Institut für Informatik

Arbeitsgruppe Autonome Intelligente Systeme

Freiburg, 7. Januar 2019

Systeme I: Betriebssysteme
Übungsblatt 10

Aufgabe 1 (200 Punkte)

Melden Sie sich zur Klausur *Systeme I: Betriebssysteme* an.

Bemerkung: Diese Aufgabe muss nicht auf Ihrem Lösungsblatt berücksichtigt werden. Die Punktevergabe erfolgt nachträglich und unabhängig von der Abgabe des Lösungsblatts.

Aufgabe 2 (1+1 Punkte)

Für die Scheduling-Strategie von UNIX wird unter anderem der Wert `CPU_USAGE` zur Bestimmung der Prioritäten benötigt. Dieser Wert kann durch den sogenannten Alterungs-Algorithmus rekursiv berechnet werden:

$$\texttt{CPU\_USAGE}\,(t) := \begin{cases} \texttt{CPU\_Anteil}(0) = 0 & \text{für } t = 0 \\ e^{-\frac{1}{T}} \cdot \texttt{CPU\_USAGE}\,(t-1) + \left(1 - e^{-\frac{1}{T}}\right) \cdot \texttt{CPU\_Anteil}\,(t) & \text{für } t > 0 \end{cases}$$

$\texttt{CPU\_Anteil}(t)$ ist dabei der von dem Prozess verbrauchte Anteil an Rechenzeit in der letzten Sekunde, also im Interval von $t-1$ bis t, und liegt zwischen 0 und 1.

a) Auf einem System sei $T := 3$ gewählt. Zum Zeitpunkt k war $\texttt{CPU\_USAGE}(k) = 0{,}32$. Im Zeitintervall k bis $k+1$ Sekunden hatte der Prozess einen CPU-Anteil von 48%, im Intervall $k+1$ bis $k+2$ einen Anteil von 96%, im Intervall $k+2$ bis $k+3$ einen Anteil von 28% und im Intervall von $k+3$ bis $k+4$ Sekunden einen Anteil von 52%. Berechnen Sie die `CPU_USAGE`-Werte dieses Prozesses zu den Zeitpunkten $k+1$, $k+2$, $k+3$ und $k+4$.

b) Wie wirken sich größere Werte von T auf die Berechnung und den zeitlichen Verlauf von `CPU_USAGE` aus?

Aufgabe 3 (3 Punkte)

Bei der *dynamischen Partitionierung* haben Sie die Speicherzuteilungsalgorithmen *Best Fit*, *First Fit* und *Next Fit* kennengelernt.

In einem Speicher sind die folgenden Blöcke frei (beachten Sie die Reihenfolge):

12 KB, 5 KB, 19 KB, 13 KB, 7 KB, 11 KB, 8 KB, 16 KB

Die letzte Speicherzuweisung erfolgte auf einen Bereich, der vor dem freien 12 KB-Block liegt. Es gibt folgende Speicheranforderungen:

1. 14 KB
2. 9 KB
3. 7 KB
4. 10 KB

Wird ein Block nur teilweise belegt, so wird der Rest des Blocks als neu entstandener freier Block an der selben Stelle in die Liste der freien Blöcke eingefügt (keine interne Fragmentierung), jedoch nicht mit einem anderen Block zusammengefasst.

Welche der freien Blöcke werden jeweils durch die einzelnen Algorithmen ausgewählt? Geben Sie auch für jeden Algorithmus die Liste der freien Blöcke im Endzustand (nach Abarbeitung aller Zuteilungen) an.

Aufgabe 4 (3+1+1 Punkte)

a) In der Vorlesung haben Sie das *Buddy-System* zur Speicherverwaltung kennengelernt. Mit diesem System werde ein zu Beginn vollständig unbelegter Speicher der Größe 16 MB verwaltet. Es sei $2^U = 16\,\text{MB}$ (Größe des größten zuteilbaren Blocks) und $2^L = 1\,\text{MB}$ (Größe des kleinsten zuteilbaren Blocks). Die Prozesse des Systems stellen nun folgende Speicheranforderungen:

   1. Anforderung A: 7 MB
   2. Anforderung B: 4 MB
   3. Anforderung C: 498 KB
   4. Freigabe A
   5. Anforderung D: 3 MB
   6. Freigabe B
   7. Freigabe C
   8. Freigabe D

Stellen Sie die Aufteilung und Belegung des Speichers nach jedem Schritt grafisch dar. Zeichnen Sie zusätzlich die zugrundeliegende Baumstruktur nach dem 6. Schritt und geben Sie die jeweilige Speichergröße an allen Knoten an. Für die übrigen Schritte müssen Sie die Baumstruktur nicht angeben.

b) *Interne Fragmentierung*: Angenommen, der verfügbare Hauptspeicher habe die Größe 2^U Bytes. Die Größe des größten zuteilbaren Blocks betrage 2^U Bytes, die des kleinsten zuteilbaren Blocks sei 2^L Bytes. Je nach Größe der Speicheranforderung teilt das Buddy-System einen Block passender Größe zu. Wie groß kann der „Verschnitt“ von Speicher durch interne Fragmentierung *innerhalb eines zugewiesenen Blocks* maximal werden? Wie groß muss die Speicheranforderung in diesem Worst-Case-Szenario gewesen sein und wie groß ist der zugeteilte Block? Gehen Sie davon aus, dass jede Speicheranforderung in ganzzahligen Bytes erfolgt.

c) *Externe Fragmentierung*: Angenommen, es seien insgesamt 2^{U-1} Bytes des Speichers belegt (evtl. in verschiedenen Blöcken). Geben Sie unter dieser Voraussetzung ein Worst-Case-Szenario für die externe Fragmentierung an, d.h. ein Beispiel für eine Speicherbelegung, bei der die Größe des größten noch verfügbaren Blocks minimal ist. Welche Struktur hat die Speicherbelegung insgesamt und welche Größe haben die verbleibenden freien Speicherblöcke? Wie kann diese Struktur entstehen?

Abgabe: Als PDF-Datei im Ilias bis zum 13. Januar 2019, 23:59 Uhr