Technische Universität Braunschweig

Institut für Betriebssysteme und Rechnerverbund

Prof. Dr. H. Langendörfer · Prof. Dr. M. Zitterbart

Klausur zur Vorlesung

Betriebssysteme und Netze

22. Juli 1999

Zugelassene Hilfsmittel:	Vorlesungsunterlagen, Übungsmitschriften
Bearbeitungszeit:	120 Minuten
Hinweis:	Jedes Blatt ist mit Namen, Vornamen und Matrikelnummer zu versehen.

Name:
Vorname:
Fachrichtung:
Matrikel-Nr.:

Wiederholer: ☐

Bewertung

Aufgabe 1	max. 8 Punkte	Punkte	
Aufgabe 2	max. 10 Punkte	Punkte	
Aufgabe 3	max. 11 Punkte	Punkte	
Aufgabe 4	max. 6 Punkte	Punkte	
Aufgabe 5	max. 15 Punkte	Punkte	
Summe	max. 50 Punkte	Punkte	

Note:

Aufgabe 1

Gegeben ist ein Betriebssystem mit einem virtuellen Speicher auf der Basis des Paging-Verfahrens. Betrachten Sie einen Prozeß, der die folgende Sequenz von Speicherzugriffen erzeugt:

$$6, 5, 4, 2, 1, 5, 3, 1, 5, 3$$

Dem Prozeß stehen 4 Kacheln (physikalisch vorhandene Speicherrahmen) zur Verfügung, die anfangs nicht belegt sind.

a) Bestimmen Sie die sich verändernden Einträge der Seiten–Kachel–Tabelle unter Verwendung der FIFO (first-in-first-out) Seitenersetzungsstrategie. Geben Sie die Anzahl der Seitenfehler an.

b) Der Working–Set eines Prozesses mit dem Referenzstring $w = r[1], r[2], \ldots, r[t]$ sei zum Zeitpunkt t wie folgt definiert:

$$W(t, T) := \{i \in I\!N | i \in \{r[t - T], r[t - T + 1], \ldots, r[t - 1]\}\}.$$

Eine auf dem Working–Set beruhende Seitenersetzungsstrategie lagert unter den Seiten, die nicht zum Working-Set gehören, diejenige Seite mit der kleinsten Nummer aus. Bestimmen Sie die Anzahl der auftretenden Seitenfehler in Abhängigkeit vom Working-Set-Parameter $T, T \in \{1, 2, 3\}$.

Geben Sie für beide Teilaufgaben die Belegung der Kacheln in einer Tabelle an. Legen Sie für jede Kachel (0–3) eine Zeile an und für jeden Speicherzugriff eine Spalte. Tragen Sie für jeden Speicherzugriff in der Tabelle ein, welche Seite welcher Kachel zugeordnet ist. Beachten Sie, daß eine Seite die Kachel nicht wechselt, während die Seite eingelagert ist. Markieren Sie jede Spalte, in der ein Seitenfehler auftritt.

(2 + 6 Punkte)

Aufgabe 2

Fünf Prozesse, P_1 bis P_5, kommen nahezu gleichzeitig in einem System an, in dem nur ein Prozessor vorhanden ist. Die Ankunftsreihenfolge ist durch (P_1, P_2, $\cdots$, P_5) gegeben. Die Zeitdifferenzen zwischen den Ankünften können jedoch vernachlässigt werden. Die Ausführungszeiten und Prioritäten (5 ist höchste Priorität) sind durch die folgende Tabelle gegeben.

Prozeß	Ausführungszeit	Priorität
P_1	5 ZE	4
P_2	4 ZE	1
P_3	1 ZE	2
P_4	2 ZE	5
P_5	3 ZE	3

a) Stellen Sie für jede der folgenden Prozessorvergabestrategien den Ablauf durch ein Gantt-Diagramm dar.

1. SJF (Shortest Job First): Der Prozessor wird den Prozessen in der Reihenfolge ihrer Ausführungszeit (kurze vor lange Ausführungszeiten) zugeteilt.
2. PRS (Prioritätsgesteuerte Prozessorvergabe): Der Prozessor wird den Prozessen in der Reihenfolge ihrer Priorität (hohe vor kleiner) zugeteilt.
3. FCFS (First Come First Serve): Der Prozessor wird den Prozessen in der Reihenfolge ihrer Ankunft zugeteilt.
4. RR (Round Robin): Alle Prozesse stehen in einer Warteschlange. Neu ankommende Prozesse werden hinten eingereiht. Das gleiche gilt für einen Prozeß, dessen Zeitscheibe (Länge 2 ZE) zwar abgelaufen ist, der aber noch nicht fertig ist.

In den Fällen 1.–3. laufen Aufträge bis zur Beendigung ohne Unterbrechung. In allen Fällen kann die Zeit für den Prozeßwechsel vernachlässigt werden. Alle Prozesse verbrauchen ausschließlich CPU-Zeit und werden nicht durch Ein/Ausgabe-Operationen blockiert.

b) Bestimmen Sie für jede der 4 Prozessorvergabestrategien aus dem Aufgabenteil a) folgende Kenngrößen:

i) mittlere Verweilzeit $\overline{e} = \dfrac{1}{n}\sum_{i=1}^{n} e_i$, ($e_i$ ist der Beendigungszeitpunkt von P_i)

ii) Gesamtdurchlaufzeit $t(S) = \max_{1 \le i \le n}\{e_i\}$

iii) mittlere Anzahl unbeendeter Prozesse im System $\overline{n} = \dfrac{1}{t(S)}\sum_{t=1}^{t(S)} n(t)$

c) Warum kann die SJF-Strategie (außer in sehr speziellen Anwendungen) in der Praxis kaum eingesetzt werden? Ist die SJF-Strategie fair?

(4 + 4 + 2 Punkte)

Aufgabe 3

In der Vorlesung wurden Semaphore zur Synchronisation paralleler Prozesse behandelt. Die Semaphor-Operationen up und down wurden folgendermaßen definiert:

```
down(s)
{
    s := s -1;
    if (s < 0) queue_this_process_and_block();
}

up(s)
{
    s := s + 1;
    if (s <= 0) wakeup_process_from_queue();
}
```

a) Sie sollen ein Datenobjekt implementieren, das von mehreren konkurrierenden Prozessen gemeinsam benutzt wird. Einige Prozesse wollen das Datenobjekt nur lesen, während andere Prozesse das Datenobjekt auch verändern wollen. Die Prozesse, die das Datenobjekt nur lesen wollen, werden als „reader“ bezeichnet und die übrigen Prozesse als „writer“. Offensichtlich können mehrere „reader“ gleichzeitig das Datenobjekt lesen ohne Inkonsistenzen zu erzeugen. Wenn hingegen ein „writer“ und ein beliebiger anderer Prozess gleichzeitig auf das Datenobjekt zugreifen, dann können Inkonsistenzen entstehen.

   Geben Sie je eine Prozedur für einen „reader“ und einen „writer“ Prozeß an, die einem „writer“ exklusiven Zugriff auf das Datenobjekt garantiert. Ferner sollen mehrere „reader“ Prozesse gleichzeitig lesen können. Außerdem sollen keine „reader“ Prozesse warten müssen, solange kein „writer“ Prozeß exklusiven Zugriff auf das Datenobjekt bekommen hat.

   Verwenden Sie zur Synchronisation Semaphore wie sie oben definiert sind. Geben Sie die Initialisierungen für alle von Ihnen verwendeten Variablen an und beschreiben Sie die Bedeutung der von Ihnen benutzten Semaphore.

b) Welches prinzipielle Fairneß-Problem besteht in der Beschreibung im Aufgabenteil a)?

(10 + 1 Punkte)

Aufgabe 4

Eine Möglichkeit der Fehlersicherung bei der Datenübertragung ist die zyklische Blocksicherung. Die übertragene Sequenz $u(x) = x^r a(x) + q(x)$ sei dabei so gestaltet, daß sie durch das Generatorpolynom $g_r(x)$ ohne Rest teilbar ist. Für fehlerfrei übertragene Sendungen gilt also $u(x) = g_r(x)p(x)$ und auch $x^r a(x) = g_r(x)p(x) + q(x)$. Mit dieser Bedingung ist $q(x)$ der Divisionsrest, der bei der Division von $x^r a(x)$ durch $g_r(x)$ entsteht.

Sei 1001 0011 1010 0011 1000 0101 die übertragende Bitfolge. Das Generatorpolynom sei $g_4(x) = x^4 + x + 1$.

a) Welche Bitfolge wird vom Sender gesendet?

b) Bei der Übertragung der Bitfolge werden die vier Bits an den Bitpositionen 5–8 alle auf 1 gesetzt. (Die Bitfolge wird von links nach rechts übertragen und das erste übertragene Bit hat die Nummer 1.) Geben Sie die empfangene Bitfolge an.

c) Kann der in b) entstandene Übertragungsfehler vom Empfänger erkannt werden? Führen Sie die entsprechende Berechnung des Empfängers aus und begründen Sie damit die Antwort.

d) Welche Art von Übertragungsfehler läßt sich mit Hilfe eines beliebigen Polynoms niemals erkennen? Begründen Sie Ihre Antwort.

$(2 + 1 + 2 + 1$ Punkte$)$

Aufgabe 5

Betrachten Sie folgende Varianten eines Protokolls zum Auslesen von Variablen eines entfernten Rechners. Gehen Sie im folgenden davon aus, daß keine Fehler auftreten bzw. daß Fehler von einer anderen Protokollschicht behandelt werden.

1. In der ersten Variante wird jede Variable einzeln durch eine Nachricht angefragt (GetRequest) und von dem entfernten Rechner in einer Antwortnachricht (Response) dem anfragenden Prozeß zugestellt. Zum Lesen von n Variablen sind also n GetRequest/Response Interaktionen notwendig. Anfragen und Antworten dürfen sich nicht überlappen.

2. In der zweiten Variante werden immer m Variablen durch eine Nachricht (GetManyRequest) angefragt und von dem entfernten Rechner in einer Antwortnachricht (Response) dem anfragenden Prozeß zugestellt. Zum Lesen von n Variablen sind also $\lceil n/m \rceil$ GetManyRequest/Response Interaktionen notwendig, wobei sich die Anfragen und Antworten nicht überlappen dürfen.

3. In der dritten Variante werden alle Variablen zusammen angefragt (GetAllRequest). Der entfernte Rechner sendet jeweils m angefragte Variablen in einer Antwortnachricht (Response) an den anfragenden Prozeß. Insgesamt werden also eine Anfragenachricht und $\lceil n/m \rceil$ Anwortnachrichten übertragen.

Jede Nachricht besteht aus einem festen Nachrichtenkopf von h Bytes. Für eine einzelne Variable werden jeweils v Bytes in einer Nachricht benötigt.

a) Stellen Sie den Ablauf der drei Protokolle für $n = 8$ Variablen und $m = 4$ in Weg-Zeit-Diagrammen dar.

b) Geben Sie für die drei Varianten jeweils eine allgemeine Formel zur Bestimmung der Anzahl der übertragenen Bytes an. Benutzen Sie die Größen n, m, h und v wie oben beschrieben.

c) Bestimmen Sie für alle drei Varianten die Anzahl der übertragenen Bytes für die konkreten Werte $h = 100$, $v = 15$, $n = 10000$ und $m = 80$.

d) Geben Sie für die drei Varianten jeweils eine allgemeine Formel zur Bestimmung der Gesamtverzögerung an. Berücksichtigen Sie dabei die Ausbreitungsverzögerung T_p und die Übertragungsverzögerung T_x. Die Verarbeitung in den Knoten kann vernachlässigt werden. Benutzen Sie die Größen n, m, h und v wie oben beschrieben sowie den Abstand d zwischen den Rechnern, die Ausbreitungsgeschwindigkeit c und die Bitrate b.

e) Bestimmen Sie für alle drei Varianten die Gesamtverzögerung für die konkreten Werte $h = 100$, $v = 15$, $n = 10000$ und $m = 80$ bei einem Koaxialkabel mit einer Bitrate von b = 10 Mbps bei einer Entfernung zwischen Sender und Empfänger von 1000 Metern.

Hinweis: In elektrischen Leitern (verdrillte Kupferkabel, Koaxialkabel) erreicht man Ausbreitungsgeschwindigkeiten von ungefähr $2 \cdot 10^8 \frac{m}{s}$. Die Ausbreitungsverzögerung (propagation delay) T_p eines Mediums ist definiert durch den Abstand von Sender und Empfänger geteilt durch die Ausbreitungsgeschwindigkeit des Mediums. Die Übertragungsverzögerung (transmission delay) T_x ist definiert durch die Anzahl der übertragenen Bits geteilt durch die auf dem Medium realisierte Bitrate.

(3 + 3 + 3 + 3 + 3 Punkte)