

Inhaltsverzeichnis

1	Einführung	1
2	Nebenläufige Programme	7
2.1	Prozesse	8
2.1.1	Prozesse als unabhängige Kontrollflüsse	8
2.1.2	Synchronisation und atomare Aktionen	8
2.1.3	Unix-Prozesse: Schwergewichtige Prozesse	9
2.1.4	Threads: Leichtgewichtige Prozesse	18
2.1.5	Threads Objektorientiert	31
2.1.6	Übungen	39
2.2	Synchronisation	40
2.2.1	Das Synchronisationsproblem	40
2.2.2	Gegenseitiger Ausschluss und Bedingungssynchronisation von Threads .	50
2.2.3	Übungen	59
2.3	Kommunikation von Prozessen	60
2.3.1	Kommunikationskonzepte	60
2.3.2	Pipes: eine Puffer-Implementierung in Unix	70
2.4	Semaphore	79
2.4.1	Das Semaphorkonzept	79
2.4.2	IPC-Semaphore	84
2.4.3	Übungen	95
2.5	Monitore	97
2.5.1	Das Monitor-Konzept	97
2.5.2	Monitor-Abstraktionen	101
2.5.3	Puffer zwischen Prozessen	112

2.5.4	Monitor-Programme	121
2.5.5	Übungen	134
3	Verteilte Programme	136
3.1	BSD Sockets	137
3.1.1	Die Socketschnittstelle	137
3.1.2	Verbindungsorientierte Kommunikation	141
3.1.3	Verbindungslose Kommunikation	154
3.1.4	Übungen	157
3.2	Abstraktionen der Socketkommunikation	158
3.2.1	Eine funktionale Abstraktion	158
3.2.2	Eine objektorientierte Abstraktion	167
3.2.3	OO Abstraktion der verbindungslosen Kommunikation	177
3.2.4	Sockets und Threads	184
3.3	Hilfsmittel verteilter Programme	189
3.3.1	Timer	189
3.3.2	Rückrufe und Ereignisorientierte Programme	205
3.3.3	Ferne Prozeduren und Verteilte Objekte	221
3.4	Schichtenarchitektur	234
3.4.1	Architektur-Muster	234
3.4.2	Beispiel: Echo-Protokoll	237
3.5	Protokollspezifikation	249
3.5.1	Protokolle und ihre Spezifikation	249
3.5.2	Methodik der Spezifikation	252
3.5.3	Schichten als bessere Objekt-Verfeinerung	258
3.5.4	Übungen	261
3.6	Protokollentwurf	262
3.6.1	Analyse und Entwurf	262
3.6.2	Fensterprotokoll als zeitüberwachter Monitor	265
3.6.3	Implementierungsmodell	268
3.6.4	Übungen	282
	Literaturverzeichnis	284
	Index	286