

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

Ergebnis Aufgabe

1: _____ / 4
2: _____ / 12
3: _____ / 12
4: _____ / 6
5: _____ / 18
6: _____ / 9
7: _____ / 6
8: _____ / 6
9: _____ / 9
10: _____ / 6
11: _____ / 12

_____ / 100 *Summe*

Aufgabe 1: Wir hatten uns mit einer minimalen Rechnerarchitektur aus Prozessor, Hauptspeicher, Massenspeicher (Festplatte) und weiterer Peripherie beschäftigt. Was ist in diesem Zusammenhang die von-Neumann- bzw. Harvard-Architektur?

4 Punkte

Aufgabe 2: In einem Rechenzentrum sind 10 Backup-Laufwerke, 10 Plotter, 10 Drucker und 10 3D-Drucker zur gemeinsamen Nutzung freigegeben: $E = (10 \ 10 \ 10 \ 10)$. 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:

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

Ohne den aus der Vorlesung bekannten Algorithmus wie oben gefordert durchzuführen, kann man diese Aufgabe auch recht einfach anders lösen. Wie?

12 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 8 Blöcke zur Verfügung.

6 Punkte

Buddy-System:

8

Aufgabe 5: Was ist virtuelle Seitenadressierung (paging) und wie funktioniert sie?
Für Rechnerarchitekturen mit mehr als 16-Bit-Adressierung (32-Bit, 64-Bit usw.) hat man mehrstufige Seitentabellen implementiert. Warum?

Was ist der Hauptvorteil von mehrstufigen Seitentabellen?

Geben Sie ein Zahlenbeispiel dazu an.

Was ist deren Nachteil im Vergleich zu einstufigen Seitentabellen?

Wie wird dieser Nachteil meist kompensiert?

Wann wird man gänzlich auf eine virtuelle Seitentabelle verzichten?

18 Punkte

Aufgabe 6: Zeichnen Sie die Seitenfehlerereignisse der Referenzfolge 0, 1, 2, 30, 31, 1, 0, 1, 0, 40, 31, 3, 40, 31, 30, ...

- für die “optimale Strategie” in den oberen Teil und
- für die „Fifo Strategie“ in den unteren Teil der Vorlage ein.
- Erklären Sie die Arbeitsweise beider Strategien ganz allgemein.

9 Punkte

[illegible][illegible]

Aufgabe 7: Was sind „protection bits“?
Vervollständigen Sie untenstehende Tabelle.

6 Punkte

Symbolische Notation	Numerische Notation	Bedeutung

-rwx-----		
-rwxrwx---		
-rwxrwxrwx		
---x--x--x		
--w--w--w-		
--wx-wx-wx		
-r--r--r--		
-r-xr-xr-x		
-rw-rw-rw-		
-rwxr-----		

Aufgabe 8: Die zentrale Datenstruktur bei Unix-Dateisystem ist die i-node?
Wie ist sie aufgebaut?
Für große Dateien „wächst“ der Speicherplatz „für“ die i-node dynamisch.
Was ist damit gemeint?

6 Punkte

Aufgabe 9: Schreiben Sie ein BASH-Shellskript, das den Text „(c) Werthebach“ in normalen Dateien sucht. Als Übergabeparameter wird der Pfad angegeben, ab dem sowie in allen Unterverzeichnissen nach Dateien gesucht werden soll. Wird kein Übergabeparameter angegeben, wird im Home-Verzeichnis des Nutzers, der das Skript ausführt, gestartet. Treten Fehler bei der Abarbeitung des Skripts auf, wird -1 als Rückgabewert gesetzt, ansonsten die Anzahl gefundener Treffer.

9 Punkte

Aufgabe 10: Kindprozesse werden in UNIX meist mit der Befehlskombination `fork()/exec()` erzeugt. Erklären Sie die Funktionen einzeln und das Zusammenspiel miteinander. **6 Punkte**

Aufgabe 11: Gegeben sei ein global definiertes Integer Array „`int input[10.000]`“ für 10.000 Werte. Schreiben Sie ein C-Programm unter Linux, das 10 Threads startet, die jeweils prüfen wieviele Primzahlen in dem aktuell zu bearbeitenden Teilarray enthalten sind. Thread Nr. i , $i=1 \dots 10$, soll dabei die i -ten Tausend Werte testen, d.h. Thread Nr. 1 prüft `input[0]` bis `input[999]` und letztlich prüft Thread Nr. 10 `input[9000]` bis `input[9999]` auf Primzahlen. Das Programm soll als einzige Ausgabe die Gesamtanzahl aller Primzahlen im Array `input` ausgeben.

Eine Funktion `fill_input` – „`void fill_input(void)`“ –, die das komplette Array `input` anfänglich mit zufälligen ganzen Zahlen füllt, können Sie als gegeben voraussetzen. Ebenso eine Funktion `is_prim` – „`int is_prim(int)`“ –, die einen übergebenen Integer Wert auf Primzahl prüft.

12 Punkte

Viel Erfolg!