

1. Grundlagen und Unterscheidung Prozess ↔ Thread

- a) Erklären Sie, was ein Thread ist
- b) Nennen Sie drei Vorteile von Threads
- c) Weshalb muss ein Betriebssystem neben dem Thread-Modell auch das Prozess-Modell unterstützen?
- d) Nennen Sie die wesentlichen Unterschiede zwischen dem Thread-Modell und dem Prozess-Modell
- e) Aus welchen Bestandteilen besteht:
 - a) Ein Prozess?
 - b) Ein Thread?
- f) Was sind Prozess-Ressourcen, die Threads innerhalb eines Prozesses gemeinsam teilen?
- g) Inwiefern bringt die gemeinsame Nutzung von Ressourcen innerhalb eines Prozesses Vorteile, und inwiefern Nachteile?

2. Implementierung des Multithreading

- a. Unterscheiden Sie die Begriffe User-Level-Thread und Kernel-Level-Thread
- b. Nennen Sie drei grundsätzliche Möglichkeiten zur Implementierung des Multithreading und erklären Sie, wie dabei UL-Threads auf KL-Threads abgebildet werden
- c. Bei einer der Implementierungs-Varianten blockieren bei einem Warten auf I/O innerhalb eines KL-Threads alle Threads eines UL-Threads.
 - a. Um welche Implementierungs-Variante handelt es sich?
 - b. Weshalb kann mit dieser Implementierungs-Variante dennoch eine parallele Verarbeitung erreicht werden?
- d. Bei einer m:n-Abbildung wird für die KL-Threads ein Thread-Pool verwendet.
Welche Vorteile ergeben sich dadurch?
- e. Wie unterscheidet sich bei einer m:n-Abbildung der Ansatz der Scheduler Activation von dem der Pop-up-Threads?

3. Threads in der Praxis

- a. Welche Probleme können sich bei der Verdrängung eines Threads ergeben, wenn sich dieser mit anderen Threads gemeinsam Ressourcen teilt?
- b. Erklären Sie das Task-Konzept moderner Programmiersprachen
- c. Welche Vorteile bietet die Arbeit mit Tasks gegenüber der direkten Verwendung von Threads?