sind alle Prozessoren belegt, so werden die Jobs in einer Warteschlange eingereiht.

- a) Beschreiben Sie das im Text angegebene System in Kendall Notation.
- b) Zeichnen Sie zu dem im Text beschriebenen System das zugehörige Zustandsdiagramm auf.
- c) Das System befinde sich nun in einem stabilen Zustand. Stellen Sie hierzu die Bilanzgleichungen auf.

Aufgabe 5 (4+4=8 Punkte): Wechselseitiger Ausschluß

Der Eiffelturm in Paris besitze 3 Aussichtsplattformen: die nebeneinander liegenden Plattformen A und B sowie die etwas tiefer gelegene Plattform C. Die beiden Plattformen A und B können jeweils 20 Personen aufnehmen. Die tiefer gelegene Plattform C hat eine Kapazität von 40 Personen. Da die höher gelegenen Plattformen touristisch attraktiver sind, werden zunächst alle Touristen auf die Plattformen A und B transportiert. Sind diese voll, werden die restlichen Kunden zur Plattform C gebracht. Die Touristen werden mit mehreren Aufzügen nach oben bzw. wieder nach unten gebracht. Jeder Aufzug kann nur eine Person befördern.

- a) Schreiben Sie einen Algorithmus `hoch()` in Pseudocode, der die Touristen auf die Plattformen befördert. Berücksichtigen Sie dabei die oben genannten Kapazitäten und die oben beschriebene Strategie. Verwenden Sie Semaphoren mit assoziierter Warteschlange.

- b) Schreiben Sie drei Algorithmen `runterA()`, `runterB()`, `runterC()`, welche die Touristen von den Plattformen wieder zurückbringen.

Aufgabe 6 (2+2+3 Punkte): Speicherverwaltung

Für die Speicherverwaltung nach dem Segmentierungsverfahren sei für ein Programm die folgende Segmenttabelle gegeben (Länge in Speicherworten):

Segment	Basis	Länge
0	1320	320
1	562	58
2	310	120
3	750	60
4	830	150
5	990	110

- a) Wieviele Speicherworte stehen dem Programm im physikalischen Speicher zur Verfügung?
- b) Welches ist die kleinste und welches die größte für das Programm verfügbare physikalische Adresse?
- c) Berechnen Sie zu den folgenden physikalischen Adressen jeweils die logischen Adressen. Welche Anfrage löst einen Segmentation Fault aus?
1. 780
 2. 1436
 3. 580
 4. 984
 5. 420

Aufgabe 7 (6 Punkte): Speicherverwaltung

Der Referenz String ω beschreibe die Folge der Seitenzugriffe. Die Tabellen zeigen die Belegung der Seitenrahmen im Hauptspeicher. Es stehen einmal drei und einmal vier Rahmen zur Verfügung. Vervollständigen Sie die Tabellen unter Verwendung der CLIMB Strategie. Markieren Sie die Seitenfehler mit einem \star .

$\omega =$	2	4	1	3	5	5	2	3	3	3	1	4	0	3	3	2	1	4	2	2	4	
Seitenfehler																						
Seitenrahmen 1	2	2	2	2																		
Seitenrahmen 2		4	4	4																		
Seitenrahmen 3				1	3																	
Seitenfehler																						
Seitenrahmen 1	2	2	2	2																		
Seitenrahmen 2		4	4	4																		
Seitenrahmen 3				1	1																	
Seitenrahmen 4						3																

Aufgabe 8 (4+2 Punkte): Lifetime-Funktion

- a) Geben Sie die Lifetime-Funktion $L(m)$ zu einem Programm an, das durch den Referenzstring ω mit

$$\omega = 3\ 5\ 1\ 3\ 2\ 4\ 0\ 1\ 4\ 1\ 3\ 2\ 1\ 3\ 0\ 3\ 5\ 1\ 3\ 2\ 4\ 0\ 1\ 4\ 1\ 3\ 2\ 1\ 3\ 0$$

gekennzeichnet ist. Die Zeit, die zwischen zwei Seitenanfragen vergeht, soll jeweils eine Zeiteinheit betragen. Die Seitenersetzung erfolge mittels FIFO-Strategie. Gehen Sie davon aus, daß das Working Set zu Beginn mit anderen als den angegebenen Seiten gefüllt ist. Zeichnen Sie die Lifetime-Funktion $L(m)$ für $m = 1 \dots 10$. Geben Sie zusätzlich die Zahl der Seitenfehler für $m = 1 \dots 10$ in einer Tabelle an.

- b) Welche optimale Einstellung der Speichergröße ergibt sich bei Anwendung des primären Knie-Kriteriums?

Aufgabe 9 (4+2=6 Punkte): Working Set

Es sei der folgende Referenzstring eines Prozesses gegeben:

$$\omega = 3\ 4\ 1\ 2\ 2\ 3\ 1\ 2\ 5\ 6\ 1\ 2\ 3\ 4\ 3\ 2\ 1\ 5\ 2\ 1\ 3\ 4\ 4\ 2$$

Der Wert h soll für diese Aufgabe $h := 5$ betragen.

- a) Konstruieren Sie einen Graphen, auf dessen x -Achse die Zeit t (von 0 bis 24) und auf dessen y -Achse die Anzahl der momentan dem Prozess zugeteilten Seiten aufgetragen wird. Gehen Sie davon aus, daß das Working Set zu Beginn, also bei $t = 0$, mit keiner Seite gefüllt ist und die Zeit, die zwischen zwei Seitenanfragen vergeht, jeweils eine Zeiteinheit beträgt.
- b) Geben Sie die Working Sets der lokalen Bereiche zu den Zeitpunkten 12, 13, 21 und 24 an.