

# 1 Thema: Speicherverwaltung

## 1.1 Strategien (12)

Für die Bewertung und den Vergleich von Speicherverwaltungsstrategien zum Belegen und Freigeben von zusammenhängenden Speicherbereichen seien die folgenden Eigenschaften von Interesse:

1. Ausnutzung des Speichers.
2. Art der Zerstückelung des Speichers, die damit verbundene Anzahl der Freibereiche und der Suchaufwand.
3. Aufwand zur Erstellung und Führung der Verwaltungsstrukturen.

Diskutieren Sie für die Strategien First Fit, Next Fit, Best Fit und Worst Fit jede dieser Eigenschaften. Analysieren Sie den schlechtesten Fall. Bei den Speicheranforderungen gehen wir davon aus, daß sie mit zufälligen Verteilungen eintreffen. Falls Sie zum Lösen der Aufgabe weitere Annahmen treffen, so beschreiben Sie diese.

## 1.2 Buddy Verfahren (9)

Das von Donald Knuth<sup>1</sup> entwickelte Buddy-Verfahren zur Verwaltung von Speicher nutzt die Binäradressierung aus, um effizient benachbarte Freibereiche zu einem grösseren Freibereich zusammenzufassen. Das Verfahren wird im folgenden kurz beschrieben.

Die mit diesem Verfahren verwalteten, zusammenhängenden Speicherbereiche können nur die Länge  $2^k$ , mit  $k = 0, 1, 2, \dots$  (meist  $k \geq 9 \Rightarrow$  kleinster Buddy 512 Byte), haben. Für einen Speicherbereich der Länge  $length$  wird ein Speicherbereich der Größe von  $2^r$ , mit  $r = \lceil \lg(length) \rceil$ , belegt. Der Gesamtspeicher habe eine Größe von  $2^M$ . Zur Verwaltung der Freibereiche existiert für jede mögliche Längen  $2^k$ , mit  $k = 0, 1, \dots, M$  eine Liste. Für eine Belegung eines Bereichs der Länge  $length$  wird in der entsprechenden Liste nach einem freien Bereich gesucht. Ist bei der Belegung kein Bereich passender Länge frei, so wird ein größerer Bereich genommen und dieser wird halbiert, wobei eine Hälfte in die zugehörige Frei-Liste aufgenommen wird, während der andere Teil entweder weiter halbiert oder entsprechend der gestellten Anforderungen reserviert wird. Bei der Halbierung entstehen zwei Buddies. Diese (und nur diese) können wieder zu einem größeren Freibereich zusammengefaßt werden, falls sie beide wieder frei sind. Zu Beginn existiert nur ein Freibereich der Länge  $2^M$ .

1. Skizzieren Sie die Belegung eines 1 MB großen Speichers, wenn nacheinander folgende Speicherbereiche angefordert bzw. freigegeben werden.
  - 70 KB belegen (A), 35 KB belegen (B)
  - 80 KB belegen (C), A freigegeben
  - 60 KB belegen (D), B freigegeben
  - D freigegeben, C freigegeben
2. Geben Sie in Programmiersprachlicher Notation die "Belege" und "Freigabe" Operationen an, die einen 1 MByte großen Speicher gemäß dem oben beschriebenen Buddy-Verfahren verwalten! Geben Sie dabei an, wie man sich bei der Zusammenlegung von Bereichen die Binärdarstellung der Adressen der Buddies zu Nutze machen kann.
3. Wie ist dieses Verfahren gegenüber den in der vorangegangenen Aufgabe untersuchten Verfahren einzustufen?

---

<sup>1</sup>D. Knuth: The Art of Computer Programming; Addison-Wesley; Second Edition, 1997

### 1.3 Wahl der Größe (6)

In dieser Aufgabe sollen Kriterien für die Wahl der Seitengröße  $p$  bei Seitenadressierung untersucht werden.

1. Unter der “internen Fragmentierung” bei Seitenadressierung versteht man denjenigen Hauptspeicheranteil, der dadurch ungenutzt bleibt, dass die Größe eines geladenen Programms kein ganzzahliges Vielfaches von  $p$  ist. Ein weiterer Teil des Hauptspeichers wird für die Führung der Seitentabellen benötigt.  
Es sei  $e$  die Größe eines Seitentableneintrags und  $s$  die durchschnittliche Größe eines Programms im Hauptspeicher. Bei welchem Wert für  $p$  wird die Summe “interne Fragmentierung” + “Größe der Seitentabelle” minimal? Nehmen Sie für diese Aufgabe eine lineare Verteilung für  $s \bmod p$  im Intervall  $[0,p)$  an.
2. Nennen Sie einige Gründe, die für die Wahl eines großen bzw. kleinen Wertes für  $p$  sprechen.
3. Welcher Wert ergibt sich nach Teilaufgabe 1 für die optimale Seitengröße  $p$  bei  $s = 32$  Kbyte und  $e = 16$  Byte?

## 2 Virtueller Speicher

### 2.1 Gründe für virtuellen Speicher (2)

Betriebssysteme verwalten den virtuellen Speicher. Was ist virtueller Speicher genau und welchen Nutzen hat er?

### 2.2 Segmentorientierter und Seitenorientierter Speicher (6)

Bei der Organisation des virtuellen Speichers gibt es zwei Ansätze:

- Segmentorientierter Speicher
- Seitenorientierter Speicher

Beschreiben Sie die beiden Konzepte und gehen Sie auf die Unterschiede ein. Veranschaulichen Sie beide Konzepte mit einem Diagramm und Beispiel.