

Semaphore

- **Vorteil:** es wird aufs **Aktive Warten** verzichtet und somit wird weniger **CPU-Zeit** verschwendet, indem der wartende Prozess durch einen **Kontextwechsel** in den **Zustand Blockiert** gebracht wird

Datenstruktur, welche **ganzzahligen Zähler**, **Warteschlange** bereitstellt und zwei **atomare Operationen P() und V()** definiert.

- der **ganzzahliger Zähler** (int) kann keine negativen Werte annehmen.
- **Warteschlange** ist Datenstruktur, die nach dem FIFO-Prinzip arbeitet und als zur geordneten Aufnahme von Prozessen dient.
- Die **Operation P()** wird in einigen Quellen auch als **down()**-Operation betitelt
 - Hat die Zählervariable einen positiven Wert (>0), dann verringere den Wert der Zählvariablen um 1.
 - Wenn Zählvariable den Wert 0 hat, dann blockiere aufrufenden Prozess und fügt ihn an das Ende ihrer Warteschlange
- analog **up()** anstatt **V()**.
 - wenn die Warteschlange nicht leer ist, dann entblockiere den ersten Prozess der Warteschlange.
 - ist die Warteschlange leer, dann erhöhe den Wert der Zählervariable um 1.
 - Mit "entblockieren" wird der erste Prozess aus der Warteschlange in den **Zustand "Bereit"** geändert.

ein **binären Semaphor** ist ein Semaphor, dessen ganzzahliger Zähler nur die Werte 0 oder 1 annehmen kann.

1. **Mutex** ist ein binären Semaphor, der mit einem **wechselseitigen Ausschluss** arbeitet und das **aktive Warten** ersetzt
 - Wenn ein Thread in **kritischen Abschnitt** befindet, darf andere Thread nicht in seinen **kritischen Abschnitt** eintreten
 - **Reihenfolgedurchsetzung** ist dabei sehr wichtig: Erst wenn erste Prozess Zugriff erledigt hat, darf der zweite
2. **Zählsemaphor** ist ein Semaphor, der den Zugriff auf eine begrenzte Anzahl **eines** Betriebsmittels regelt.
 - Der Zugriff auf das Betriebsmittel stellt deshalb einen **kritischen Abschnitt** dar und muss geschützt werden.

- **Erzeuger- / Verbraucherproblem:** Erzeuger erzeugt etwas, was Verbraucher verbraucht. Natürlich kann Verbraucher erst etwas verbrauchen, wenn zuvor Erzeuger etwas erzeugt hat. Erzeuger kann nur soviel erzeugen, bis alle Lagerplätze für Erzeugnis **belegt** sind. Dann muss Verbraucher wieder verbrauchen, bis Erzeuger wieder etwas erzeugen kann.

Ein Semaphore hat keinen ungünstigen Moment, weil die P() Methode atomar ist!

Revision #2

Created 24 September 2022 16:23:31 by Merith Holtmann

Updated 2 October 2022 20:48:55 by Merith Holtmann