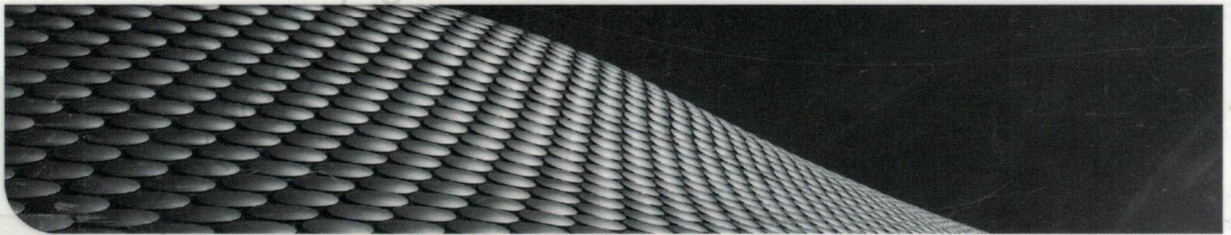




Newnes

EMBEDDED RTOS DESIGN

Insights and Implementation



มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

ห้องสมุดสาขาพระนครเหนือ



502006110

Contents

Foreword.....	xiii
Preface	xv
1 Program structure and real time	1
Why use an operating system?	1
Software structure	3
Embedded program models	5
What is real time?	6
2 Multitasking and scheduling.....	9
Tasks, threads, and processes	9
Schedulers.....	11
Run to completion scheduler.....	11
Round robin scheduler.....	12
Time slice scheduler	12
Priority scheduler.....	14
Composite scheduler.....	15
Task states.....	15
Task suspend	15
Other task states	16
Task identifiers.....	16
The context switch.....	17
What is a task?	17
Saving context	18
Dynamic task creation	18
Requirements for dynamic task creation	19
Implications of dynamic task creation	19
Interrupts	20
Interrupts and the kernel	20
Context save.....	20
Interrupts and the scheduler.....	21
Tick clock	21

3 RTOS services and facilities	23
Intertask communication services	23
Other RTOS services	28
Beyond the kernel	34
4 Nucleus SE	37
Nucleus SE target users	38
Design decisions and trade-offs	38
Memory utilization	39
Application program interface implementation	39
Scalability	40
Which application program interface?	41
Services	41
Parameter checking	42
Configuration	42
Naming conventions	42
Data structures	43
Key differences from Nucleus RTOS	44
5 The scheduler	47
Scheduling in Nucleus RTOS	47
Scheduling in Nucleus SE	48
Scheduler types	48
Task states	53
Options	55
Context saving	57
Global data	60
Implementing other scheduling schemes	61
6 Tasks	63
Configuring tasks	63
Task service calls	65
Task control services	66
Task utility services	72
Data structures	81
Unimplemented API calls	84
Compatibility with Nucleus RTOS	88

7 Partition memory	91
Using partitions	91
Configuring memory partitions	91
Partition allocation and deallocation services	93
Partition pool utility services	99
Data structures	104
Unimplemented API calls	106
Compatibility with Nucleus RTOS	108
8 Signals	111
Using signals.....	111
Configuring signals.....	111
Application program interface enables.....	112
Signals service calls.....	112
Signals send and receive services.....	113
Data structures	115
Unimplemented API calls.....	116
Compatibility with Nucleus RTOS	117
9 Event flag groups	119
Using event flags	119
Configuring event flag groups.....	119
Event flag service calls	120
Event flag group set and retrieve services.....	121
Event flag group utility services	128
Data structures	132
Unimplemented API calls.....	133
Compatibility with Nucleus RTOS	135
10 Semaphores	137
Using semaphores.....	137
Configuring semaphores.....	137
Semaphore service calls	138
Semaphore obtain and release services	139
Semaphore utility services.....	144
Data structures	149
Unimplemented API calls.....	150
Compatibility with Nucleus RTOS	152

11 Mailboxes	155
Using mailboxes	155
Mailboxes and queues	155
Configuring mailboxes	156
Mailbox service calls	157
Mailbox write and read services.....	158
Mailbox utility services.....	165
Data structures	170
Unimplemented API calls.....	171
Compatibility with Nucleus RTOS	174
12 Queues	177
Using queues	177
Queues and pipes	177
Configuring queues	177
Queue service calls	179
Queue write and read services	179
Queue utility services	190
Data structures	196
Unimplemented API calls.....	198
Compatibility with Nucleus RTOS	201
13 Pipes	203
Using pipes	203
Pipe utility services.....	217
Data structures	223
Unimplemented API calls.....	226
Compatibility with Nucleus RTOS	229
14 System time	231
Time in an RTOS.....	231
Configuring system time.....	232
System time service calls.....	233
System time set and obtain services	234
Data structures	235
Unimplemented API calls.....	236
Compatibility with Nucleus RTOS	236

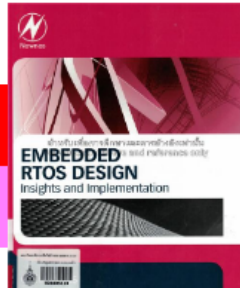
15 Application timers	237
Using timers.....	237
Configuring timers.....	237
Timer service calls.....	239
Timer services.....	239
Timer utility services.....	243
Data structures.....	248
Unimplemented API calls.....	251
Compatibility with Nucleus RTOS.....	253
16 Interrupts in Nucleus SE	255
Native and managed interrupts.....	255
Real time clock ISR.....	260
Nucleus RTOS compatibility.....	263
17 Nucleus SE initialization and start-up	271
Memory initialization.....	271
The main() function.....	272
The NUSE_Init() function.....	272
Application code initialization.....	280
Starting the scheduler.....	281
18 Diagnostics and error checking	283
Configuration checks.....	283
Application program interface parameter checking.....	285
Task stack checking.....	286
Version information.....	286
Compatibility with Nucleus RTOS.....	287
User diagnostics.....	288
Tracing and profiling.....	291
19 Unimplemented facilities and compatibility	293
The scheduler.....	293
API calls.....	294
API call functionality.....	296
Data sizes.....	298

20 Using Nucleus SE	301
What is Nucleus SE?.....	301
CPU and tool support.....	302
Configuring a Nucleus SE application.....	302
Which application program interface?.....	306
Debugging a Nucleus SE application.....	306
Nucleus SE configuration checklist.....	309
21 Nucleus SE reference information	311
Alphabetic API function list.....	311
Nucleus SE symbols and data structures.....	336
Typedefs.....	348
Global data.....	349
Kernel object tables.....	350
Obtaining Nucleus SE source code.....	357
Index.....	359

For educational use and reference only

Search and borrow a book

<https://lib.rmutp.ac.th/catalog/BibItem.aspx?BibID=b00106704>



Embedded RTOS design : insights and implementation / Colin Walls.

Author	Walls, Colin
Published	Amsterdam : newnes, 2021
Edition	1 st ed
Detail	xvi, 357 p. : ill ; 24 cm
Subject	Real-time data processing(+) Embedded computer systems(+)
ISBN	9780128228517
ประเภทแหล่งที่มา	 Book



For educational use and reference only