Rechnerstrukturen

Vorlesung im Sommersemester 2008

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Cache-Kohärenz
    - sichert, dass mehrere Prozessoren eine kohärente Sicht auf den Speicher haben
  - Wichtige Frage:
    - Wann muss ein Prozessor den Wert sehen, den ein anderer Prozessor aktualisiert hat?
    - Oder:
      - In welcher Reihenfolge muss ein Prozessor die Schreiboperationen eines anderen Prozessors beobachten?
    - Oder
      - Welche Bedingungen zwischen Lese- und Schreiboperationen auf verschiedene Speicherstellen durch verschiedene Prozessoren müssen gelten?

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Problem: Was kann passieren?

```
P1:     A=0;                          P2:     B=0;
        …                                     …
        A=1;                                  B=1;
L1:     if (B==0)                     L2:     if (A==0)
                …führe Aktion a2 aus                  …führe Aktion a1 aus
```

Mögliche Fälle:

a1 wird ausgeführt und a2 nicht
a2 wird ausgeführt und a1 nicht
a1 und a2 werden beide nicht ausgeführt
a1 und a2 werden beide ausgeführt

Ursache für dieses Verhalten:

- Verzögerungen der Schreiboperationen im Schreibpuffer
- Verzögerungen im Netzwerk

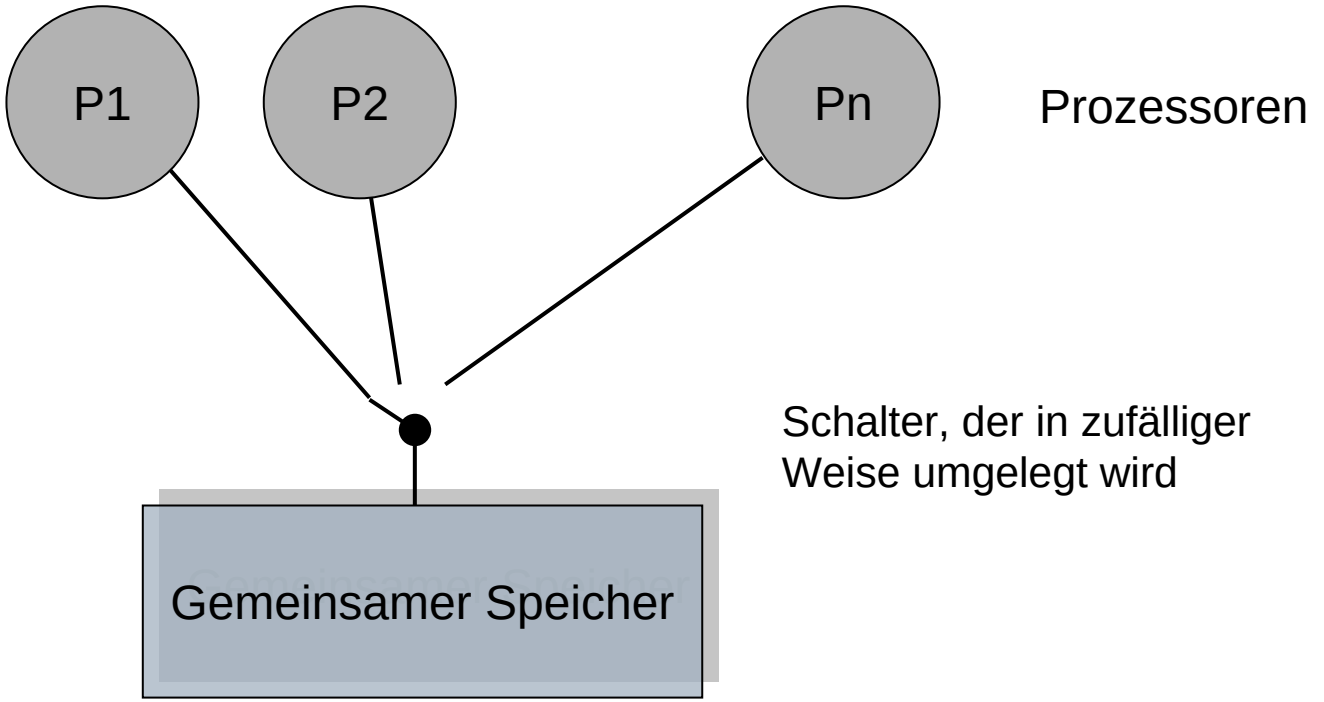
Soll dieses Verhalten erlaubt sein, und wenn ja, unter welchen Bedingungen?

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Spezifizieren die Reihenfolge, in der Speicherzugriffe eines Prozesses von anderen Prozessen gesehen werden
  - Sequentielle Konsistenz
    - Ein Multiprozessorsystem heißt sequentiell konsistent, wenn das Ergebnis einer beliebigen Berechnung dasselbe ist, als wenn die Operationen aller Prozessoren auf einem Einprozessorsystem in einer sequentiellen Ordnung ausgeführt würden. Dabei ist die Ordnung der Operationen der Prozessoren die des jeweiligen Programms.
    - Alle Lese- und Schreibzugriffe werden in einer beliebigen sequentiellen Reihenfolge, die jedoch mit den jeweiligen Programmordnungen konform ist, am Speicher wirksam.
    - Entspricht einer überlappenden sequentiellen Ausführung sequentieller Operationsfolgen anstelle einer parallelen Ausführung

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Sequentielle Konsistenz
    - Veranschaulichung der sequentiellen Konsistenz



Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Sequentielle Konsistenz
    - Programmierer geht von sequentieller Konsistenz aus
    - Führt aber zu sehr starken Einbußen bzgl. Implementierung und damit der Leistung
      - Verbietet vorgezogene Ladeoperationen, nichtblockierende Caches
  - Abgeschwächte Konsistenzmodelle
    - Konsistenz nur zum Zeitpunkt einer Synchronisationsoperation
      - Lese- und Schreiboperationen der parallel arbeitenden Prozessoren auf den gemeinsamen Speicher zwischen den Synchronisationszeitpunkten können in beliebiger geschehen.

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Definitionen
    - Ein Lesezugriff durch Prozessor Pi heißt zu einem bestimmten Zeitpunkt bezüglich Pk ausgeführt, wenn ein Schreibzugriff durch Prozessor Pk den Wert, den Pi durch den Lesezugriff auf dieselbe Adresse erhält, nicht mehr beeinflussen kann
    - Ein Schreibzugriff durch Pi heißt zu einem bestimmten Zeitpunkt bezüglich Pk ausgeführt, wenn ein Lesezugriff durch Pk auf dieselbe Adresse den Wert liefert, der von Pi geschrieben worden ist
    - Ein Zugriff gilt als ausgeführt, wenn er bezüglich aller Prozessoren im System ausgeführt ist
    - Ein Lesezugriff heißt global ausgeführt, wenn sowohl er als auch der Schreibzugriff, der den gelesenen Wert erzeugt, ausgeführt worden ist

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Definitionen
    - Unterschied zwischen ausgeführten und global ausgeführten Lesezugriff nur in Systemen, in denen der Schreibzugriff kein atomarer Vorgang ist, d.h. wenn der geschriebene Wert für alle Prozessoren des Systems nicht gleich lesbar ist
  - Sequentielle Konsistenz:
    - Hinreichende Bedingung:
      - Bevor ein Lese- oder Schreibzugriff bezüglich eines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Lesezugriffe global ausgeführt und alle vorhergehenden Schreibzugriffe ausgeführt sein
    - Reihenfolge der Operationen auf einem Prozessor wird beibehalten, und für alle anderen Prozessoren ist dieselbe Reihenfolge sichtbar.
    - Jeder Lese- und Schreibzugriff muss allen anderen Prozessoren vor einem nachfolgenden Lese- oder Aschreibzugriff bekannt gemacht werden
    - Wenig Spielraum für Optimierungen der Speicherzugriffe
      - Z.B. Puffern der Schreibzugriffe

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Prozessorkonsistenz
    - Definition (nach Goodman, 1989)
      - Ein Prozessor ist prozessorkonsistent, wenn das Ergebnis irgendeiner Ausführung dasselbe ist, als wenn die Operation eines jeden Prozessors in der sequentiellen Reihenfolge, die durch sein Programm bestimmt wird, erscheinen
    - Unterschied zur sequentiellen Konsistenz
      - Bei der Prozessorkonsistenz muss es nicht mehr für alle Prozessoren eine einheitliche Reihenfolge der Speicherzugriffe geben
      - Schreibzugriffe zweier Prozessoren können von einem dritten Prozessor in einer anderen Reihenfolge gesehen werden, als von den beiden schreibenden Prozessoren
      - Die Schreibzugriffe eines Prozessors werden jedoch von allen Prozessoren in der in seinem Programm angegebenen Reihenfolge gesehen

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Prozessorkonsistenz
    - Definition (nach Gharachorloo, 1990)
      - Bedingung der globalen Ausführung der Lesezugriffe wird fallen gelassen
      - Es gilt:
        - » Bevor ein Lesezugriff bezüglich eines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Lesezugriffe ausgeführt worden sein
        - » Bevor ein Schreibzugriff irgendeines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Schreibzugriffe ausgeführt worden sein

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Prozessorkonsistenz
    - Definition (nach Gharachorloo, 1990)
      - Konsequenz
        - » Prozessorkonsistenz führt nicht mehr unbedingt zur korrekten Sequentialisierung der Speicherzugriffe, da das Lesen schon erlaubt ist, bevor die vorhergehenden Schreibzugriffe alle ausgeführt worden sind.
        - » Lesezugriff kann zwar von den anderen Prozessoren nicht beobachtet werden, der Prozessor selbst sieht jedoch eine Reihenfolge der Speicheroperationen, die nicht seiner Programmordnung entspricht
        - » Schreibender Prozessor muss nicht auf die Ausführung des Schreibzugriffs bezüglich aller Prozessoren warten, bevor er eine weitere Schreiboperation veranlassen kann
        - » Puffern von Schreibzugriffen ist wieder erlaubt

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Schwache Konsistenz
    - Bisherige Konsistenzmodelle lassen Synchronisation paralleler Threads außer Acht
    - Konkurrierende Zugriffe auf gemeinsame Daten werden durch geeignete Synchronisationen geschützt

```
mutex m;
…
lock(m)
y=0;
x=0;
unlock(m)
…
P1:     lock(m)
        A=1;
        if (B==0)
                …führe Aktion a2 aus
        unlock(m)
```

```
P2:     lock(m)
        B=1;
        if (A==0)
                …führe Aktion a1 aus
        unlock(m)
```

Multiprozessor mit gemeinsamem Speicher

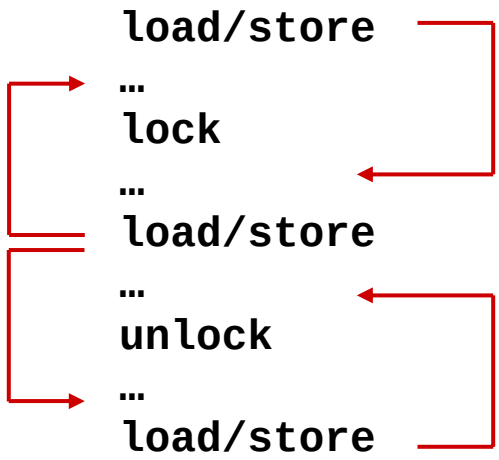
- Speicherkonsistenzmodelle
  - Schwache Konsistenz (weak consistency)
    - Idee
      - Die Konsistenz des Speicherzugriffs wird nicht mehr zu allen Zeiten gewährleistet, sondern zu bestimmten, vom Programmierer in das Programm eingesetzten Synchronisationspunkten
      - Einführung von kritischen Bereichen
        - » Innerhalb dieser Bereich wird die Inkonsistenz der gemeinsamen Daten zugelassen
        - » Voraussetzung: konkurrierende Lese-/Schreibzugriffe sind durch den kritischen Bereich unterbunden

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Schwache Konsistenz (weak consistency)
    - Bedingungen
      - Bevor ein Schreib- oder Lesezugriff bezüglich irgendeines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Synchronisationspunkte erreicht worden sein
      - Bevor eine Synchronisation bezüglich irgendeines anderen Prozessors ausgeführt werden darf, müssen alle vorhergehenden Schreib- oder Lesezugriffe ausgeführt worden sein.
      - Synchronisationspunkte müssen sequentiell konsistent sein
        - » *bevor* und *vorübergehend* beziehen sich auf die Programmordnung

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Schwache Konsistenz (weak consistency)
    - Auswirkung
      - Synchronisationsbefehle stellen Hürden dar, die von keinem Lese- oder Schreibzugriff übersprungen werden



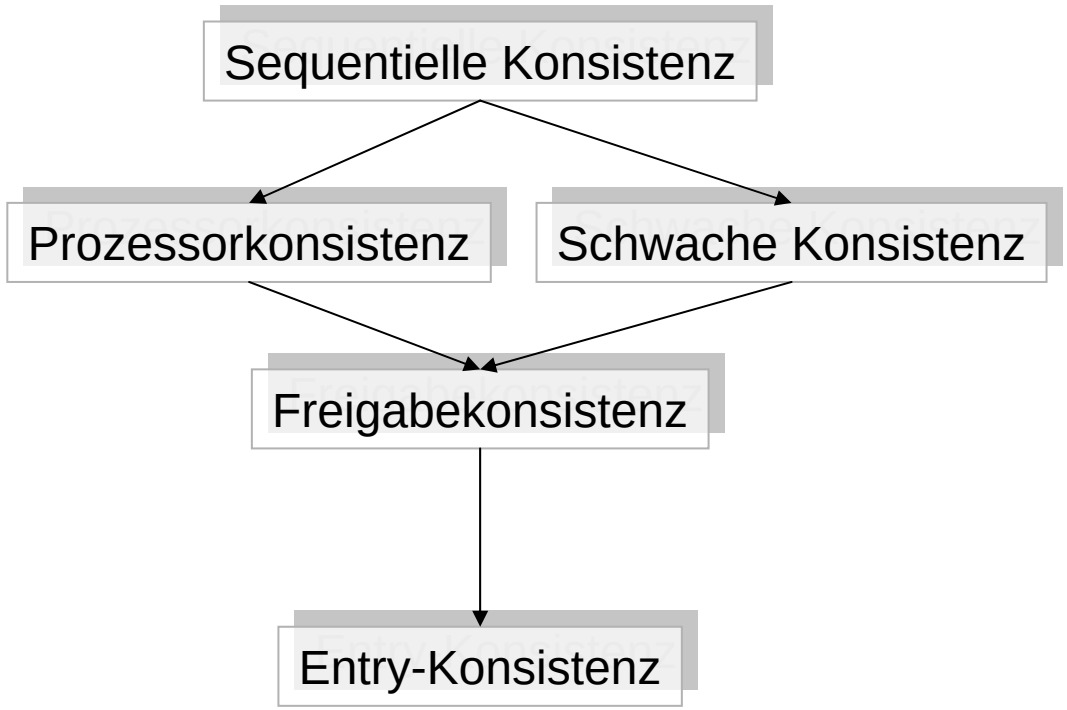
Rote Pfeile: verboten

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Schwache Konsistenz (weak consistency)
    - Auswirkung
      - Voraussetzung für die Implementierung der schwachen Konsistenz ist die hardware- und softwaremäßige Unterscheidung der Synchronisationsbefehle von den Lade- und Speicherbefehlen und eine sequentiell konsistente Implementierung der Synchronisationsbefehle
      - Das Puffern von Schreibzugriffen ist erlaubt, das von Synchronisationsbefehlen nicht!
      - „Hürdeneigenschaft“ der Synchronisationsbefehle
        - » In heutigen Mikroprozessoren mit Hilfe von Spezialbefehlen implementiert
      - Die Ordnung der Speicherbefehle wird durch die sehr viel losere Ordnung der Synchronisationsbefehle ersetzt

Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Weitere Konsistenzmodelle
    - Zusammenhang



Multiprozessor mit gemeinsamem Speicher

- Speicherkonsistenzmodelle
  - Weitere Konsistenzmodelle
    - definieren theoretisch weitere Abschwächungen oder
    - leiten sich aus Hardware-Implementierungen spezieller Prozessoren oder Multiprozessorsysteme ab
      - SPARC-Prozessor-spezifisches schwaches Konsistenzmodell TSO (total store order) oder das hiervon abgeleitete PSO (patial total store)