

Contents

	<i>List of illustrations</i>	page viii
	<i>List of tables</i>	x
	<i>Foreword</i>	xi
	<i>Preface</i>	xiii
1	Introduction and overview	1
1.1	Parallel programming	2
1.2	Distributed programming	11
1.3	Real-time systems	12
	Summary	19
	Exercises	20
2	Sequential programming with Ada	23
2.1	Control structures	26
2.2	Subprograms	30
2.3	The Ada type model	35
2.4	Blocks and exceptions	62
2.5	Programming in the large	65
2.6	Object-oriented programming	76
2.7	Low-level programming	82
	Summary	102
	Exercises	103
3	Task basics	107
3.1	Defining tasks	107
3.2	The task life cycle	109
3.3	Task hierarchies	113
3.4	Exceptions	117
3.5	The implementation of Ada tasking	119
3.6	Other task features	119

vi	<i>Contents</i>	
	Summary	121
	Exercises	122
4	Communication and synchronization based on shared objects	126
	4.1 Mutual exclusion	126
	4.2 The protected object	130
	4.3 Synchronization	134
	4.4 The protected entry	135
	4.5 Restrictions	140
	4.6 Entry queues	141
	4.7 Some useful concurrent patterns	143
	4.8 Requeue and private operations	149
	4.9 Pragmas Atomic and Volatile	153
	4.10 Interrupts	155
	Summary	161
	Exercises	162
5	Communication and synchronization based on direct interaction	166
	5.1 The rendezvous	166
	5.2 The selective accept statement	171
	5.3 Entry call options	180
	5.4 State machines	181
	Summary	191
	Exercises	192
6	Distributed systems with Ada	195
	6.1 What are distributed systems?	195
	6.2 Middleware, architectures, and concepts	200
	6.3 DSA, the Distributed Systems Annex	202
	6.4 POLYORB: compilation chain and run-time for the DSA	212
	6.5 Advanced DSA concepts	215
	6.6 CORBA, the Common Object Request Broker Architecture	221
	6.7 Advanced CORBA concepts	236
	6.8 CORBA versus the DSA	247
	Summary	248
	Exercises	250
7	Real-time systems and scheduling concepts	251
	7.1 Task characteristics	253
	7.2 Real-time schedulers	257

	<i>Contents</i>	vii
7.3	Dependent tasks	278
	Summary	285
	Exercises	286
8	Real-time programming with Ada	294
8.1	Expressing time	295
8.2	Implementing periodic tasks	298
8.3	Ada implementation of the car application	303
8.4	Handling shared resources	305
8.5	The Ada scheduling model	308
8.6	Ravenscar	312
8.7	POSIX 1003.1b and its Ada binding	314
8.8	POSIX implementation of the car application	324
8.9	Ada tasks versus POSIX processes	328
	Summary	329
	Exercises	330
9	Tools for building and verifying real-time applications	333
9.1	Ada run-times to implement real-time applications	334
9.2	Some variants of the GNAT run-time	339
9.3	Validating scheduling of a system	347
	Summary	355
	Exercises	356
	<i>References</i>	359
	<i>Index</i>	365