

7. Übung

Ausgabe Abgabe Diskussion
29.11.13 06.12.13 09.12.-13.12.13

Bitte bei der Abgabe Name/Matr.Nr. der Mitglieder einer Gruppe, Nummer der Übung/Teilaufgabe und Datum auf den Lösungsblättern nicht vergessen! Darauf achten, dass die Lösungen beim richtigen Tutor abgegeben werden. Achten Sie bei Programmieraufgaben außerdem darauf, dass diese im Linuxpool kompilierbar sind.
Zu spät abgegebene Lösungen werden nicht berücksichtigt!

Aufgabe 1: Speicherverwaltung mit Buddy-Systemen (4+4=8 Punkte)

Gegeben sei ein Rechner, der mit 128 MByte Hauptspeicher ausgerüstet ist. Dieser Speicher ist im Moment leer und wird komplett mit einem Buddy-System verwaltet. Sinnvollerweise wird die Speicherbelegung mit Hilfe eines Binärbaumes dargestellt. Dabei stellt die Wurzel den gesamten Speicher von 128 MByte dar, die nächste Ebene zwei Segmente mit jeweils 64 MByte usw.

Der Reihe nach werden Speicherbereiche folgender Größe angefordert bzw. freigegeben:

Operation	Name	Größe
Anforderung	A	6 MB
Anforderung	B	4 MB
Anforderung	C	8 MB
Anforderung	D	16 MB
Anforderung	E	32 MB
Freigabe	A	
Anforderung	F	16 MB
Freigabe	E	
Freigabe	B	

- a) Stellen Sie die einzelnen Schritte als Binärbaum dar (Beginn nur mit Wurzel)! Hat der Algorithmus die Wahl zwischen zwei Wegen, so wählt er stets den linken.
- b) Nun werden gewichtete Buddies betrachtet. Hierbei wird ein Knoten nicht in zwei gleich große Äste aufgeteilt, sondern Knoten, deren Größe ohne Rest durch 3 teilbar ist, werden in Buddies im Verhältnis 1:2 aufgeteilt. Sind die Knoten durch 4, aber nicht durch 3 teilbar, werden Buddies im Größenverhältnis 1:3 erzeugt.

Wiederholen Sie die obigen Operationen, beginnend mit einem leeren Speicher von 128 MByte. Bei einer Speicheranforderung teilt der Algorithmus den kleinsten noch passenden Buddy so, dass der kleinere der entstehenden Buddies zum linken und der größere zum rechten Sohn im Baum wird. Sollten mehrere Buddies zum Teilen in Frage kommen, wird derjenige gewählt, der sich am weitesten links im Baum befindet.

Aufgabe 2: Working-Set (1+3+2+1=7 Punkte)

Das Working-Set $W(t, h)$ eines Prozesses zur Zeit t ist die Menge der Seiten, die bei den letzten h Zugriffen mindestens einmal referenziert wurden. Wird eine Seite aktuell gebraucht, dann befindet sie sich im Working-Set. Wenn sie nicht länger benötigt wird, wird sie nach h Zeiteinheiten aus dem Working-Set geworfen.

- a) Was sind die negativen Folgen, falls der Wert für h zu hoch bzw. zu niedrig gewählt wird?
- b) Es sei der folgende Referenzstring eines Prozesses gegeben:

$$\omega = 123442314243567432567334834873487887118871878187$$

Für diese Aufgabe gilt $h = 8$! Konstruieren Sie einen Graphen, auf dessen x-Achse die Zeit t (von 0 bis 48) 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.

- c) Aus Ihrem Graphen sollten sich nun verschiedene lokale Bereiche des Prozesses und die jeweiligen „Phasenwechsel“ (d.h. die Übergänge von einem lokalen Bereich zum nächsten) erkennen lassen. Geben Sie die Working-Sets der lokalen Bereiche an.
- d) Was versteht man unter *Thrashing*? Kann es bei Zuteilung von 5 Rahmen zum Thrashing kommen?

Aufgabe 3: Speicherverwaltung (1+1+1+1+1=5 Punkte)

Eine Speicherverwaltung benutzt Virtual-Paging mit TLB (Translation-Look-aside-Buffer). Ein TLB ist ein schneller Puffer, der als Cache für Zugriffe auf die Page-Table dient. Die Framenummer wird zunächst im TLB gesucht (100 Nanosekunden). Ist sie dort nicht vorhanden, wird in der Page-Table im Rechnerspeicher nachgesehen (500 Nanosekunden). In beiden Fällen wird, falls die Page im Speicher ist, anschließend der Speicher an der physikalischen Adresse ausgelesen (500 Nanosekunden). Sonst gibt es einen Seitenfehler, und die Seite muss von der Platte gelesen werden (was alles in allem 20 Millisekunden für den gesamten Page Fault dauert). Der TLB hat eine Hit Ratio von 80%, und falls die Seite nicht im TLB ist und in der normalen Page Table nachgesehen werden muss, führt dies in 1% der Fälle zu einem Seitenfehler.

Was ist die Zugriffszeit, falls

1. die Framenummer im TLB steht,
2. die Framenummer nicht im TLB steht, die Seite aber geladen ist,
3. die Framenummer nicht im TLB steht und ein Seitenfehler erzeugt wird.
4. Was ist die durchschnittliche Zugriffszeit?
5. Falls eine Seite nicht im TLB steht, wie häufig darf dann ein Seitenfehler vorkommen, damit die durchschnittliche Zugriffszeit unter 4 Mikrosekunden liegt?