

# Grundlagen der Betriebssysteme

## Blatt 07

### Gruppe 055

Marco Deuscher  
Ibrahim Hasan

Juni 2019

## 1 Deadlocks

(a) Ein Szenario in dem keines der Fahrzeuge die Kreuzung überqueren kann, wäre wenn alle vier Fahrzeuge auf die erste Spur vorfahren. Dann wird jedes der Fahrzeuge durch ein anderes blockiert, so dass alle Fahrzeuge warten.

(b) wie ist dieser Fall zu verstehen

(c)

- Fahrzeuge entsprechen Aktivitätsträger
- Kreuzung entsprechen geteilten Ressourcen
- Warten entspricht blockiert sein
- Nie überqueren entspricht dem starvation Fall, in dem ein Aktivitätsträger dauerhaft blockiert ist und nie mehr zum laufen kommt

(d) Wenn ein Auto fahren möchte, fährt es bis zur ersten Spur nach vorne und versucht ob es die Kreuzung überqueren kann. Ist dies möglich, so überquert das Auto die Kreuzung. Kann er nicht fahren, weil die zweite Spur blockiert ist, fährt er wieder zurück, bis er wieder in seiner Haltebucht steht. Dort wartet er dann eine zufällig lang gewählte Zeit bevor er es erneut versucht.

## 2 Semaphore und aktives Warten (Spinlock)

- ein Semaphore ruft in der Regel yield auf, so dass der Aktivitätsträger nicht die CPU belastet. Dies ist ein Vorteil, da dann diese CPU Zeit für Prozesse genutzt werden kann, so dass der blockierte Aktivitätsträger schneller auf die Ressource zugreifen kann

- ein Spinlock implementiert busy-waiting, so dass dieser die CPU nicht abgibt. Ein Spinlock kann bspw. in einer ISR verwendet werden, da dort der Aktivitätsträger nicht schlafen gelegt werden kann
- (b) Spinlocks eignen sich nicht auf Einprozessorgeräten, da diese die CPU blockieren. Auf einem System mit nur einem Kern, verliert man damit effektiv nur Rechenzeit und hat keinen Vorteil.
- (c) Die Operationen P und V selbst müssen nicht atomar sein, allerdings muss das Erhöhen oder Erniedrigen des Semaphores atomar ablaufen. Dafür benötigt man eine Instruktion, welche innerhalb einer Instruktion eine Speicherzelle lesen und schreiben kann.