

Inhaltsverzeichnis

1	Einleitung	1
1.1	Die Bedeutung von Java	1
1.2	Die Bedeutung verteilter Systeme	7
1.3	Nebenläufigkeit, Verteilung und Persistenz	10
1.4	Literaturhinweise	15
I	Java in verteilten Systemen	17
2	Nebenläufigkeit in Java	19
2.1	Threads in Java	22
2.2	Synchronisationsmechanismen	30
2.2.1	Monitor	30
2.2.2	Semaphore	31
2.2.3	ReadWriteLock	33
2.3	Nebenläufigkeit und Verteilung	36
2.3.1	Server und Handler	36
2.3.2	Asynchrone Aufrufe	38
2.4	Literaturhinweise	41
3	Java Sockets	43
3.1	TCP/IP	43
3.1.1	Die Protokollschichten	44
3.1.2	Ports	46
3.2	Sockets	47
3.3	Streams	48
3.4	Filter	51
3.5	Ein Chat-System mit Sockets	53
3.6	Multicast	58
3.6.1	Ein Chat-System mit IP-Multicast	61
3.7	iBus: Ein zuverlässiger Multicast	62
3.7.1	Der Protokollstack	65
3.7.2	Ein Chat-System mit iBus	69
3.8	Literaturhinweise	71

4	RMI	73
4.1	Die Architektur von RMI	74
4.2	Ein einfaches Beispiel	76
4.2.1	Der Server	77
4.2.2	Der Client	79
4.3	Ein Chat-System mit RMI	80
4.4	Literaturhinweise	86
5	CORBA	87
5.1	Die Struktur der OMG	88
5.2	Der Standardisierungsprozeß	89
5.3	OMA – die Gesamtarchitektur	90
5.4	Die CORBA-Architektur	91
5.5	IDL	94
5.6	Der IDL-Compiler	98
5.7	Das IDL-Java-Mapping	102
5.8	Ein einfaches Beispiel	107
5.9	Ein Chat-System mit CORBA	109
5.10	Starten eines Servers	112
5.11	Starten eines Clients	113
5.12	Literaturhinweise	114
6	Voyager	117
6.1	Die Voyager-Laufzeitumgebung	120
6.2	Entfernte Objekte	121
6.3	Migration von Objekten	127
6.4	Entfernte Aufrufe	134
6.5	Gruppenkommunikation	137
6.5.1	Ein Chat-System mit Voyager Space	139
6.6	Literaturhinweise	141
7	Mobile Agenten	143
7.1	Aglets	148
7.1.1	Ein Chat-System mit Aglets	152
7.2	Mobile Agenten mit Voyager	156
7.2.1	Ein Chat-System mit Voyager Agenten	157
7.3	Literaturhinweise	160
8	JDBC	163
8.1	Aufbau und Struktur von JDBC	164
8.2	Verbindungsaufbau zu einer Datenbank	167
8.3	Anfragen und Antworten	168
8.4	Ein Beispiel: BulletinBoard	170
8.5	Literaturhinweise	174

9	Objektorientierte Datenbanken	175
9.1	ObjectStore	176
9.2	Zugriff auf persistente Objekte	177
9.3	Der Postprozessor	182
9.4	Das BulletinBoard mit ObjectStore	183
9.5	Reaktivierung von Objekten	190
9.6	Literaturhinweise	195
10	Eine persistente Programmiersprache	197
10.1	PJama	197
10.2	Der persistente Store	199
10.3	Stabilisierung	202
10.4	Garbage Collection	203
10.5	Ein einfaches Beispiel: Ein persistenter Zähler	204
10.6	Literaturhinweise	206
11	Tupelräume in Java	207
11.1	JavaSpaces	209
11.2	TSpaces	212
11.3	Literaturhinweise	216
12	Jini	217
12.1	Voraussetzungen	218
12.2	Dienste	219
12.3	Discovery and Join	220
12.4	Lookup	224
12.5	Leasing	228
12.6	Jini starten	229
12.7	Literaturhinweise	230
II	Ein verteiltes Java	231
13	Verteilte Programmiersprachen und Nebenläufigkeit	233
13.1	Eine verteilte Programmiersprache: Emerald	234
13.2	Die Vision der <i>Unified Objects</i>	237
13.2.1	Unterschiede zwischen lokaler und verteilter Programmierung	238
13.2.2	Latenzzeit	239
13.2.3	Speicherzugriff	240
13.2.4	Teilausfälle	241
13.2.5	Nebenläufigkeit	243
13.2.6	Wahrung des Unterschieds	245
13.3	Ist Java eine verteilte Programmiersprache?	245
13.4	Ein Konzept für Nebenläufigkeit: Eiffel SCOOP	249

13.4.1	Nebenläufigkeit durch Prozessoren	250
13.4.2	Nebenläufigkeit und Synchronisation	252
13.4.3	Verteilung	253
13.5	Anforderungen an ein verteiltes Java	254
13.6	Literaturhinweise	256
14	Virtuelle Prozessoren	257
14.1	Das Konzept des virtuellen Prozessors	260
14.2	Migration	263
14.3	Verteilung und Nebenläufigkeit	265
14.4	Persistenz	267
14.5	Literaturhinweise	269
15	Dejay: Ein verteiltes Java	271
15.1	Ein einfaches Beispiel	271
15.2	Virtuelle Prozessoren	275
15.3	Entfernte Objekte	277
15.4	Migration	281
15.5	Namensdienst	283
15.6	Persistenz	284
15.7	Ausnahmebehandlung	285
15.8	Der Compiler dejayc	286
15.9	Programmstart	287
15.10	Einschränkungen	288
15.11	Literaturhinweise	291
16	Beispiele	293
16.1	Verteilung	293
16.2	Nebenläufigkeit	303
16.2.1	Implementationsvergleiche	309
16.3	Persistenz	314
16.4	Literaturhinweise	318
A	Die Chat-Oberfläche	319
B	Die BulletinBoard-Oberfläche	323
	Abbildungsverzeichnis	329
	Literaturverzeichnis	333
	Index	347