

Systemprogrammierung

Betriebsarten: Postskriptum

Wolfgang Schröder-Preikschat

Lehrstuhl Informatik 4

Gliederung

1 Netzbetrieb

- Offene Systeme
- Verteiltes System
- Rechnernetze

2 Integrationsbetrieb

- Eingebettete Systeme
- Drahtlose Sensornetze

3 Zusammenfassung

Systeme mit offenen Schnittstellen und Spezifikationen

Programme bekommen Zugriff auf Betriebsmittel eines Rechnerverbunds

- das Netzwerk ist in verschiedener Weise „transparent“
 - **Netzwerktransparenz** (Zugriffs- und Ortstransparenz)
 - Replikations-, Migrations-, Fehler-, Ausfall-, Leistungs-, Skalierungs-, Nebenläufigkeits-, . . . , X-transparenz
- ferne Betriebsmittel werden über lokale Repräsentanten virtualisiert

Programmverarbeitung geschieht (ein Stück weit) verteilt

- verteilte Kompilierung, verteiltes `make(1)`, `ftp(1)`, `rsh(1)`
- Betriebssystemkerne enthalten Kommunikationsprotokolle (TCP/IP)

Problem

- Heterogenität, netzwerkzentrische Betriebsmittelverwaltung

Prozessgrenzen überwindende Unterprogrammaufrufe

Prozedurfernaufruf (engl. *remote procedure call*[5], RPC)

- liefert die Illusion des lokalen Zugriffs auf entfernte Prozeduren

Stümpfe (engl. *stubs*) **virtualisieren Aufgerufenen/Aufrufer^a**

^aDie Art eines Stumpfes bezeichnet seine Lage, nicht wovon er abstrahiert!

- beim Aufrufer (Klient): repräsentiert Aufgerufenen (Anbieter)
 - Klientenstumpf (engl. *client stub*)
- beim Aufgerufenen (Anbieter): repräsentiert Aufrufer (Klient)
 - Anbieterstumpf (engl. *server stub*)

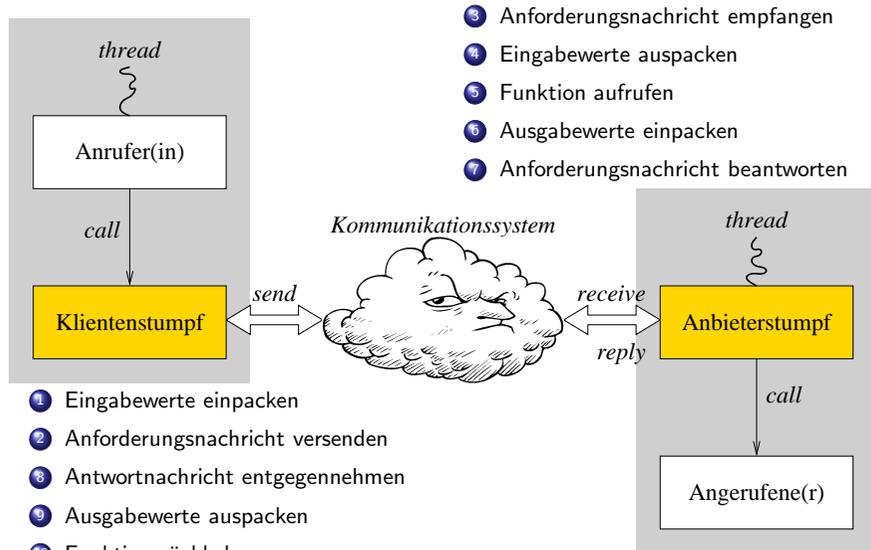
Klient/Anbieter \mapsto verschiedene Fäden, Adressräume und/oder Rechner

- **Interprozesskommunikation** (engl. *inter-process communication*, IPC)

Problem

- Parameterübergabe, Serialisierung von Datenstrukturen, Fehlermodell

Fernaufrufe mittels Stellvertreterfunktionen



Verteilung wird (funktional) nicht wahrgenommen

Verteiltes Betriebssystem...

- A distributed operating system is one that looks to its users like an ordinary centralized operating system but runs on multiple, independent central processing units (CPUs).
- The key concept here is transparency.
- In other words, the use of multiple processors should be invisible (transparent) to the users.
- Another way of expressing the same idea is to say that the user views the system as a "virtual uniprocessor", not as a collection of distinct machines.

[6]

Systeme von interagierenden Prozessen (Prozessoren)

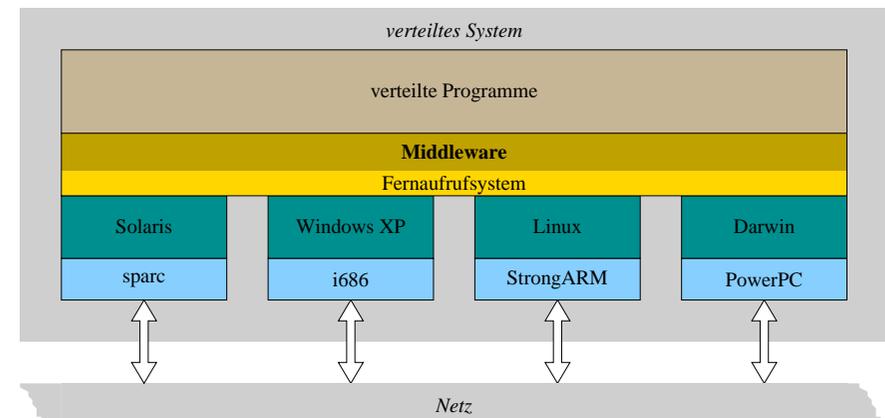
A distributed computing system consists of multiple autonomous processors that do not share primary memory, but cooperate by sending messages over a communication network. [1]

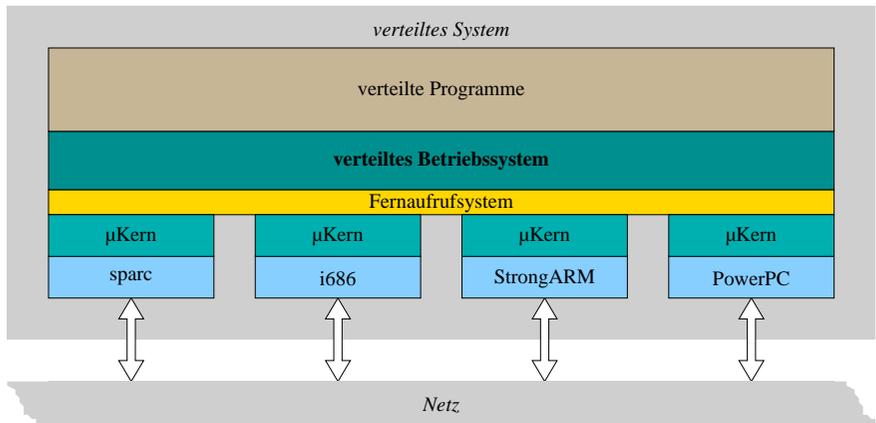
A collection of independent computers that appears to its users as a single coherent system. [7]

A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable. [4]

Diensteschicht (engl. *middleware*)

Als einzelnes System präsentierter Zusammenschluss vernetzter Rechner





- LAN** (engl. *local area network*)
 - typisch für die Vernetzung von Arbeitsplatzrechnern
 - meist ein **homogenes System**: dieselben Rechner/Betriebssysteme
- MAN** (engl. *metropolitan area network*)
 - typisch für die Vernetzung von Großrechnern und LANs
 - **heterogenes System**: verschiedene Rechner/Betriebssysteme
- WAN** (engl. *wide area network*)
 - typisch für die Vernetzung von LANs und WANs: **Internet**
 - **heterogenes System**: verschiedene Rechner/Betriebssysteme

Problem

- Skalierbarkeit, Echtzeitfähigkeit

- CAN** (engl. *control area network*)
 - typisch für die Vernetzung elektronisch gesteuerter Maschinen
 - Netzknoten sind Steuergeräte (engl. *electronic control units, ECU*)
 - je nach Anwendungsfall ein **heterogenes System**, z.B. KFZ



(Quelle: Bosch — Antriebsstrangnetz, Motorsteuerungsgerät und -anschlüsse)

- 1 Netzbetrieb**
 - Offene Systeme
 - Verteiltes System
 - Rechnernetze
- 2 Integrationsbetrieb**
 - Eingebettete Systeme
 - Drahtlose Sensornetze
- 3 Zusammenfassung**

Spezialbetriebssystem (engl. *special purpose operating system*)

Betriebssystem und Anwendungsprogramm(e) sind mit- bzw. ineinander verwoben, sie bilden eine (in sich geschlossene) Einheit:

im Sinne der Funktionalität (obligatorisch)

- das Betriebssystem ist maßgeschneidert und anwendungsgewahr

im Sinne der Repräsentation (optional)

- das Betriebssystem liegt in Form einer (Quelltext-) Bibliothek vor

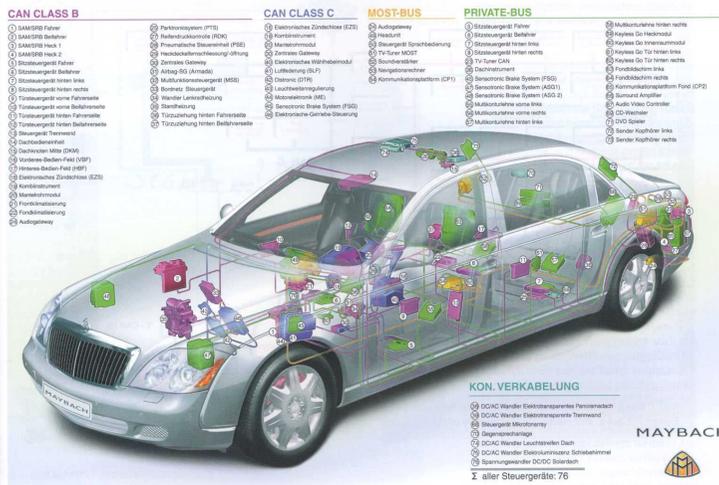
Eingebettetes System (engl. *embedded system*)

Jedes in einem Produkt versteckte Rechensystem, wobei das Produkt selbst jedoch kein Rechner ist: Kühlschrank, Mikrowelle, Kochplatte, Backofen, Esse, Wasserkocher, Waschmaschine, ...

Wenn Kompromisslösungen impraktikabel sind



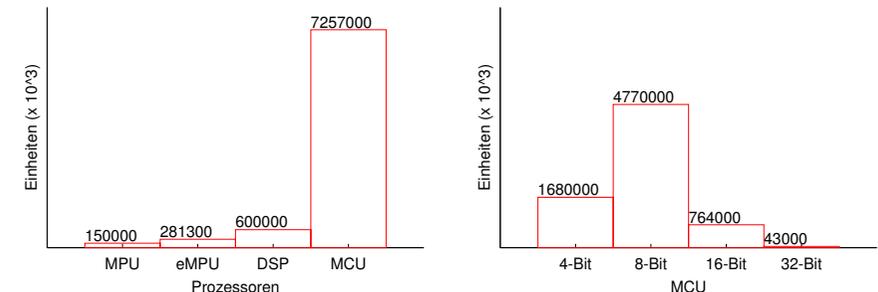
Verteiltes System auf Rädern



(Quelle: DaimlerChrysler [3])

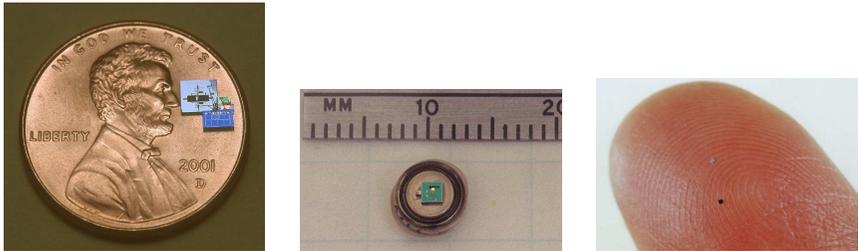
Y2K Prozessorproduktion

„Where have all the processors gone?“



über 8 Mrd. Prozessoren [8]

1.8 % (MPU) Server, Desk-/Laptops, ...
98.2 % (eMPU, DSP, MCU) eingebettete Systeme



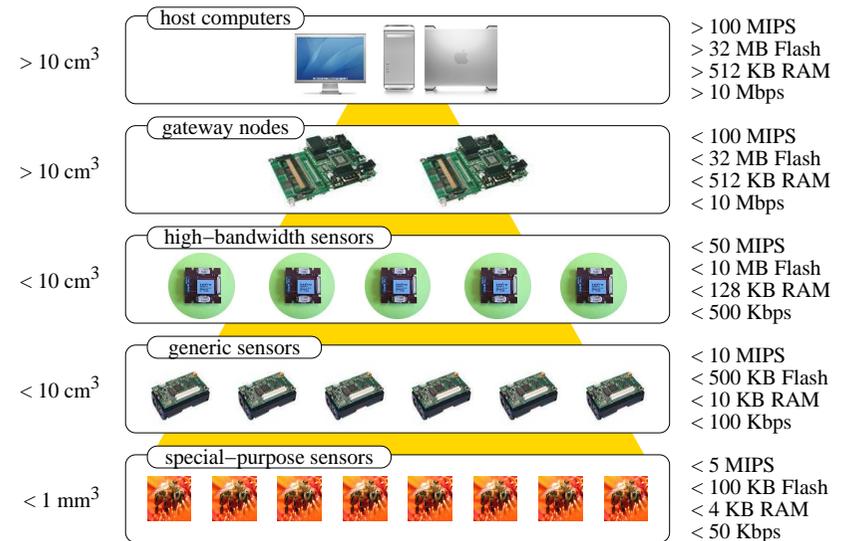
μ Controller von Sand-/Staubkorngröße, die über Radiofrequenztechnik miteinander kommunizieren [2]:

- jeder einzelne Kleinstrechner bildet einen kubischen Sensor (*mote*)
- u. A. gedacht zur Überwachung menschenfeindlicher Umgebungen

$\left\{ \begin{array}{l} \text{[verteiltes]} \text{ grid} \\ \text{[durchdringendes]} \text{ pervasive} \\ \text{[allgegenwärtiges]} \text{ ubiquitous} \end{array} \right\} \text{ computing} \Rightarrow \text{ambient intelligence}$

- nahezu jedes „Gerät“ ist mit Kleinstrechnern (Sensoren, Aktoren) bestückt, die die unbegrenzte globale Vernetzung ermöglichen
- die Gerätenetze sind so in die Umgebung eingebettet, dass ihre Konnektivität jederzeit verfügbar und höchst unaufdringlich ist

Fiktion? Ja, noch ...



{BlueCat, HardHat} Linux, Embedix, Windows {CE, NT Embedded}, ...

Problem

- Skalierbarkeit, Betriebsmittelbedarf (insb. RAM und Energie)

..., BOSS, C{51, 166, 251}, CMX RTOS, C-Smart/Raven, eCos, eRTOS, Embos, Ercos, Euros Plus, Hi Ross, Hynet-OS, ITRON, LynxOS, MicroX/OS-II, Nucleus, OS-9, OSE, OSEK {Flex, Plus, Turbo, time}, Precise/{MQX, RTCS}, proOSEK, pSOS, PURE, PXROS, QNX, Realos, RTMOSxx, Real Time Architect, RTA, RTOS-UH, RTX{51, 166, 251}, RTXC, Softune, SSXS RTOS, ThreadX, TinyOS, VRTX, VxWorks, ...

Der Anteil proprietärer Lösungen liegt bei über 50% [9]

- in vielen Fällen wird das Rad neu erfunden...
- Softwaretechnik \rightsquigarrow Wiederverwendbarkeit und Variantenverwaltung

Gliederung

1 Netzbetrieb

- Offene Systeme
- Verteiltes System
- Rechnernetze

2 Integrationsbetrieb

- Eingebettete Systeme
- Drahtlose Sensornetze

3 Zusammenfassung

Literaturverzeichnis

- [1] BAL, H. E.:
Programming Distributed Systems.
Prentice Hall International, 1991. –
ISBN 0-13-722083-9
- [2] CULLER, D. E. ; HONG, W. :
Wireless Sensor Networks — Introduction.
In: *Communications of the ACM* 47 (2004), Jun., Nr. 6, S. 30–33
- [3] DAIMLERCHRYSLER AG:
Der neue Maybach.
In: *ATZ/MTZ Sonderheft* (2002), Sept., S. 125
- [4] LAMPORT, L. :
distribution.
<http://research.microsoft.com/en-us/um/people/lamport/pubs/distributed-system.tex>
Mai 1987
- [5] NELSON, B. J.:
Remote Procedure Call.
Pittsburg, PA, USA, Department of Computer Science, Carnegie-Mellon University, Diss.,
Mai 1981

Resümee

- **Netzbetrieb** räumlich voneinander getrennte Rechensysteme
 - Interoperation wird erreicht durch offene Schnittstellen/Spezifikationen
 - Rechnernetz und Dienst-/Funktionsverteilung sind „transparent“
 - ferne Betriebsmittel werden über lokale Repräsentanten virtualisiert
- **Integrationsbetrieb** (tiefst) eingebetteter Rechensysteme
 - Betriebssystem und Anwendungsprogramm bilden eine Einheit
 - bzgl. Funktionalität (obligatorisch) und Repräsentation (optional)
 - Spezialbetriebssysteme für echtzeitabhängige Anwendungsprogramme

Over the past 40 years, computer science have addressed only about 2% of the world's computing requirements. It's time to get physical, get real, and get out to build proactive^a systems. [8]

^a[pro-ac-tive] Acting in advance to deal with an expected difficulty; anticipatory

Literaturverzeichnis (Forts.)

- [6] TANENBAUM, A. S. ; VAN RENESSE, R. :
Distributed Operating Systems.
In: *ACM Computing Surveys* 17 (1985), Dez., Nr. 4, S. 419–470
- [7] TANENBAUM, A. S. ; VAN STEEN, M. :
Distributed Systems: Principles and Paradigms.
Prentice Hall, 2002. –
ISBN 0-130-88893-1
- [8] TENNENHOUSE, D. :
Proactive Computing.
In: *Communications of the ACM* 43 (2000), Mai, Nr. 5, S. 43–50
- [9] WALLS, C. :
The Perfect RTOS.
In: *Proceedings of the embedded world 2004, 17.–19. Februar 2004, Nürnberg, 2004*