

Informatik 3 - Aufgabenblatt 2

Aufgabe 3:

1. Was ist der Unterschied zwischen kooperativem und konkurrierendem Threading?

Threads arbeiten kooperativ, wenn diese an einer Aufgabe gleichzeitig arbeiten. Konkurrierendes Threading ist das unabhängige Arbeiten von Threads. Diese kämpfen dann um einen beschränkten Prozessorraum mit anderen Threads.

Bei Aufgabe 1 arbeiten die jeweiligen Producer- und Consumer-Threads beim Befüllen (Producer) und entnehmen der Elemente (Consumer) des Buffers kooperativ. Sie konkurrieren jedoch, wenn der Consumer ein Element entnehmen möchte, aber gleichzeitig der Producer etwas einfügen möchte.

2. Erklären Sie in eigenen Worten die Begriffe Synchronisation, Kommunikation und Koordination.

Synchronisation: Eine Synchronisation im Bezug auf Threads ist das Steuern des zeitlichen Ablaufs von mehreren nebenläufigen Threads. So muss ein Thread warten, wenn dieser eine Lösung eines anderen Threads braucht.

Kommunikation: ... ist die Methodik Informationen, Signale, Botschaften, Lösungen, Ereignisse... mit anderen Threads auszutauschen.

Koordination: Der Oberbegriff von Synchronisation und Kommunikation ist die Koordination. Erwartet ein Thread ein Ereignis eines anderen, wird dieser gesperrt (Lock) bis der andere Thread eine Meldung gibt, dass er fertig ist.

3. Was ist der Unterschied zwischen scheinbarer und echter Nebenläufigkeit? Nennen Sie die Voraussetzung für echte Nebenläufigkeit.

Die scheinbare Nebenläufigkeit ist eine Illusion, dass Prozesse gleichzeitig arbeiten. Tatsächlich verarbeitet der Prozessor (Einprozessormaschine) die Prozesse nicht gleichzeitig, sondern teilt die Prozesse auf, welche dann abgearbeitet werden. Der Scheduler des Betriebssystems regelt dies indem er zwischen den verschiedenen Prozessstücken im Millisekundentakt umschaltet. Eine Nebenläufigkeit wird simuliert.

Die echte Nebenläufigkeit kann nur mithilfe eines Mehrkernprozessors realisiert werden. Jeder Prozess bekommt einen eigenen Prozessorbereich, in denen die Prozesse tatsächlich parallel abgearbeitet werden.

4. Erklären Sie das Lost-Update- und das Dirty-Read-Problem.

Das Lost Update Problem tritt dann auf, wenn mehrere Prozesse zeitgleich an etwas, wie zum Beispiel einer Zahl arbeiten. Sie holen sich diese Zahl. Alle treffen dann eine Änderung an dieser Ausgangszahl. Ein Prozess lädt sein Ergebnis hoch und hat somit die Ausgangszahl verändert. Doch durch das Verändern anderer Prozesse der alten Ausgangszahl und deren Hochladen, gehen Veränderungen alter Prozesse verloren.

Das Dirty Read Problem ist ähnlich wie das Lost Update Problem, doch es geht nicht um das Schreiben sondern Lesen. Eine Zahl wird von einem Prozess ausgelesen, welche zeitgleich von einem anderen Prozess verändert wird. Nun hat der Lese-Prozess ein „nicht aktuelles Ergebnis“. Man nennt diese Zahl dann „verschmutzt“.