

Günther Bengel

**Verteilte Systeme**

# Lehrbuch

Die Reihe „Lehrbuch“, orientiert an den Lehrinhalten des Studiums an Fachhochschulen und Universitäten, bietet didaktisch gut ausgearbeitetes Know-how nach dem State-of-the-Art des Faches für Studenten und Dozenten gleichermaßen.

Unter anderem sind erschienen:

**Neuronale Netze  
und Fuzzy-Systeme**

von D. Nauck, F. Klawonn  
und R. Kruse

**Interaktive Systeme**

von Christian Stary

**Evolutionäre Algorithmen**

von Volker Nissen

**Stochastik**

von Gerhard Hübner

**Neuronale Netze**

von Andreas Scherer

**Rechnerverbindungs-  
strukturen**

von Bernhard Schürmann

**Rechnerarchitektur**

von Paul Herrmann

**Termersetzungssysteme**

von Reinhard Bündgen

**Konstruktion  
digitaler Systeme**

von Fritz Mayer-Lindenberg

**Theorie der Medizinischen  
Informatik**

von Hans-Jürgen Seelos

**SPSS für Windows**

von Wolf-Michael Kähler

**PASCAL**

von Doug Cooper und  
Michael Clancy

**Programmieren mit JAVA**

von Andreas Solymosi  
und Ilse Schmiedecke

**Bausteinbasierte Software**

von Günther Bauer

**Anwendungsorientierte  
Wirtschaftsinformatik**

von Paul Alpar, Heinz Lothar Grob,  
Peter Weimann und Robert Winter

**Software Engineering**

von Reiner Dumke

**Grundlagen der Theoretischen  
Informatik mit Anwendungen**

von Gottfried Vossen und  
Kurt-Ulrich Witt

**Grundlagen und Konzepte  
der Informatik**

von Hartmut Ernst

**Von Pascal zu Assembler**

von Peter Kammerer

**Verteilte Systeme**

von Günther Bengel

Günther Bengel

# **Verteilte Systeme**

Client-Server-Computing  
für Studenten und Praktiker

2., überarbeitete und erweiterte Auflage



Die Deutsche Bibliothek – CIP-Einheitsaufnahme  
Ein Titeldatensatz für diese Publikation ist bei  
Der Deutschen Bibliothek erhältlich.

1. Auflage März 2000
- 2., überarbeitete und erweiterte Auflage Januar 2002

Alle Rechte vorbehalten  
© Springer Fachmedien Wiesbaden 2002  
Ursprünglich erschienen bei Friedr. Vieweg & Sohn Verlagsgesellschaft in 2002

[www.vieweg.de](http://www.vieweg.de)



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Höchste inhaltliche und technische Qualität unserer Produkte ist unser Ziel. Bei der Produktion und Auslieferung unserer Bücher wollen wir die Umwelt schonen: Dieses Buch ist auf säurefreiem und chlorfrei gebleichtem Papier gedruckt. Die Einschweißfolie besteht aus Polyäthylen und damit aus organischen Grundstoffen, die weder bei der Herstellung noch bei der Verbrennung Schadstoffe freisetzen.

Konzeption und Layout des Umschlags: Ulrike Weigel, [www.CorporateDesignGroup.de](http://www.CorporateDesignGroup.de)

Gedruckt auf säurefreiem Papier

ISBN 978-3-528-15738-8

ISBN 978-3-322-96915-6 (eBook)

DOI 10.1007/978-3-322-96915-6

---

## Vorwort

---

Der bedeutendste Trend bei Informationssystemen in den letzten Jahren ist das Aufkommen vom Client-Server-Computing. Dieser Rechnerbetrieb ersetzt einerseits großrechnerdominierte oder zentralisierte Ansätze, andererseits bindet er lokale und isolierte Rechner zusammen und fördert ihre Zusammenarbeit. Die dazugehörige Strukturierungsmethode der Software in Clients und Server ist heute weit verbreitet und findet immer mehr Zugang und Einsatz bei der verteilten Verarbeitung.

Die Untergliederung der Software in Clients und Server ist eine Strukturierungsmöglichkeit bei verteilter Informationsverarbeitung. Um dieses Modell für Verteilte Systeme zu erschließen, wurden in Abschnitt 2 die möglichen Client-Server-Systeme klassifiziert. Die Klassifikation enthält neben Clients und Server eine weitere Klasse, nämlich den verteilten Prozess. Dies führt zu einem vertieften allgemeinen Verständnis für das Client-Server-Computing und der entstehenden Client-Server-Strukturen, weiterhin ermöglicht es eine nachfolgende einheitliche Betrachtung der verteilten Systeme unter dem Client-Server-Gesichtspunkt.

In die Programmierkonzepte und -methoden zur Gestaltung der Client-Server-Beziehungen führt Abschnitt 3 ein. Die einzelnen Programmiermethoden und -verfahren bauen in dem Sinne aufeinander auf, dass die höheren abstrakteren Programmierverfahren mit den Verfahren in einer darunter liegenden Schicht implementiert sind. Wir starten mit den Verfahren der untersten Ebene, den Sockets, und gehen über entfernte Prozeduraufrufe hin zu entfernten Methodenaufrufe und enden bei den Kommunikationsverfahren für das World Wide Web, dem heute aktuellsten und immer mehr eingesetzten Verfahren der Client-Server-Programmierung. Bei den vorgestellten und ausgewählten Programmierkonzepten habe ich mich an den Konzepten, die den Standard gesetzt haben oder die schon standardisiert sind, orientiert.

Abschnitt 4 führt in die fundamentalen und häufig benötigten verteilten Algorithmen ein. Es bildet mit der Klassifikation der Client-Server-Strukturen in Abschnitt 2 die Basis für die in Abschnitt 5 vorgestellten wichtigsten Server. Die vorgestellten Server starten bei einfachen Namens-Server und gehen dann über File-Server hin zu Transaktions-Server. Eng verknüpft mit den

---

atomaren Aktionen bei Transaktionen sind dann die nachfolgend beschriebenen Konkurrenzdienste zur Erreichung eines wechselseitigen Ausschlusses bei verteilten Prozessen. Neben der in Abschnitt 4 beschriebenen logischen Zeitordnung von Ereignissen werden anschließend Synchronisationsverfahren für physikalische Uhren beschrieben. Abschluss bilden schließlich die bei Verteilten Systemen eingesetzten Kryptosysteme und Authentifikationsprotokolle.

Der Inhalt des Buches entwickelte sich aus einer Vorlesung über Verteilte Systeme, die ich an der Fachhochschule Mannheim, Hochschule für Technik und Gestaltung, im Fachbereich Informatik seit 1993 halte. Diese Vorlesung liegt im zweiten Drittel des Informatikstudiums und setzt deshalb fundamentale Programmier-, Betriebssystem- und Kommunikationssoftware-Kenntnisse voraus.

Mit der Vorlesung einher gehen Übungen, die als Projekt ausgelegt sind. Das Projekt umfasst den Entwurf und die Programmierung eines Spieles als Verteiltes System, z. B. des Spieles „Mensch ärgere dich nicht“. Dabei soll eines der in Abschnitt 3 beschriebenen Kommunikationsverfahren zum Einsatz kommen. Beim Entwurf des Client-Server-Systems und den damit einhergehenden Kommunikationsprotokollen sollen die fundamentalen verteilten Algorithmen von Abschnitt 4 und die in Abschnitt 5 vorgestellten Dienste mit berücksichtigt werden und einfließen.

In dieser Form am Zustandekommen des Werkes beigetragen haben die Studenten mit konstruktiven Diskussionen über den Stoffinhalt und die Gliederung des Stoffes sowie mit ihren Hinweisen auf Fehler und didaktisch bessere Darstellungen. Mein Dank gilt weiterhin dem Kollegen aus dem Fachbereich Nachrichtentechnik Herrn Prof. Dr. Erich Eich und meinem alten Freund aus Schul- und Studienzeiten Herrn Dipl.-Math. Wolfgang Knierim. Herr Eich hat die Client-Server-Klassifikation kritisch durchleuchtet und hat bei den Client-Server-Programmierverfahren viele Verbesserungsvorschläge genannt. Herr Knierim, ein Vertreter aus der Praxis und der Industrie, hat durch seine Hinweise und Bemerkungen den praktischen Bezug des Werkes sichergestellt. Nicht vergessen und in meinen Dank mit einschließen möchte ich meinen Sohn Ralf Bengel und den Informatikstudenten Martin Hawlisch. Beide haben mich bei der Formatierung des Textes unterstützt und gaben viele Tipps im Umgang mit dem Textverarbeitungsprogramm.

---

Angesichts des sich dynamisch und stürmisch entwickelnden Gebiets der Verteilten Systeme, besonders im Bereich des World Wide Web, ist dieses Buch durch seine einführende Klassifikation von Client-Server-Strukturen eine Hilfe, sich besser in dem Dschungel der Techniken des Client-Server-Computing und der Client-Server-Programmierung zurechtzufinden. Beim Gang durch den Client-Server-Dschungel wünsche ich Ihnen nun viel Spaß und einen erfolgreichen Einsatz der Client-Server-Techniken bei Ihrer verteilten Anwendung.

Altrip, im Januar 2000

Günther Bengel

## **Vorwort zur 2. Auflage**

---

Das rasche Voranschreiten der Programmierkonzepte und –methoden des Client-Server-Computing und deren heutiges schnelles Einfließen in die Praxis, bedingten in der 2. Auflage Erweiterungen. Sie betreffen die Kommunikation und Koordination bei verteilter Verarbeitung und somit Abschnitt 3. Neu in diesem Abschnitt sind die Themen Remote Method Invocation (RMI), Extensible Markup Language (XML) und Enterprise JavaBeans (EJB).

Abschnitt 3 startet nun mit der Nachrichten-basierten Kommunikation, den TCP/IP-Sockets, und führt über Remote Procedure Calls hin zu den Objekt-basierten Kommunikationsmethoden, dem Remote Method Invocation (RMI) und CORBA. Die Web-basierten Programmierkonzepte gehen über HTML, Common Gateway Interface (CGI), Servlets, Extensible Markup Language (XML) und den XML-Parsern, hin zu den Applets. Den Abschluss von Abschnitt 3 bildet die Komponenten-basierte Methode der Enterprise JavaBeans, einem Rahmenwerk für die Client-Server-Programmierung.

Mit der 2. Auflage steht ein kostenloser Online-Service zur Verfügung. Die Internet-Adresse der Web-Seiten ist

<http://www.vts.fh-mannheim.de>

---

Die folgenden Informationen können auf den Web-Seiten gefunden werden:

- Informationen über den Autor mit der Email-Adresse, die zum Senden von Anmerkungen, Kommentaren und Berichtigungen verwendet werden kann.
- Alle Abbildungen des Buches zum Herunterladen; sie lassen sich in der Lehre einsetzen und wiederverwenden.
- Alle Programmbeispiele des Buches zum Herunterladen. Sie sollen den Leser ermuntern die Programme auszuprobieren und dienen zur Gewinnung von praktischer Erfahrung mit den Techniken der verteilten Programmierung.
- Ein Errata, d.h. Korrekturen zu Fehlern, die erst nach der Drucklegung des Buches gefunden wurden.
- Aktuelle Informationen zu Weiter- und Neuentwicklungen bzgl. der im Buch beschriebenen Technologien.

Die Web-Seiten werden kontinuierlich weiterentwickelt und ausgebaut. Zukünftige Erweiterungen sehen vor, zu jedem Kapitel Übungsaufgaben mit Musterlösungen anzubieten. Außerdem sollen größere Anwendungsbeispiele mit aufgenommen werden, die das in der Übung programmierte verteilte Spiel behandeln.

Altrip, im Dezember 2001

Günther Bengel



---

## Inhaltsverzeichnis

---

Einführung und Grundlagen .....	1
1.1 Netzwerkbetriebssysteme .....	7
1.2 Verteilte Betriebssysteme .....	14
1.3 Verteilte Systeme .....	17
1.3.1 Schichtenmodelle .....	17
1.3.2 Middleware .....	20
1.3.2.1 Distributed Computing Environment (DCE) .....	22
Klassifikation von Client-Server-Strukturen .....	27
2.1 Client-Server .....	29
2.1.1 Interaktionssemantik .....	30
2.1.2 Parallele Server .....	37
2.1.2.1 Threads .....	38
2.1.2.2 Implementierung von Threads .....	40
2.1.2.3 Pthreads .....	45
2.1.3 Serveraktivierungen .....	57
2.1.4 Serverzustände .....	58
2.1.5 Client Caching .....	61
2.2 Verteilte Prozesse .....	67
2.3 Client-Server-Server .....	68
2.3.1 Proxy .....	69
2.3.2 Broker .....	70
2.3.3 Trader .....	73
2.3.4 Balancer .....	74
2.3.5 Agent .....	75
2.4 Client-Server-Ketten .....	78
2.5 Client-Server-Bäume .....	80
Kommunikation und Koordination bei verteilter Verarbeitung .....	83
3.1 Nachrichten-basierte Kommunikation und Koordination .....	83

---

3.1.1	TCP/IP-Sockets .....	84
3.1.1.1	Datagram-Sockets .....	92
3.1.1.1.1	Anwendungsbeispiel echo-serving .....	96
3.1.1.2	Stream-Sockets .....	101
3.1.1.2.1	Anwendungsbeispiel rlogin.....	105
3.1.2	Kommunizierende Prozesse (ComPro) .....	108
3.2	Entfernter Prozeduraufruf (Remote Procedure Call) .....	110
3.2.1	Parameter- und Ergebnisübertragung.....	114
3.2.2	Identifikation und Binden der Aufrufpartner.....	119
3.2.3	Beispiele für RPC-Systeme .....	121
3.2.4	RPC-Programmierung.....	123
3.2.4.1	RPC-Compiler und RPC-Sprache.....	126
3.2.4.2	Anwendungsbeispiel für entfernte Prozeduren .....	133
3.3	Objekt-basiert .....	142
3.3.1	Remote Method Invocation (RMI) .....	144
3.3.1.1	Package java.rmi .....	146
3.3.1.2	Package java.rmi.registry .....	148
3.3.1.3	Package java.rmi.server .....	150
3.3.1.4	Serialisieren von Objekten .....	152
3.3.1.5	RMI-Programmierung .....	153
3.3.2	Common Object Request Broker Architecture (CORBA) .....	158
3.3.2.1	Objekt Modell .....	159
3.3.2.2	Object Management Architecture (OMA).....	161
3.3.2.3	Object Request Broker (ORB).....	164
3.3.2.4	Interface Definition Language (IDL).....	169
3.4	Web-basiert .....	176
3.4.1	HyperText Markup Language (HTML) .....	184
3.4.2	Dynamische Dokumente.....	199
3.4.2.1	Common Gateway Interface (CGI).....	199
3.4.2.2	Aktive Serverseiten .....	203
3.4.2.3	Servlets .....	205

---

3.4.2.4	Extensible Markup Language (XML).....	208
3.4.2.5	XML-Werkzeuge .....	220
3.4.3	Aktive Dokumente .....	225
3.4.3.1	Applets .....	226
3.4.3.2	Skripte .....	233
3.5	Komponenten-basiert .....	235
3.5.1	Enterprise JavaBeans-Architektur .....	236
3.5.2	EJB-Container .....	239
3.5.3	Enterprise-Beans.....	243
3.5.3.1	Home-Interface .....	244
3.5.3.2	Remote-Interface.....	245
3.5.3.3	Enterprise-Bean-Interface.....	246
3.5.3.3.1	Session-Beans.....	246
3.5.3.3.2	Entity-Beans .....	252
3.5.4	Entwickeln und Installieren einer Enterprise-Bean .....	256
3.5.5	EJB-Programmierung.....	258
	Fundamentale verteilte Algorithmen.....	261
4.1	Logische Ordnung von Ereignissen .....	264
4.2	Auswahlalgorithmen .....	267
4.3	Übereinstimmungsalgorithmen.....	273
	Dienste .....	279
5.1	Namensdienst.....	280
5.1.1	Domain Name System (DNS).....	281
5.1.2	DCE Directory Service.....	283
5.2	Filedienst .....	291
5.2.1	Network File System (NFS).....	291
5.2.2	Distributed File System (DFS).....	295
5.2.3	Filereplikationen.....	299
5.3	Transaktionsdienst .....	306
5.3.1	Transaktionen als Erweiterung des File-Servers .....	308
5.3.1.1	Recovery-Kontrolle .....	310

---

5.3.1.2	Konkurrenz-Kontrolle.....	316
5.3.2	Verteilte Transaktionen .....	322
5.4	Konkurrenzdienst.....	324
5.4.1	Zentralisierter Algorithmus.....	324
5.4.2	Verteilte Algorithmen .....	326
5.5	Zeitdienst.....	331
5.5.1	Zentralisierte Algorithmen zur Uhrensynchronisation.....	332
5.5.2	Verteilter Algorithmus zur Uhrensynchronisation.....	337
5.6	Sicherheitsdienst.....	338
5.6.1	Kryptosysteme .....	339
5.6.1.1	Symmetrische und asymmetrische Kryptosysteme .....	342
5.6.1.2	Schlüsselverteilungsproblem.....	344
5.6.2	Authentifikation .....	349
5.6.2.1	Passwort-basierte Authentifikation .....	350
5.6.2.2	Kryptographie-basierte Authentifikation .....	352
5.6.3	Kerberos.....	359
	Literatur.....	367
	Schlagwortverzeichnis .....	372