

Beispiellösung - Übungsblatt 4

Erstellt von Daniel Bub

Aufgabe 1.

Erklären Sie kurz und prägnant die nachstehenden Begriffe:

- a) Programm & Prozess
- b) Multitasking
- c) Quasiparallelität
- d) Befehlszeiger

Lösung 1.

a) **Programm & Prozess:**

Als Prozess bezeichnet man in der Regel ein Programm, welches sich in Ausführung befindet. Jeder Prozess besitzt unter anderem einen *Befehlszähler*, einen Adressraum, die Inhalte der zum Prozess gehörenden *Prozessor-Register*, das *Programmsegment*, welches den ausführbaren Code des Programms enthält, etc.

Ein **Programm** ist eine Datei, deren Inhalt Regeln und Anweisungen einer bestimmten Programmiersprache folgt.

b) **Multitasking:**

Von Multitasking ist dann die Rede, wenn temporär mindestens zwei Prozesse zur Ausführung bereitstehen. Es gibt zwei Arten von Multitasking: das **echte** Multitasking, bei dem mindestens zwei CPUs zur Ausführung bereitstehen und das *Quasi*-Multitasking, wo zur Ausführung der Prozesse nur eine CPU zur Verfügung steht (siehe Quasiparallelität in der nächste Teilaufgabe).

c) **Quasiparallelität:**

Besitzt ein System nur einen Prozessor und existieren mehr als nur ein auszuführender Prozess, so kann *Quasiparallelität* erreicht werden.

Quasiparallelität kann über geschicktes Multiprogramming bzw. Time-Sharing realisiert werden. Hierbei wechselt die CPU sehr schnell (im Bereich von Millisekunden) zwischen den einzelnen Prozessen hin und her, so dass für den Nutzer der Eindruck entsteht, die Prozessverarbeitung läuft parallel ab.

d) **Befehlszeiger:**

Der *Befehlszeiger* ist eine Hardware-Komponente des Computers und zeigt auf die aktuell auszuführende Befehlsanweisung im Speicher eines Prozesses. Wird ein Prozess A durch einen zweiten Prozess B unterbrochen, so muss die Position des Befehlszeiger von Prozess A zwischengespeichert werden, um nach der Beendigung / Unterbrechung von Prozess B wieder an die korrekte Ausführungsanweisung von Prozess A zu gelangen.

Aufgabe 2.

Ein Betriebssystem läuft auf einem Server mit vier Prozessoren. Während des Betriebs wurde gemessen, dass die 7 Prozesse jeweils 10% ihrer Zeit auf Ein-/Ausgabe warten.

- Wie hoch ist die zu erwartende CPU-Auslastung
- Wie verändert sich die CPU-Auslastung, wenn die 7 Prozesse jeweils 20% bzw. 40% ihrer Zeit auf Ein-/Ausgabe warten?

Lösung 2.

Zur Erinnerung:

Laufen auf einem System n Prozesse, so beträgt die Wahrscheinlichkeit, dass sich alle Prozesse in einem wartenden Zustand befinden p^n (unabhängig der CPU-Anzahl).

- Die zu erwartende CPU-Auslastung beträgt somit:

$$\text{CPU-Auslastung}_{10\%} = 1 - p^n = 1 - 0.1^7 = 0.9999999 \hat{=} 99.99999\%$$

- analog zu Teilaufgabe a) ergeben sich hier die Werte:

$$\text{CPU-Auslastung}_{20\%} = 1 - 0.2^7 = 0.9999872 \hat{=} 99.99872\%$$

$$\text{CPU-Auslastung}_{40\%} = 1 - 0.4^7 = 0.9983616 \hat{=} 99.83616\%$$

Aufgabe 3.

Auf einem Betriebssystem läuft eine parallele Software für astrophysikalische Berechnungen. Das Betriebssystem verwaltet dabei mehrere CPUs, die für das Berechnungsprogramm benutzt werden. Verwenden Sie für $T_1 = 10$ Sekunden bzw. den Wert 1, da sich der Wert kürzen lässt. Berechnen Sie den theoretisch möglichen **Speedup** für die folgenden Fälle:

Anzahl CPUs	2	8	32	128	1024
75% Anteil serieller Code					
25% Anteil serieller Code					

Freiwillige Zusatzausgabe:

Halten Sie ein Upgrade von 8 auf 32 bzw. von 128 auf 1024 CPUs für sinnvoll (Kosten-Nutzen-Rechnung)?

Lösung 3.

Zur Erinnerung:

Der Beschleunigungsfaktor S_p ist definiert als

$$S_p = \frac{T_1}{T_p} = \frac{T_1}{T_1 * ((1-f) + \frac{f}{p})},$$

wobei p = Prozessoranzahl, S_p = theor. möglicher Speedup, T_1 = Ausführungszeit auf einem Ein-Kern-System, T_p = Ausführungszeit auf einem p -Prozessorsystem und f = parallelisierbarer Code-Anteil des Prozesses.

Somit erhalten wir folgende Tabellenwerte:

Anzahl CPUs	2	8	32	128	1024
75% Anteil serieller Code	1.14	1.28	1.32	1.330	1.333
25% Anteil serieller Code	1.6	2.9	3.66	3.91	3.99