

Script generated by TTT

Title: Grundlagen_Betriebssysteme (31.10.2012)

Date: Wed Oct 31 13:16:34 CET 2012

Duration: 43:25 min

Pages: 12

Grundlagen: Betriebssysteme und Systemsoftware (GBS)

- Prof. J. Schlichter
 - Lehrstuhl für Angewandte Informatik / Kooperative Systeme, Fakultät für Informatik, TU München
 - Boltzmannstr. 3, 85748 Garching
 - Email: schlichter@in.tum.de
 - Tel.: 089-289 18654
 - URL: <http://www11.informatik.tu-muenchen.de/>

[Übersicht](#)
[Einführung](#)
[Parallele Systeme - Modellierung, Strukturen](#)
[Prozess- und Prozessorverwaltung](#)
[Speicherverwaltung](#)
[Prozesskommunikation](#)
[Dateisysteme](#)
[Ein-/Ausgabe](#)
[Sicherheit in Rechensystemen](#)
[Entwurf von Betriebssystemen](#)
[Zusammenfassung](#)

Generated by Targeteam



In einem allgemeinen Rechensystem finden eine Vielzahl paralleler Abläufe statt. Dieser Abschnitt beschäftigt sich mit den Eigenschaften paralleler Systeme.

[Fragestellungen](#)

[Grundlagen](#)

[Modellierung paralleler Systeme](#)

[Thread-Konzept](#)

[Synchronisation](#)

[Verklemmungen](#)

Generated by Targeteam



Sequentielle Aspekte von systemnaher Programmierung:

Programm als Repräsentation eines Algorithmus;

Programm als **sequentielle** Folge von Aktionen/Anweisungen (alle Schritte nacheinander);

determinierte Programme: unter gleichen Bedingungen und Eingaben produziert das Programm stets das gleiche Ergebnis;

deterministische Abläufe: eindeutig vorbestimmter Ablauf, keine willkürliche Auswahl von alternativen Schritten.

Übergang von sequentiellen Systemen hin zu parallelen Systemen/Programmen. Dabei werden die folgenden Aspekte näher betrachtet.

gleichzeitige Aktivität von Komponenten, die möglicherweise miteinander kommunizieren oder aufeinander einwirken.

Beispiele für Parallelität

HW-Komponenten eines Rechensystems, z.B. Prozessor und E/A-Controller.

SW-Komponenten eines Rechensystems, z.B. parallel ablaufende Programme; u.U. Zugriff auf gemeinsame Ressourcen, z.B. Daten im Arbeitsspeicher.

SW-Komponenten in einem verteilten System, d.h. Rechensysteme, die über ein Rechnernetz miteinander verbunden sind.

[Aspekte des Abschnitts](#)

Generated by Targeteam



Modellierungstechniken zur Analyse und Beschreibung der Eigenschaften paralleler Systeme (Spuren, Petrinetze).

Threads als Mittel zur Realisierung von Parallelität innerhalb eines Prozesses.

Synchronisation in parallelen Systemen.

Mechanismen zur Behandlung von Verklemmungen.

Generated by Targeteam



In diesem Teilabschnitt werden kurz die wichtigsten Begriffe definiert sowie Konzepte zur Formulierung paralleler Aktivitäten aufgelistet. Einige dieser Konzepte werden in nachfolgenden Teilabschnitten ausführlicher diskutiert.

[Begriffsdefinitionen](#)

[Beschreibungskonzepte](#)

Generated by Targeteam



[Nebenläufigkeit](#)

Parallelität

Parallelität wird häufig synonym zu Nebenläufigkeit verwendet;

spezielle Form der Nebenläufigkeit: mehrere Prozessoren stehen zur Ausführung zur Verfügung.

Verteiltheit

Verteiltheit (engl. distribution) ist die räumliche oder auch nur konzeptionelle Aufteilung der Komponenten eines Systems, z.B. vernetzte PCs.

[Interaktion](#)

[Nichtdeterminismus](#)

Generated by Targeteam



Nebenläufigkeit (engl. concurrency) bezieht sich auf die zeitliche Beziehung zwischen den Aktivitäten von Komponenten, die gleichzeitig oder zeitlich verzahnt (engl. interleaving) ablaufen können.

z.B. gleichzeitig: Drucken eines Dokuments und Berechnen einer Formel in einer Tabellenkalkulation (CPU-Nutzung).

z.B. zeitlich verzahnt: Benutzerauftrag muss CPU abgeben und wartet; ein anderer Auftrag wird (teilweise) ausgeführt und gibt CPU wieder zurück, bevor das Ende erreicht ist.

Generated by Targeteam



Nebenläufigkeit

Parallelität

Parallelität wird häufig synonym zu Nebenläufigkeit verwendet;
 spezielle Form der Nebenläufigkeit: mehrere Prozessoren stehen zur Ausführung zur Verfügung.

Verteiltheit

Verteiltheit (engl. distribution) ist die räumliche oder auch nur konzeptionelle Aufteilung der Komponenten eines Systems, z.B. vernetzte PCs.

Interaktion

Nichtdeterminismus

Generated by Targeteam



Kommunikation, Synchronisation, Konkurrenz betrifft die kausalen Beziehungen zwischen räumlich verteilten und nebenläufig ausgeführten Aktivitäten.

kausale Abhängigkeit zwischen Ampel und Fußgänger:

Ereignis von Fußgänger:

er drückt Knopf

Ereignis beeinflusst Ampel: sie schaltet auf Rot für Auto

Kommunikation: Prozesse auf dem gleichen oder unterschiedlichen Rechensystemen tauschen Nachrichten untereinander aus.

Koordinierung: Beziehung zwischen Auftraggeber und Auftragnehmer \Rightarrow Erstellen eines Auftrages durch Client und anschließendes Bearbeiten des Auftrags durch Server.

Konkurrenz: Aktivitäten eines Prozesses behindern die eines anderen Prozesses z.B. Warten, wenn CPU von anderem Prozess belegt.

Generated by Targeteam



System zeigt bei gleichen Ausgangsbedingungen und gleichen Eingaben unterschiedliches Verhalten (Reihenfolge der Prozesse spielt eine Rolle!). Problem: Nichtreproduzierbarkeit von Ergebnissen, Testen von parallelen Programmen bei Nichtdeterminismus ist schwierig. Beispiel:

$x = 2$ Prozess 1: $x = x+5$ Prozess 2: $x = x*2$

Ausführungsreihenfolge

P1 vor P2: $x == 14$

P2 vor P1: $x == 9$

Generated by Targeteam



Beschreibungskonzepte



Es gibt eine Vielzahl von Konzepten zur Formulierung paralleler Aktivitäten.

modell-basierte Konzepte

ereignis-orientiert: es wird der Ablauf von Prozessen betrachtet; jeder Ablauf wird durch eine Menge von Ereignissen charakterisiert. Jeder Ablauf repräsentiert eine **Ereignisspur**. Ereignisse können in kausalem Zusammenhang stehen.

graphisch-orientiert: der Prozessablauf wird graphisch dargestellt. Petrinetze sind ein Beispiel dafür.

Sprachkonstrukte in Programmiersprachen

parallele Komposition von Teilabläufen innerhalb eines Prozesses, z.B. Java Threads.

Prozesshierarchien in Betriebssystemen: u.a. fork, join

Kommunikationskonzepte: u.a. send, receive

Synchronisationskonzepte: u.a. lock, unlock

Konzepte in Betriebssystemen

Prozesskonzept = Programm in Ausführung;

Threadkonzept = leichtgewichtiger Prozess;

Kommunikation: Shared memory, Dateien, Nachrichten etc.

Synchronisation: Unterbrechungen (Interrupts), Sperren etc.

Generated by Targeteam



In diesem Teilabschnitt werden kurz die wichtigsten Begriffe definiert sowie Konzepte zur Formulierung paralleler Aktivitäten aufgelistet. Einige dieser Konzepte werden in nachfolgenden Teilabschnitten ausführlicher diskutiert.

[Begriffsdefinitionen](#)

[Beschreibungskonzepte](#)