

Fragen zur Systemprogrammierung

Teil C – Vorlesung 10.3

Prozesssynchronisation / Semaphore

1. Welches Wissen bringe ich aus den vorangegangenen Vorlesungen mit?¹
2. Was ist ein Semaphor, wozu dient er, welche Operationen mit welcher Semantik stellt er zur Verfügung und welche zwei Hauptarten werden wie unterschieden? (S. 6)
3. Wie kann ein begrenzter Puffer mit Hilfe von Semaphore implementiert werden, welche Semaphorart wird dabei wofür verwendet? (S. 8/9)
4. Weshalb bilden die Programme für *P* und *V* kritische Abschnitte, welche kritischen Wettlaufsituationen können sich hier durch gleichzeitige Prozesse ergeben? (S. 10)
5. Wie lassen sich die Programme für *P* und *V* gegenüber gleichzeitigen Prozessen absichern? Welche Randbedingungen sind dabei zu beachten? (S. 11)
6. Inwiefern kann das Monitorkonzept als Programmierkonvention zur Absicherung der Programme für *P* und *V* dienen? (S. 12)
7. Aus welchen Datenstrukturen und Variablen als Mitglieder lässt sich ein Semaphormodell in der Programmiersprache C aufbauen, wozu dienen die einzelnen Bestandteile? (S. 13)
8. Welche Schritte sind zur Prozessblockade zu gehen, um die Aufhebung der Wartebedingung zu erwarten? Welche wettlaufkritische Aktionsfolge zeigt sich dabei? (S. 15/16)
9. Wie ist das Schlafenlegen eines Prozesses, der die Aufhebung seiner Wartebedingung erwartet, technisch durchzuführen? Welche Hauptschritte legen einen Prozess schlafen und welcher davon erfolgt hier noch innerhalb des kritischen Abschnitts? (S. 17)
10. Welche Gemeinsamkeiten und Unterschiede bestehen zwischen Semaphore und Mutex auf konzeptioneller und technischer Ebene? (S. 19/20)
11. Wie lässt sich ein Mutex durch einen Semaphore implementieren, welche Art von Semaphore ist dabei nur relevant und was für eine Reaktion ist bei unberechtigter Freigabe eines Mutex im Gegensatz zum Semaphore zweckdienlich? (S. 21/22)
12. Welche Maßnahmen zur Vorbeugung gleichzeitiger Prozesse können die Atomarität der Programme für *P* und *V* physisch gewährleisten, welche Vor- und Nachteile sind damit jeweils verbunden? (S. 24)
13. Wie stellen sich die verschiedenen Maßnahmen auf programmiertechnischer Ebene dar, was ist bei Verschachtelungen – von kritischen Abschnitten, die letztlich durch *P/V* geschützt werden – zu beachten, welche Aktion oder Aktionsfolge wird durch welche Sperrmaßnahme verzögert und was kann diese Verzögerung jeweils zur Folge haben? (S. 25–27)
14. Welcher Stoff wurde in dieser Vorlesung vermittelt, was sollte ich gelernt haben?¹

¹Diese Frage ist nicht prüfungsrelevant.