

## Script generated by TTT

Title: Eingebettete\_Systeme (13.11.2012)

Date: Tue Nov 13 14:49:13 CET 2012

Duration: 66:16 min

Pages: 21

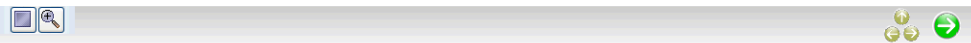
Eingebettete vernetzte Systeme (IN8014)

Teil I: Betriebssysteme und Systemsoftware

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

[Übersicht](#)  
[Einführung](#)  
[Synchronisation und Verklemmungen](#)  
[Prozess- und Prozessorverwaltung](#)  
[Speicherverwaltung](#)  
[Prozesskommunikation](#)  
[Zusammenfassung](#)

Generated by Targeteam



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.

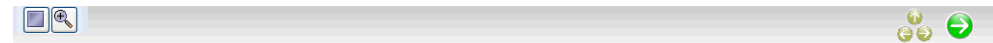
Netzwerkprogrammierung auf der Basis von Ports und Sockets.

### Einführung

[Nachrichtenbasierte Kommunikation](#)

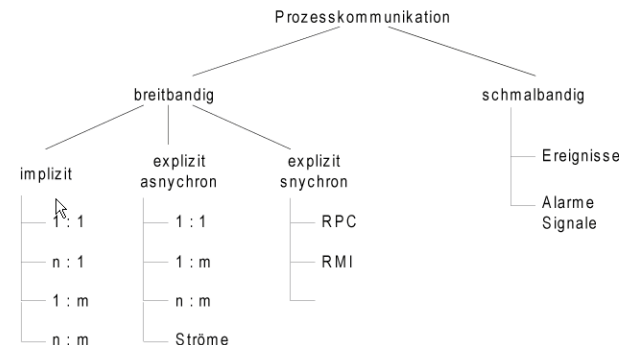
[Client-Server-Modell](#)

[Netzwerkprogrammierung](#)



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

### Kommunikationsarten



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 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 Alarme entstehen (z.B. arithmetische Alarme).

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

Generated by Targeteam



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:

- a) gemeinsame Bereiche sind nicht immer vorhanden: z.B. in physisch verteilten, vernetzten Systemen gibt es i.d.R. keinen gemeinsamen Speicher.
- b) 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.

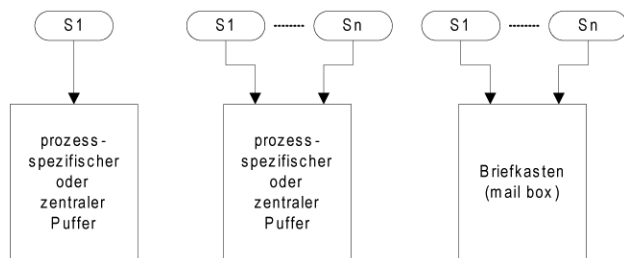
Sender: Prozesse senden Druckaufträge  
Empfänger: Drucker-Server

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.



### Implizite Kommunikationsformen



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

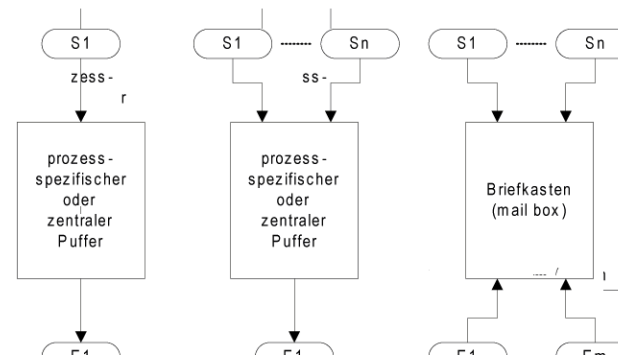
Sender: Prozesse senden Druckaufträge  
Empfänger: Drucker-Server

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.



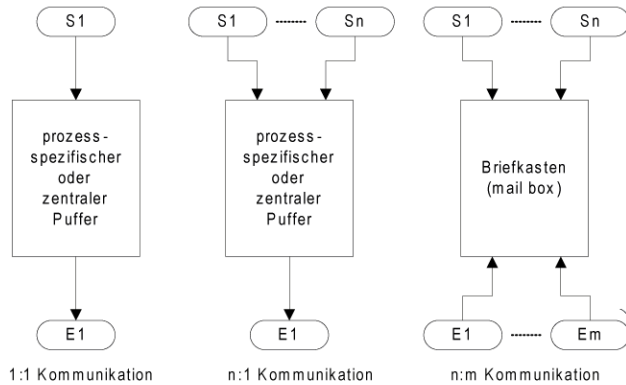
Empfänger: Drucker-Server

1. 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 Erzeuger schreiben in Puffer und m Verbraucher lesen aus Puffer.



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:

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

## Implizite Kommunikationsformen

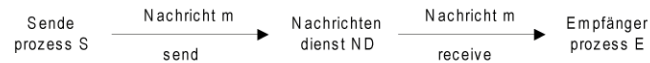
Generated by Targeteam

Diese Kommunikationsart wird realisiert durch den Austausch von Nachrichten ("message passing")  $\Rightarrow$  **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

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

Generated by Targeteam



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:

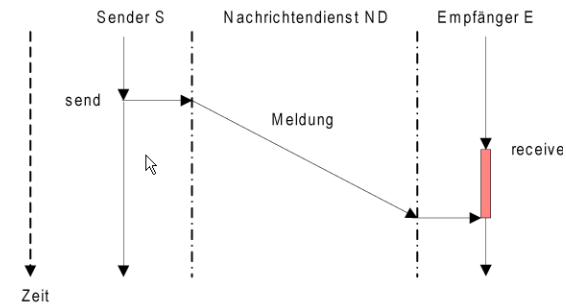
**asynchron** : Entkopplung des Senders und Empfängers.

**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.

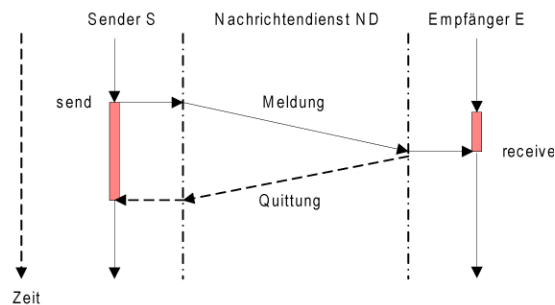
Generated by Targeteam



## Synchrone Meldung



Sender und Empfänger von Meldungen sind zeitlich gekoppelt.



**Rendezvous** -Technik: Sender und Empfänger stellen vor Austausch der eigentlichen Meldung die Sende- und Empfangsbereitschaft her.

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](#)

[Asynchrone Auftrag](#)

[Vorteile/Nachteile asynchrones Senden](#)

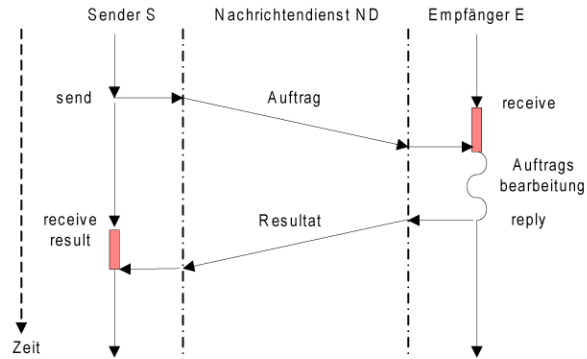
Generated by Targeteam



## Asynchroner Auftrag



Auftrag und Resultat werden als Paar unabhängiger Meldungen verschickt.



Generated by Targeteam



## Vorteile/Nachteile asynchrones Senden



### Vorteile asynchrones Senden

- nützlich für Realzeitanwendungen.
- ermöglicht parallele Abarbeitung durch Sender und Empfänger.
- anwendbar zum Signalisieren von Ereignissen.

### Nachteile asynchrones Senden

- Verwaltung des Nachrichtenpuffers durch BS erforderlich.
- Benachrichtigung des Senders S im Fehlerfall und Behandlung von Fehlern ist problematisch.
- Entwurf und Nachweis der Korrektheit des Systems ist schwierig.

Generated by Targeteam



Klassifikationsschema für die Nachrichtenkommunikation anhand von 2 Dimensionen:

generelles Muster der Nachrichtenkommunikation.

zeitliche Kopplung der beteiligten Prozesse.

### Klassifikationsschema

#### Meldung

[Asynchrone Meldung](#)

[Synchrone Meldung](#)

#### Auftrag

[Synchrone Auftrag](#)

[Asynchrone Auftrag](#)

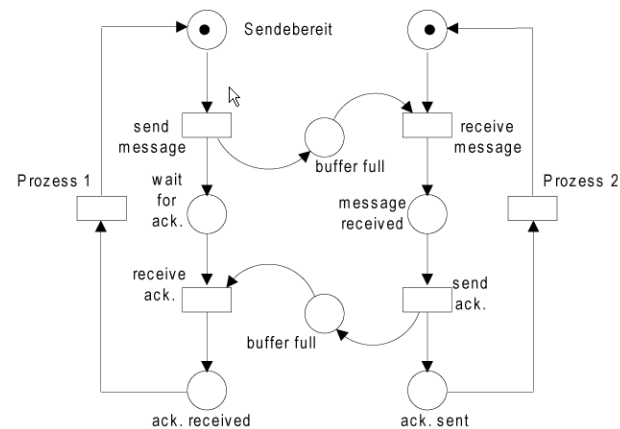
[Vorteile/Nachteile asynchrones Senden](#)



Generated by Targeteam



Petri-Netze dienen häufig zur Modellierung von Kommunikationsabläufen, sogenannten Kommunikationsprotokollen. Sie ermöglichen die Analyse der Protokolle, z.B. Erkennung von Verklemmungen. Modellierung einer synchronen Kommunikation:



### Problem: unendliches Warten

Pragmatische Lösung mit Hilfe von Timeouts

Sender bzw. Empfänger warten nur eine festgelegte Zeit, Sender: falls kein Acknowledgement (Quittung) eintrifft, z.B. erneutes Senden.

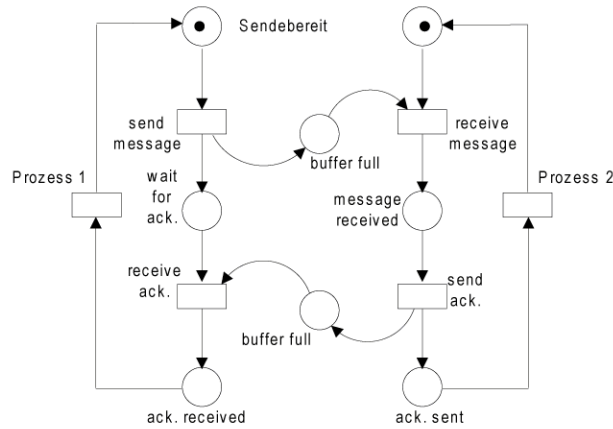
Probleme dabei? u.a. Duplikate müssen vom Empfänger erkannt werden: resendete Nachrichten



Generated by Targeteam



Modellierung einer synchronen Kommunikation.



### Problem: unendliches Warten

Pragmatische Lösung mit Hilfe von Timeouts

Sender bzw. Empfänger warten nur eine festgelegte Zeit, Sender: falls kein Acknowledgement (Quittung) eintrifft, z.B. erneutes Senden.

Probleme dabei? u.a. Duplikate müssen vom Empfänger erkannt werden; gesendete Nachrichten kommen zu spät an, sind veraltet etc.

Generated by Targem

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](#)

Generated by Targem