

Table of Contents



Acknowledgments	xxiii
Preface	xxv
Using This Book	xxvi
Coding Conventions	xxviii
1. Introduction to Threads	1
What Are Threads?	1
The Benefits of Threads	3
Throughput	3
Multiple Processors	4
User Interface Responsiveness	4
Server Responsiveness	4
Server Deadlock Avoidance	6
Program Structure	7
Communication	8
A Warning	9
Controlling Concurrency	10
Concurrent Programming	11
Summary	12
2. Getting Started	13
A Simple Example Using Threads	13
Threads Interfaces	15
Creating Threads	15
Using the Thread Start Routine Argument	16
Destroying Threads	17
Waiting for Threads to Exit	18
Getting the Current Thread-id	19
Comparing Thread-ids	19
Thread Attributes	19

Detached Threads	20
Summary	21
3. Synchronization	23
The Synchronization Problem	23
Synchronization Variables	25
Mutual Exclusion	26
Initializing and Destroying Mutexes	26
Locking and Unlocking Mutexes	28
Data Races	30
The POSIX Memory Model	32
Condition Variables	33
Initializing and Destroying Condition Variables	34
Waiting for and Signalling Conditions	35
Using Condition Variables	35
Using <code>pthread_cond_broadcast()</code>	42
Dynamic Package Initialization	43
Summary	45
4. Using Libraries in Multithreaded Programs	47
System Calls	47
<code>errno</code>	47
Thread-Safe Functions	48
Serializability	48
Restrictions on Concurrency	49
POSIX Thread-Safe Functions	50
Memory Allocation	51
<code>stdio</code>	51
C++ <code>iostream</code> Library	52
Explicitly Locked Functions	54
Thread-Unsafe Functions	54
Other Thread-Safety Variations	55
Package-Unsafe Functions	55
Object-Unsafe Functions	55
Summary	56
5. Threads and Process Resources	57
Process Resources	57

Unshared Resources	58
Positioned I/O	59
Arguments and Environment Variables	62
Timers and Waiting	63
Signals	63
Process Creation and Destruction	64
Using <code>fork()</code>	64
Waiting for Processes to Exit	65
Summary	65
6. Synchronization Strategies	67
Single Global Lock	67
Thread-Safe Functions	70
Structured Code Locking	70
Monitors	74
Internal Functions	74
Invariants	75
Releasing the Monitor Lock	77
Multiple Conditions	83
Structured Data Locking	83
Object Deletion	86
Locking and Modularity	86
Nested Monitors	87
Long Operations	87
Reentrant Monitors	89
Monitors Revisited	90
Deadlock	91
Lock Ordering	91
Resource Deadlocks	94
C++	95
Scoped Locking	96
Locking Guidelines	98
7. Using Threads	99
Creating Threads	99
Improving I/O Throughput	99
Using Multiple Processors	102



Improving User Interface Responsiveness	104
Improving Server Responsiveness	105
Program Structure	105
Server Deadlock Avoidance	105
Failure Attributes	106
Threads vs. Event Loops	107
Processes vs. Threads	107
Thread Paradigms	108
Deferring Work	108
Anticipating Work	109
Pipelines, Pumps, and Serializers	109
Sleepers and One-Shots	109
Deadlock Avoiders	110
Task Rejuvenation	110
Threading Data Structures	110
Linked Lists	111
Queues	116
Hash Tables	120
An Example	132
Summary	150
8. Thread-Specific Data	151
Thread-Specific Data	151
Allocating TSD keys	151
Accessing TSD Values	152
Using TSD	153
Deleting TSD keys	157
Considerations in Using TSD	157
Redefining Interfaces for Threads	157
Summary	160
9. Signals	161
Signal Mask	161
Signal Generation	162
Sending Signals to Threads	162
raise()	163
Signal Delivery	163
Signal Handlers	164



Async-Safe Functions	164
Nonlocal goto	165
Waiting for Signals	166
sigsuspend()	168
Pending Signals	169
Per-Thread Signal Handlers	169
Summary	172
10. Advanced Synchronization	173
Interprocess Synchronization	173
Synchronization Variables in Files	174
Interprocess Synchronization and Process Death	177
Semaphores	177
Named and Unnamed Semaphores	178
Creating and Opening Named Semaphores	179
Unlinking Named Semaphores	179
Initializing and Destroying Unnamed Semaphores	179
Waiting For and Posting Semaphores	180
Getting a Semaphore's Value	181
Using Semaphores	182
Semaphores and Signal Handlers	185
Semaphores and the Memory Model	185
Summary	185
11. Thread Cancellation	187
Destroying Other Threads	187
Cancelling Threads	192
Cancellation Points	192
Cancellation Cleanup Handlers	194
Cleanup Handlers and Condition Variables	196
Detaching Threads During Cancellation	198
Testing for Cancellation	199
Asynchronous Cancellation	199
Disabling Cancellation	201
Using Cancellation	202
Summary	203

12. Threads Scheduling	205
Thread Scheduling Models	205
Using Contention Scope	208
Thread Scheduling Attributes	209
Inheriting Scheduling Attributes	211
Thread Scheduling Policies	212
Priorities	213
Yielding	214
Scheduling Policies	214
Scheduling Allocation Domain	216
Priority Inversion	216
Priority Ceiling Protocol	218
Priority Inheritance	220
Scheduling Strategies	220
Thread Scheduling in Solaris 2	222
Concurrency and Process Contention Scope Threads	223
System Contention Scope Threads	226
Contention Scope and Process-Shared Synchronization Variables	226
Summary	227
13. Threads and Processes: Advanced Topics	229
Managing Thread Stacks	229
Default Stack Management	229
Thread Stack Size Limits	232
Managing Your Own Thread Stacks	233
Strategies for Using <code>fork()</code>	238
Fork Handlers	240
Summary	243
14. Advanced Synchronization Strategies	245
Critical Section Concurrency	245
Lessons from Queuing Theory	245
Read/Write Locks	248
Read/Write Lock Conditions	251
Breaking Up Locks	253
Explicitly Locked Functions	259
Recursive Locks	259

Object-Unsafe Functions	261
Locking Guidelines II	262
15. Multiprocessor Issues	263
Shared Memory Multiprocessors	263
Caches and Cache Consistency Protocols	264
Synchronization Instructions	270
Synchronization on Multiprocessors	271
Spin Blocking	271
Wait-Free Synchronization	273
Convoys	274
Livelock	275
Summary	275
16. Parallel Computation	277
Using Threads to Parallelize Algorithms	277
Thread Paradigms	278
Master-Slave Paradigm	279
Master-Slave Example: Matrix Multiply	281
Barrier Synchronization	286
Master-Slave Example: Relaxation	290
Spin Barriers	294
Workpile Paradigm	296
Workpile Controllers	298
Workpile Example: Quicksort	301
Pipeline Paradigm	303
Pipeline Example: Graphics Rendering	305
Pipeline Example: Relaxation	307
Parallelizing Using Compiler Techniques	311
Performance	316
Imbalance	316
Synchronization Overhead	316
Lock Contention	316
Data Sharing and False Sharing Overhead	317
Performance and Problem Size	317
Guidelines for Writing Parallel Programs	318
Summary	320

17. Multithreading Existing Code	321
Libraries	321
Thread-Safety	321
Cancellation-Safety	329
Fork-Safety	331
Establishing Fork Handlers	332
Applications	333
Multithreading a Graphical User Interface	333
Summary	338
18. Threads Development Environment	339
Compiling Multithreaded Programs	339
Other POSIX Libraries	340
Header Files	340
Static Data Race Analysis	342
lock_lint	342
Debugging	347
debugger on Solaris 2	347
Performance Tuning	348
Critical Path Analysis	348
Critical Section Analysis	351
Profiling	353
Tracing	353
Summary	362
A. Example Programs	363
B. UNIX International Threads	365
Overview	365
POSIX/UI Common Features	366
Additional Features of the UI Interface	371
Read/Write Locks	371
UI Resource Limits	373
Suspend and Continue	374
Daemon Threads	375
Forking Multithreaded Processes	376
Avoiding Data Races	377
Thread Scheduling	378

Concurrency Level	379
Thread Priorities	381
Scheduling Classes	381
Compiling and Linking UNIX International Programs	382
Header Files	382
C. Manual Pages	383
threads(3T)	385
cancellation(3T)	392
condition(3T)	398
mutex(3T)	406
pthread_atfork(3T)	413
pthread_attr_init(3T)	414
pthread_condattr_init(3T)	419
pthread_create(3T)	421
pthread_detach(3T)	428
pthread_equal(3T)	429
pthread_exit(3T)	430
pthread_join(3T)	432
pthread_kill(3T)	437
pthread_mutex_setprioceiling(3T)	439
pthread_mutexattr_init(3T)	440
pthread_once(3T)	443
pthread_self(3T)	444
pthread_setschedparam(3T)	445
pthread_sigmask(3T)	449
rwlock(3T)	452
semaphore(3T)	455
thr_min_stack(3T)	458
thr_setconcurrency(3T)	459
thr_suspend(3T)	461
thr_yield(3T)	462
sched_get_priority_max(3R)	463
sched_setparam(3R)	465
sched_setscheduler(3R)	467
sched_yield(3R)	469
sem_close(3R)	470



sem_destroy(3R)	471
sem_getvalue(3R)	472
sem_init(3R)	473
sem_open(3R)	474
sem_post(3R)	476
sem_unlink(3R)	478
sem_wait(3R)	479
sigwaitinfo(3R)	481
sigwait(2)	483
ctime(3C)	485
directory(3C)	489
getgrnam(3C)	493
getlogin(3C)	497
getpwnam(3C)	499
rand(3C)	503
string(3C)	504
sysconf(3C)	507
ttyname(3C)	512
D. Annotated Bibliography	515