Übungen
zur Vorlesung
Betriebssysteme
Prof. Dr. Kornmayer

Übungen 3 - Interprozesskommunikation

Aufgabe 3.1: Nennen Sie die Bedingungen für wechselseitigen Ausschluss!

Aufgabe 3.2: Beschreiben Sie in einem kurzen Abriss, wie ein Betriebssystem, das Interrupts ausschalten kann, Semaphoren realisieren könnte?

Aufgaben 3.3 und 3.4: : Erzeuger-Verbraucher Problem
In den folgenden beiden Aufgaben soll das Erzeuger-Verbraucher Problem näher betrachtet werden. Der beschränkte Puffer soll als Ringpuffer (Array) realisiert werden. Der Erzeuger soll 100-mal in den Ringbuffer schreiben und der Verbraucher soll 100-mal aus dem Ringpuffer lesen. Die Kapazität des Ringbuffers soll auf 5 beschränkt bleiben. Machen Sie sich nochmals klar, was diese Anforderungen bzw. Randbedingungen bedeuten!

Aufgabe 3.3: Erzeuger-Verbraucher mit Java-Threads
Realisieren Sie eine Lösung für das Erzeuger-Verbraucher Problem mit Hilfe von Java Threads, indem Sie jeweils eine Klasse für den Erzeuger und für den Verbraucher erstellen. In einer weiteren Klasse soll dann jeweils ein Erzeuger und ein Verbraucher gestartet werden. Schauen Sie sich dazu nochmals die JavaDoc von Threads an!

- Laden Sie zuerst den Code mit dem Ringpuffer von der Webseite herunter.
- Erstellen Sie die benötigten Klassen unter Verwendung von JAVA Threads
- Ist diese Lösung frei von Deadlook? Begründen Sie ihre Antwort!
- Erzeugen Sie jeweils zwei Erzeuger bzw. Verbraucher Instanzen und führen Sie das Programm nochmals aus! Was beobachten Sie? Ist die Lösung frei von Verklemmungen?

Aufgabe 3.4: Erzeuger-Verbraucher mit Semaphoren
Realisieren Sie eine Lösung für das Erzeuger-Verbraucher Problem mit Hilfe von Semaphoren unter JAVA. Eine einfache Implementierung von Semaphoren steht mit dem BSync Paket auf der Downloadseite zur Verfügung. Erstellen Sie jeweils eine Klasse für den Erzeuger und den Verbraucher. Für die Synchronisation wird eine weitere Klasse erstellt.

- Laden Sie zuerst den Code mit dem Ringpuffer von der Webseite herunter.
- Erstellen Sie die benötigten Klassen unter Verwendung von JAVA Threads
- Ist diese Lösung frei von Deadlook? Begründen Sie ihre Antwort!
- Erzeugen Sie jeweils zwei Erzeuger bzw. Verbraucher Instanzen und führen Sie das Programm nochmals aus! Was beobachten Sie? Ist die Lösung frei von Verklemmungen?

Aufgabe 3.5: Dateien als Sperrmechanismus
Schreiben Sie ein C-Programm, das den Zugriff auf eine Ressource durchführen will. Die Ressource soll durch ein Lockfile (z.B. /tmp/.fileSema) geschützt werden. Die Erzeugung der Datei geschieht mit open() unter von write-only und exclusiv Optionen. Der Lock wird mit dem Befehl unlink() wieder freigegeben.