

Script generated by TTT

Title: Grundlagen_Betriebssysteme (09.01.2013)

Date: Wed Jan 09 13:20:40 CET 2013

Duration: 40:17 min

Pages: 14



Disjunkte Prozesse, d.h. Prozesse, die völlig isoliert voneinander ablaufen, stellen eher die Ausnahme dar. Häufig finden Wechselwirkungen zwischen den Prozessen statt \Rightarrow Prozesse interagieren. Die Unterstützung der Prozessinteraktion stellt einen unverzichtbaren Dienst dar.

Fragestellungen

Dieser Abschnitt beschäftigt sich mit den Mechanismen von Rechensystemen zum Austausch von Informationen zwischen Prozessen.

Kommunikationsarten.

nachrichtenbasierte Kommunikation, insbesondere Client-Server-Modell.

Netzwerkprogrammierung auf der Basis von Ports und Sockets.

[Einführung](#)

[Nachrichtenbasierte Kommunikation](#)

[Client-Server-Modell](#)

[Netzwerkprogrammierung](#)

Generated by Targeteam



Prozessinteraktion kann Rechner-lokal und Rechner-übergreifend stattfinden. Prozesse können auf vielfältige Weise Informationen austauschen.

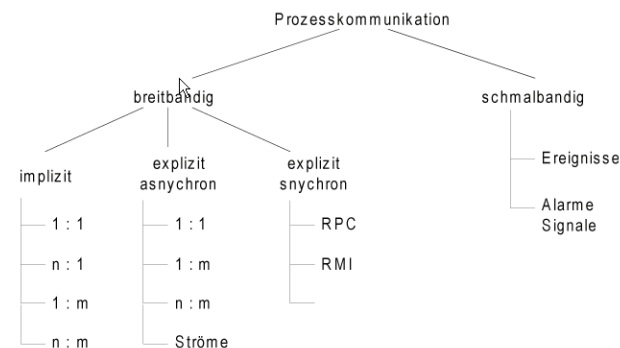
[Kommunikationsarten](#)

Verteilte Systeme

Bisher haben wir uns mit systemnahen Konzepten von zentralen Systemen beschäftigt. Seit Ende der 80er Jahre haben jedoch verteilte Systeme rapide an Bedeutung gewonnen.

[Ansätze zur Kopplung von Recheneinheiten](#)

Generated by Targeteam



Die Bandbreite des Kommunikationskanals bestimmt die Datenrate, in der Daten zwischen Prozessen ausgetauscht werden können.

[Schmalbandige Kanäle](#)

[Implizite Kommunikation](#)

[Explizite Kommunikation](#)

Generated by Targeteam



Schmalbandige Kanäle



Schmalbandige Kanäle werden im Betriebssystem zum Melden von Ereignissen oder für die Synchronisation unterstützt. Übertragung von wenigen Bits an Information, z.B. Setzen von Flags

Dienste des Betriebssystems

- Melden von Ereignissen,
- Warten auf Ereignisse,
- Ereignisverwaltung.

Beim Ablauf von Prozessen können Alarmer entstehen (z.B. arithmetische Alarmer).

Die Alarmer werden über Namen identifiziert, die im BS vereinbart sind. Das BS stellt Dienste zur Zustellung von Alarmen zur Verfügung.

Generated by Targeteam



Implizite Kommunikation



Implizite Kommunikation ist eine breitbandige Kommunikationsform. Die Kommunikation erfolgt über einen gemeinsamen Speicher (Dateien, Register, Datenstrukturen).

Die Kommunikation findet ohne direkte Unterstützung und ohne Kenntnis des BS statt.

Vorteil: einfach und schnell (kein Kopieren zwischen Adressräumen).

Nachteil:

- gemeinsame Bereiche sind nicht immer vorhanden: z.B. in physisch verteilten, vernetzten Systemen gibt es i.d.R. keinen gemeinsamen Speicher.
- gegebenenfalls aufwendiges busy waiting \Rightarrow Mischform: Ereigniszustellung, d.h. schmalbandige Kommunikation, die das Vorhandensein von Daten signalisiert.

Implizite Kommunikationsformen

Generated by Targeteam



Verschiedene Formen der impliziten Kommunikation

1:1 ein Puffer pro Sender/Empfänger-Paar

n:1 n Sender senden Nachrichten an einen Empfänger, z.B.

Sender: Prozesse senden Druckaufträge

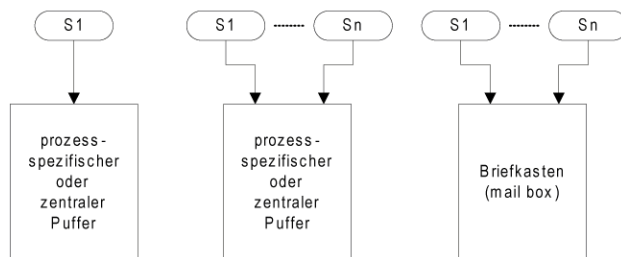
Empfänger: Drucker-Server

1:m Mitteilung an alle Prozesse (Broadcast, Multicast);

Broadcast : z.B. Erfragen, wo ein spezieller Dienst angeboten wird; Shutdown Message an alle Rechner.

Multicast : z.B. Nachricht an Gruppe gleichartiger Server.

n:m n Erzeuger schreiben in Puffer und m Verbraucher lesen aus Puffer.



Verschiedene Formen der impliziten Kommunikation

1:1 ein Puffer pro Sender/Empfänger-Paar

n:1 n Sender senden Nachrichten an einen Empfänger, z.B.

Sender: Prozesse senden Druckaufträge

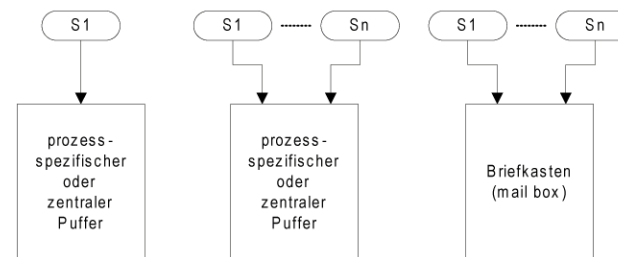
Empfänger: Drucker-Server

1:m Mitteilung an alle Prozesse (Broadcast, Multicast);

Broadcast : z.B. Erfragen, wo ein spezieller Dienst angeboten wird; Shutdown Message an alle Rechner.

Multicast : z.B. Nachricht an Gruppe gleichartiger Server.

n:m n Erzeuger schreiben in Puffer und m Verbraucher lesen aus Puffer.





Explizite Kommunikation

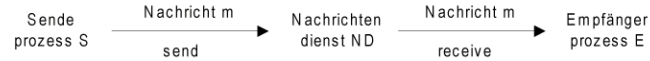


Diese Kommunikationsart wird realisiert durch den Austausch von Nachrichten ("message passing") => **nachrichtenbasierte Kommunikation**.

Betriebssystem enthält einen **Nachrichtendienst ND** (das Kommunikationssystem), der den Austausch der Nachrichten zwischen Prozessen realisiert. ND unterstützt 2 Systemdienste:

```
send (E: process, m: message)
receive (S: process, m: message)
```

prinzipieller Ablauf



Aufbau einer Nachricht

Eine Nachricht besteht aus 2 grundlegenden Komponenten:

- Nachrichtenkopf: Verkehrsinformation, z.B. Sender- und Empfängeridentifikation
- Nachrichteninhalt: Nutzlast (payload)

explizite Kommunikation ist besonders geeignet in verteilten, vernetzten Systemen.

Generated by Targeteam



Einführung



Prozessinteraktion kann Rechner-lokal und Rechner-übergreifend stattfinden. Prozesse können auf vielfältige Weise Informationen austauschen.

Kommunikationsarten

Verteilte Systeme

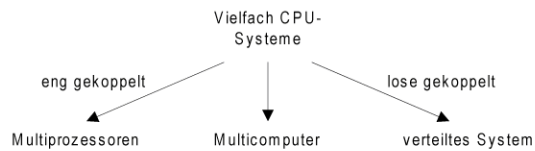
Bisher haben wir uns mit systemnahen Konzepten von zentralen Systemen beschäftigt. Seit Ende der 80er Jahre haben jedoch verteilte Systeme rapide an Bedeutung gewonnen.

[Ansätze zur Kopplung von Recheneinheiten](#)

Generated by Targeteam



Ansätze zur Kopplung von Recheneinheiten



	Multiprozessor	Multicomputer	verteiltes System
Konfiguration	nur CPU	CPU, Arbeitsspeicher, Netzchnittstelle	kompletter Rechner
Peripherie	gemeinsam	gemeinsam, außer Festplatte	getrennt
Ort	gemeinsames Gehäuse	gemeinsamer Raum	weltweit verteilt
Kommunikation	gemeinsamer Arbeitsspeicher	direkte Verbindung	Netzwerk
Betriebssystem	ein gemeinsames BS	getrennt, jedoch gleicher Typ	getrennt, vielfach unterschiedlich
Verwaltung	eine Organisation	eine Organisation	viele Organisationen

Generated by Targeteam



Nachrichtenbasierte Kommunikation



Bei nachrichtenbasierter Prozessinteraktion tauschen Prozesse gezielt Informationen durch Verschicken und Empfangen von Nachrichten aus; ein Kommunikationssystem unterstützt an der Schnittstelle wenigstens die Funktionen **send** und **receive**.

[Elementare Kommunikationsmodelle](#)

[Erzeuger-Verbraucher Problem](#)

[Modellierung durch ein Petrinetz](#)

[Ports](#)

[Kanäle](#)

[Ströme](#)

[Pipes](#)

Generated by Targeteam



Klassifikationsschema



Elementare Kommunikationsmuster sind Meldung ("signal") und Auftrag ("request").

Meldung : Einweg Nachricht vom Sender zum Empfänger (unidirektional).

Auftrag : Zweiweg Nachricht zwischen Sender und Empfänger (bidirektional). Sie beginnt mit dem Versenden eines Auftrags an den Empfänger und endet mit der Übergabe einer Erfolgsbestätigung über den durchgeführten Auftrag an den Sender.

Synchronität definiert den Kopplungsgrad zwischen Prozessen bei der Durchführung einer Nachrichtentransaktion:

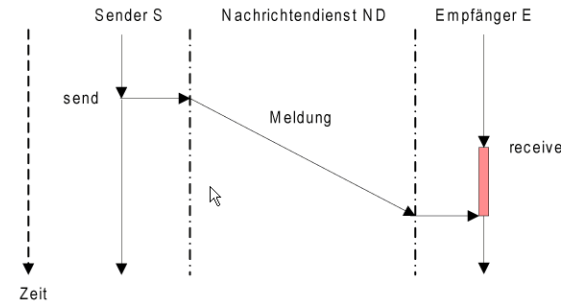
asynchron : Entkopplung des Senders und Empfängers.

synchron : beide Prozesse werden zur Nachrichtenübertragung synchronisiert.

Generated by Targeteam



Sender wird lediglich bis zur Ablieferung der Meldung an das Nachrichtensystem (Kommunikationssystem) blockiert.



Nachrichtendienst des Betriebssystems puffert Nachricht;

Sender S kann seine Ausführung fortsetzen, sobald Nachricht N in den Nachrichtenpuffer des ND eingetragen ist.

S wartet **nicht**, bis E die Nachricht empfangen hat.

Empfänger E zeigt durch receive an, dass er am Empfang der Nachricht N interessiert ist.

Empfänger wird blockiert, bis Sender Nachricht bereit stellt.

Generated by Targeteam



Elementare Kommunikationsmodelle



Klassifikationsschema für die Nachrichtenkommunikation anhand von 2 Dimensionen:

generelles Muster der Nachrichtenkommunikation.

zeitliche Kopplung der beteiligten Prozesse.

Klassifikationsschema

Meldung

[Asynchrone Meldung](#)

[Synchrone Meldung](#)

Auftrag

[Synchroner Auftrag](#)

[Asynchroner Auftrag](#)

[Vorteile/Nachteile asynchrones Senden](#)

Generated by Targeteam