
Inhaltsverzeichnis

Vorwort	v
1 Einleitung	1
2 Nebenläufigkeit	7
2.1 Nebenläufige Ausführung	7
2.1.1 Beispiel: Multiplikation von Matrizen	8
2.2 Atomare Teile und Synchronisation	9
2.2.1 Beispiel: Parallelrechner-Programmlauf	10
2.2.2 Annahmen zum Prozessor-Speichermodell	12
2.3 Explizite Synchronisation	14
2.3.1 Interferenz	15
2.3.2 Programmeigenschaften	16
2.4 Threads	21
2.4.1 Threads und Prozesse	22
2.4.2 Allgemeines Benutzungsschema	24
3 Java-Threads	27
3.1 Erzeugen und Beenden von Threads	28
3.2 Atomare Bereiche	38
3.3 Warten auf Bedingungen	50
3.4 Semaphore	52
3.4.1 Implementierung von Semaphoren	55
3.4.2 Barrieren mit Semaphoren	58
3.5 Das Bounded-Buffer-Problem	64
3.5.1 Produzenten-Konsumenten-Beispiel	68

3.5.2	Winkel zwischen Vektoren	73
3.6	Events und Applets	77
3.6.1	Ein Multithreaded Applet	78
4	Kommunikation	85
4.1	Kanäle, Senden und Empfangen	86
4.1.1	Programmeigenschaften	88
4.2	Netz-Topologien	89
4.3	channel mit Shared Memory in Java	92
5	Java-Sockets	101
5.1	Kanäle, Senden und Empfangen	102
5.1.1	Ströme und Kanäle	106
5.1.2	Packen und Versenden	110
5.1.3	Empfangen und Entpacken	111
5.1.4	Eine Kanalfabrik	112
5.1.5	Produzenten-Konsumenten-Beispiel	117
5.2	Virtuelle Kanäle	125
5.2.1	Der Client des Kanalservers	127
5.2.2	Der Kanalserver	130
5.2.3	Anwendung des Kanalservers	141
5.3	Broadcast und Multicast	147
5.3.1	Chat mit virtuellen Kanälen	153
6	Strategien zur Parallelisierung	163
6.1	Master - Slave	164
6.2	Client - Server	165
6.3	Workpile	166
6.3.1	Kleine Primzahlen	167
6.3.2	Thread-Pools	174
6.3.3	Rekursion	180
6.4	Pipelining	194
6.4.1	Ein Integer-Sort-Algorithmus	195

6.4.2	Integer-Sort-Implementierung	199
6.5	Scheduling	204
6.5.1	Prioritäten	205
6.6	Allokation von Ressourcen	207
6.6.1	Reader- und Writer-Problem	207
6.6.2	Dining Philosophers	216
7	Java OpenMP	223
7.1	OpenMP	223
7.2	Parallele Ausführung	225
7.2.1	Sichtbarkeit von Variablen	228
7.2.2	Lastverteilung	229
7.3	Atomare Bereiche	233
7.4	Reduktionen	238
8	Java MPI	241
8.1	Message Passing Interface (MPI)	242
8.2	Kanäle, Senden und Empfangen	247
8.3	Reduktionen	255
8.4	Matrix-Multiplikation	260
9	Java Message Service	265
9.1	Message Services	265
9.2	Das JMS API	269
9.3	Chatten mit Message Service	272
9.4	Implementierung mit virtuellen Kanälen	281
10	Remote Method Invocation	293
10.1	Die RMI-Architektur	293
10.2	RMI-Benutzung	295
10.2.1	RMI-Tools	296
10.2.2	Remote-Interface und Implementierung	299
10.2.3	Methoden-Registrierung	300
10.2.4	Methoden-Verwendung	303

10.3	Erzeugung von kleinen Primzahlen	306
10.3.1	Die genSmallPrimes-Methode.....	306
10.3.2	Die Verwendung von genSmallPrimes	310
11	JavaParty	317
11.1	Die remote-Spracherweiterung.....	317
11.2	Das Laufzeitsystem	320
11.2.1	Hallo Welt	322
11.3	Matrix-Multiplikation	324
11.4	Kommunikationstasks	330
11.4.1	Schnittstelle und Anwendung	331
11.4.2	Die Implementierung	334
11.4.3	Broadcast mit entfernten Objekten	341
11.5	Semaphore für entfernte Objekte	343
11.5.1	Zentrale Semaphore.....	344
11.5.2	Dezentrale Semaphore.....	345
11.5.3	Logische Uhren	355
11.5.4	Request Message Queue	357
11.5.5	Aufgaben	363
12	Ausblick	369
12.1	CORBA.....	369
12.1.1	CORBA-Benutzung	370
12.2	Java-Threads und POSIX-Threads	376
12.3	Granularität	379
12.4	Java-Speichermodell	381
12.5	Jini	383
12.6	JAXM	387
12.7	Java Grande	388
12.7.1	Stand der Entwicklung	389
12.7.2	Tipps für Numerik in Java	392
12.8	Schlussbemerkung	394

A	Zusammenfassung der Notation	395
A.1	con, atomic, await	395
A.2	channel, send, receive	396
A.3	UML-Sequenzdiagramme	397
B	Beispiele auf der Website	401
B.1	Benutzung der Beispielprogramme	401
B.2	Verzeichnis der Beispielprogramme	402
B.3	Inhalt des Jar-Files	404
	Literaturverzeichnis	405
	Index	413