Literatur

- Concurrency is not Parallelism
- Parallelism /= Concurrency
- Parallelism Is Not Concurrency
- Nebenläufige Programmierung mit Java : Konzepte und Programmiermodelle für Multicore-Systeme
- Simple, fast, and practical non-blocking and blocking concurrent queue algorithms
- Verteilte Systeme
- Kooperation und Konkurrenz
- Nichtsequentielle und Verteilte Programmierung mit Go

Grundlagen zu Threading

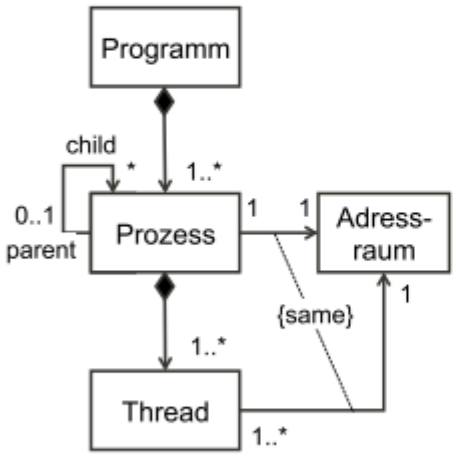
Was ist ein Thread

- „Die Leistung eines Programms wird durch das Ablaufen vieler Einzelschritte erbracht, die durch Anweisungen und Eingaben gesteuert sind. Jeden dieser Schritte bezeichnen wir als \emph{Aktivität}. Wie man Aktivitäten abgrenzt ist eine Frage der Abstraktion. Z.B. Grenze zwischen Ausführungen von Anweisungen einer höheren Programmiersprache oder auch zwischen Ausführungen von Maschinenbefehlen.
  Ein \emph{Prozess} ist eine sequentielle Folge von Aktivitäten, durch die eine in sich abgeschlossene Aufgabe bearbeitet wird [Nehmer1985]. In Abgrenzung zu anderen Prozessbegriffen nennt man einen solchen Prozess auch \emph{Rechenprozess}
- Threads aus Java-, OS- und Hardwaresicht -> Bild (S13)

PROZESSE UND THREADS

Programme, Prozesse, Threads

Begriffsmodell:



- Ein Programm besteht aus einer Menge von Anweisungen einer höheren Programmiersprache
- Prozesse entstehen bei der Ausführung von Programmen
  - Prozesse können selbst wieder Kind-Prozesse starten (mit unabhängigem Adressraum)
- Prozesse innerhalb eines gemeinsamen Adressraums heißen leichtgewichtig oder Threads
  - Alle Threads teilen sich den selben Adressraum (des zugehörigen Prozesses)

VERTEILTE SYSTEME | © PROF. DR. DANIEL JOBST | WINTERSEMESTER 2018/19 | VERSION 1.7 39

-
- Context switch ist bei Threads einfacher als bei Prozessen, da weniger zwischengespeichert werden muss.

Was ist Nebenläufigkeit

- Anweisungen werden als nebenläufig bezeichnet, wenn sie parallel oder in beliebiger Reihenfolge parallel ausgeführt werden können.
- „Zwei oder mehrere Aktivitäten (Tasks) heißen nebenläufig, wenn sie zeitgleich bearbeitet werden können.“
- Die Autoren bezeichnen Thread als Abstraktionskonzept der Nebenläufigkeit in Java.
- Die Autoren identifizieren die Zerlegung in unabhängige Teilaktivitäten und Synchronisierung bzw. Steuerung als Aufgaben der nebenläufigen Programmierung
- Programming as the composition of independently executing processes
- Concurrency is about dealing with lots of things at once.
- Concurrencys Goal is a good structure

Was ist Parallelität

- Programming as the simultaneous execution of (possibly related) computations.
- Parallelism is about doing lots of things at once.

Unterschied zwischen Nebenläufigkeit und Parallelität

- Concurrency is about structure, parallelism is about execution.
- Concurrency provides a way to structure a solution to solve a problem that may (but not necessarily) be parallelizable.

Folgen von Nebenläufigkeit

- Ausführung ist nichtdeterministisch, die Reihenfolge der Aktiviäten ist nicht definiert und kann sich bei jedem Programmdurchlauf ändern. Dabei kann das Programm weiterhin determiniert sein[1], also bei den gleichen Eingaben immer die gleichen Ausgaben liefern. Determiniertheit bei einem nichtdeterministischen Programm wird erreicht, wenn es bei jeder beliebigen Ausführreihenfolge der Aktivitäten dasselbe Ergebnis liefert.
- Nichtreproduzierbarkeit, weil Ausführung Nichtdeterministisch ist S.20
- Übernahme von Scheduling durch System.
- Race Conditions <-> Threadsicherheit
    - „Situationen, bei denen Threads auf gemeinsame Daten lesend oder schreibend zugreifen und deren Konsistenz von der Ausführreihenfolge abhängt, heißen Race Conditions (Wettkampfbedingungen).
    - „Im Allgemeinen ist das Lesen der „Readonly“-Daten (immutable shared data) unkritisch.
    - Daten sind Threadsicher, wenn sie von anderen Objekten immer in eineim gültigen Zustand gesehen werden
- Notwendigkeit der Koordinierung (Synchronisierung) von Zugriffen auf geteilte Ressourcen

Synchronisierung

- „to algorithms for concurrent data structures, including FIFO queues, fall into two categories:blockingandnon-blocking.“
- Einschränkung der zeitlichen Beliebigkeit und Abstimmung konkurrierender Zugriffe

Blockende Synchronisierung

- Blocking algorithms allow a slow or delayed process to prevent faster processes from completingoperations on the shared data structure indefinitely.

---

[1] Es kann durchaus sein, dass Determiniertheit in einem Programm nicht gewünscht ist. Man betrachte beispielsweise ein Programm, das einen echten Zufallsgenerator beschreibt. Hier wäre Determiniertheit ein direkter Widerspruch zur Aufgabe des Programms.

Nicht Blockende Synch

- Zb. Synch durch Nutzung atomarer Operationen?
- „Non-blocking algorithms guarantee that if there are one or more active processes trying to perform operations on a shareddata structure, some operation will complete within a finite number of time steps“
- Compare_and_swap
- wait-free Algorithm

Vergleich zwischen Block und nichtblock

- „On asynchronous (especially multi-programmed) multiprocessor systems, blocking algorithmssuffer significant performance degradation when a processis halted or delayed at an inopportune moment. Possible sources of delay include processor scheduling preemption,page faults, and cache misses. Non-blocking algorithmsare more robust in the face of these events.“