

## Books

*Linux Kernel Development* (2nd Edition), Robert Love, Novell Press;  
ISBN-10: 9780672327209

*Programming with Posix Threads*, David R. Butenhof, Addison-Wesley Professional  
(May 26, 1997), ISBN-10: 9780201633924

*The Art of Multiprocessor Programming*, Maurice Herlihy and Nir Shavit, Morgan  
Kaufmann (March 14, 2008), ISBN-10: 0123705916

Java Concurrency in Practice, Peierls, Bloch, et al, Addison-Wesley Professional; 1st  
edition (May 19, 2006), ISBN-10: 0321349601

*Advanced Programming in the UNIX Environment*, Second Edition (Addison-Wesley  
Professional Computing Series), Richard Stevens, 2005, ISBN-10: 0321525949

*Unix Network Programming, Volume 1: The Sockets Networking API* (3rd Edition),  
Stevens, Fenner, and Rudoff. Addison-Wesley Professional; 3 edition 2003, ISBN-  
10: 9780131411555

*Operating System Concepts*, Silberschatz, Galvin, Gagne, Wiley & Sons 2008, ISBN-  
10: 9780470128725

*The Design and Implementation of the FreeBSD Operating System*, [Marshall Kirk McKusick](#) (Author), [George V. Neville-Neil](#) (Author), Addison-Wesley Professional; 1  
edition (August 12, 2004), ISBN-10: 9780201702453

*Coders at Work*, Peter Seibel, Apress, 2009, ISBN-10: 1430219483

## Web

[A Curious Course on Coroutines and Concurrency. David Beazley. Pycon 2009](#)

[Generator Tricks for System Programmers. David Beazley. Pycon 2008](#)

[Coroutines in C](#)

Inside the Python GIL, David Beazley, June 11 2009,

<http://blip.tv/carlfk/mindblowing-python-gil-2243379>

<http://www.dabeaz.com/python/GIL.pdf>

Communicating Sequential Processes - C. A. R. Hoare (June 21, 2004) <http://www.usingcsp.com/cspbook.pdf>

[The Libevent Book](#)

Revisiting Thread Priorities, David Beazley, <http://dabeaz.blogspot.com/2010/02/revisiting-thread-priorities-and-new.html>

Yieldable Threads, David Beazley  
<http://dabeaz.blogspot.com/2010/07/yieldable-threads-part-1.html>

Base Definitions (XBD) [http://www.unix.org/version3/xbd\\_contents.html](http://www.unix.org/version3/xbd_contents.html)

Jython (book) <http://www.jython.org/jythonbook/en/1.0/>

Inside the Erlang VM With Focus on SMP, Kenneth Ludin, Elang Users Conference, Stockholm(November 13, 2008) [http://www.erlang.se/euc/08/euc\\_smp.pdf](http://www.erlang.se/euc/08/euc_smp.pdf)

An Outsider's Look at Coroutines, Peter Portante, Pycon 2011 <http://pycon.tv/#/video/44>

The Linux Documentation Project <http://tldp.org/>

The Go Memory Model, [http://golang.org/doc/go\\_mem.html](http://golang.org/doc/go_mem.html)

Bell Labs and CSP Threads, Russ Cox, <http://swtch.com/~rsc/thread/>

Concurrency and Message Passing in Newsqueak, Rob Pike, Google Tech Talks, <http://video.google.com/videoplay?docid=810232012617965344#>

An Introduction to Concurrent Programming, Rob Pike, <http://herpolhode.com/rob/lec1.pdf>

[Poll versus Select versus Event Based](#)

[Benchmarking libevent against libev](#)

[Benchmarking BSD and Linux](#)

Nicholas Piel <http://nichol.as/>

GIL in Python 3.2  
<http://docs.python.org/py3k/whatsnew/3.2.html#multi-threading>

[Why Should I Care What Color the Bikeshed Is?](#)

Java Thread Concurrency Tutorial

<http://download.oracle.com/javase/tutorial/essential/concurrency/index.html>

[Using epoll with Python](#)

<http://code.google.com/p/pubsubhubbub/>

node.js is Backwards

<http://blog.ankurgoyal.com/post/6433642218/node-js-is-backwards>

100,000 tasklets with Stackless Python and Go

[http://dalkescientific.com/writings/diary/archive/2009/11/15/100000\\_tasklets.html](http://dalkescientific.com/writings/diary/archive/2009/11/15/100000_tasklets.html)

Large Scale Data Conditioning and Processing with Stackless Python and Pypes

<http://pycon.tv/#/video/48>

[Pypy Stackless](#)

[Continuations and Stackless Python](#)

[Stackless Python Meets the Twisted Matrix](#)

[The Implementation of Newsqueak by Rob Pike](#)

[Select in Stackless Python](#)

[Newsqueak](#)

[Stackless Scheduling](#)

[Python docs - multiprocessing](#)

[Feature Comparisons of Python Non-blocking IO Libraries](#)

[Poll vs Epoll once again](#)

[Comparing and Evaluating epoll, select, and poll Event Mechanisms](#)

[epoll Scalability Page](#)

[HTTP 1.1 Pipelining FAQ](#)

[Linux scalability: Accept\(\) scalability on Linux - Thundering Herd Problem](#)

[High Performance Server Design](#)

[IO Completion Ports](#)

[Reactor Design Pattern](#)

[Reactor Behavioral Pattern](#)

[network performance with libevent and libev](#)

[Binary Heap](#)

[Software Transactional Memory](#)

[Types of Interrupts](#)

[Little's Law of Queueing](#)

[Java Fork / Join Framework by Doug Lea](#)

[Tim Bray on C vs P](#)

[Parallelism != Concurrency](#)

[Algorithm + Strategy = Parallelism](#)

[Programming Paradigms for Dummies: What Every Programmer Should Know](#)

[Closure / Elang tradeoffs](#)

[Wide Finder Project](#)

[Systems that Never Stop \(and Erlang\)](#)

[Intel's James Reinders on parallelism: Part 1](http://news.cnet.com/8301-13556_3-10390800-61.html)

[Intel's James Reinders on parallelism - Part 2](#)

[Project Kenai - Resources for Future-of-Concurrency Research](#)

[Erlangs parallelism is not parallelism!](#)

[Runtime Support for Multicore Haskell](#)

[Dave Beazley: Revisiting thread priorities and the new GIL](#)

[TMalloc\(\)](#)

[Brett Slatkin's Proposal for Dealing with the GIL](#)

[/carbonio-and-bluenet-next-level-network-technology](#)

[How to Kill the GIL](#)

[StacklessExamples](#)

[TCP\\_CORK: More than you ever wanted to know](#)

## Testing

[On HTTP Load Testing](#)

[Autobench](#)

[The Web Server Benchmarking We Need](#)

## Math

[Amdahl's law](#)

[Work and Span](#)

[Gustafson's law](#)

[Stackless IO](#)

[Deadlock Fallacy](#)

## Software

[gevent.org](#)

[http://eventlet.net/](#)

<http://www.stackless.com/>  
<http://twistedmatrix.com/trac/>  
<http://www.tornadoweb.org/>  
<http://picklingtools.com/>  
<http://pypi.python.org/pypi/pyev/>  
<http://pypi.python.org/pypi/evnet/1.0-4>

<https://github.com/benoitc/gunicorn/>  
<http://www.xenoclast.org/autobench/>  
<http://opensource.hyves.org/concurrence/>  
<http://www.fapws.org/y>  
<http://tsung.erlang-projects.org/>  
[Dan Bernstein's IO Library Interface](#)  
[ab - Apache HTTP server benchmarking tool](#)  
<http://projectfortress.java.net/>  
[Go Scheduler](#)  
[gevent-socketio](#)  
[javascript socket.io](#)  
[Durus - a persistent object system for applications written in the Python](#)

## Operating System

[Processor \(CPU\) Affinity](#)  
[Linux Kernel Process Scheduling](#)  
[High Performance Servers](#)

## ZeroMQ

[ZeroMQ An Introduction](#)  
[Designing and Testing PyZMQ Applications – Part 1](#)

## Python

[PEP 3148 futures - execute computations asynchronously](#)