

Vor- und Nachname, Mat.-Nr. _____ / _____

Ergebnis Aufgabe

1: _____ / 5
2: _____ / 10
3: _____ / 20
4: _____ / 10
5: _____ / 15
6: _____ / 15
7: _____ / 5
8: _____ / 10
9: _____ / 10

_____ / 100 *Summe*

Aufgabe 1: Speicher ist fundamentaler Bestandteil von Computern. Was wird im allgemeinen in folgenden Speichertypen gehalten:

- a) Hauptspeicher
- b) Festplatte
- c) Cache
- d) DVD-ROM
- e) Magnetband
- f) SD-Card

Wo (a-f) wird das Betriebssystem oder Teile davon gehalten?

5 Punkte

Aufgabe 2: In einem Rechenzentrum sind 1 Backup-Laufwerke, 1 Plotter, 1 Drucker und 1 3D-Drucker zur gemeinsamen Nutzung freigegeben: $E = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$. Es gebe 5 Benutzerprozesse, die bereits Ressourcen nutzen und auch noch weitere Ressourcen belegen wollen. Führen Sie den aus der Vorlesung bekannten Algorithmus zur Verklemmungs-erkennung durch. Die Belegungsmatrix B und die Wünschematrix C ist wie folgt gegeben:

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

10 Punkte

[illegible]

Aufgabe 4: Führen Sie die Berechnungen aus Aufgabe 3 für das Buddy-System durch:
Start A:3; Start B:3; Start C:2; Ende B; Start D:1; Start E:5.

Hinweis: es stehen insgesamt 32 Blöcke zur Verfügung.

10 Punkte

Buddy-System:

32

Aufgabe 5: Ausgehend von der virtuellen Seitenadressierung einer betagten PDP-11 mit 16Bit Adressierung wollen wir uns neben einstufigen auch 2- und 3-stufige Seitentabellen modernerer Architekturen ansehen. Der Page Offset sei jeweils 12 Bit.

(a) Wir betrachten einen Prozess, der lediglich eine Seite füllt. Führen Sie eine Berechnung für 16-, 32- und 64-Bit Architekturen durch, wobei bei den mehrstufigen Seitentabellen die Seitengröße jeder Stufe möglichst gleich sein soll. Geben Sie zuerst den qualitativen Bedarf (was wird benötigt) und dann den quantitativen Bedarf (wieviel Platz wird benötigt) an.

(b) Machen alle Kombinationen aus 1-, 2-, 3-stufigen Seitentabellen für 16-, 32-, 64-Bit Architekturen Sinn? Warum/Warum nicht?

(c) Wann werden 4-stufige Seitentabellen benötigt?

(d) Wie funktioniert die virtuelle Seitenadressierung allgemein?

15 Punkte

Aufgabe 6: Zeichnen Sie die Seitenfehlerereignisse der Referenzfolge 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, 4, 5, 6, 0, ...

- (a) für die “optimale Strategie” in den oberen Teil und
- (b) für die „NRU/RNU-Fifo Strategie“ in den unteren Teil der Vorlage ein. Wird zum ersten Mal auf Seite 6 zugegriffen, wird sie auch in diesem Schritt modifiziert. Beim hinteren Zugriff sollen die R-Bits in diesem Schritt gelöscht werden.
- (c) Erklären Sie die Arbeitsweise beider Strategien ganz allgemein.
- (d) Warum sind die ersten drei „page faults“ nicht so gravierend wie die weiteren?

15 Punkte

[illegible][illegible]

Aufgabe 7: In unserem tree.sh Shellskript wurde öfter die Anweisung:
`set "" $(ls -AdF $args 2>/dev/null)`
oder ähnlich ausgeführt. Was für einen Zweck hatte diese Anweisung?

5 Punkte

Aufgabe 8: Schreiben Sie ein Shellskript, das die Anzahl der normalen Dateien und die Anzahl der Verzeichnisse in einem Verzeichnis zählt. Das zu untersuchende Verzeichnis wird als erster Parameter erwartet. Fehlt dieser Parameter oder ist er ungültig, wird das aktuelle Verzeichnis als Ausgangspunkt genutzt.

10 Punkte

Aufgabe 9: Schreiben Sie ein C-Programm, das 8 Threads startet, die in regelmäßigen Abständen eine gemeinsame Variable „static volatile int count“ von 0 aufsteigend bis 100 hochzählen. Jede Erhöhung soll ausgegeben werden. Thread 1 darf die Variable jede Sekunden um 1, Thread 2 alle 2 Sekunden um 2, usw. bis Thread 8 alle 8 Sekunden um 8 erhöhen. Schützen Sie das Erhöhen und Ausgeben durch einen Mutex.

10 Punkte

Viel Erfolg!