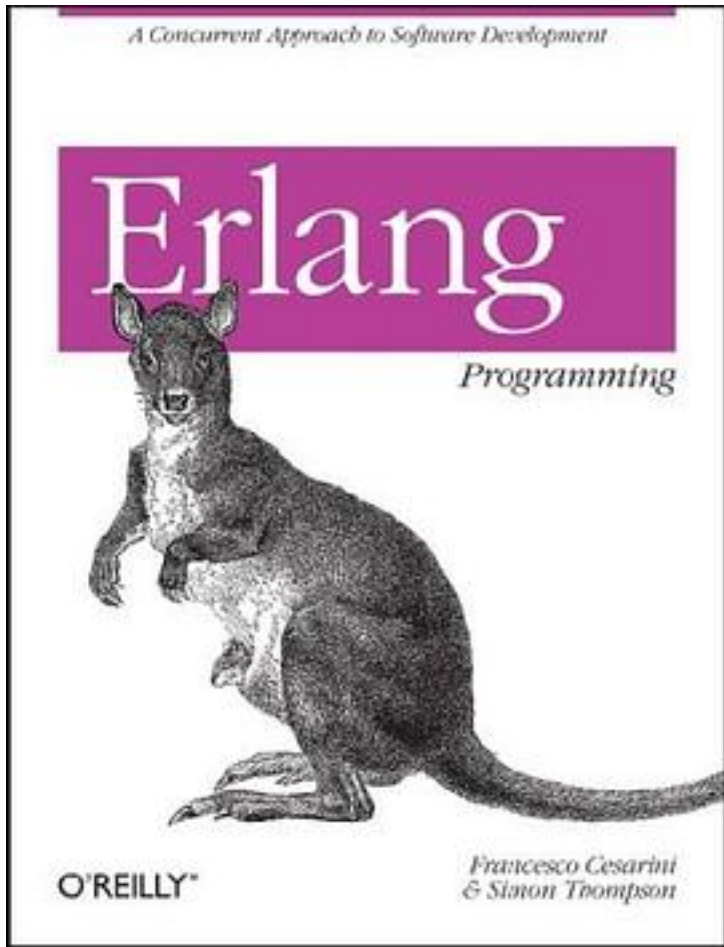


Erlang Programming



[Erlang Programming 下载链接1](#)

著者:Francesco Cesarini

出版者:O'Reilly Media

出版时间:2009-6-19

装帧:Paperback

isbn:9780596518189

This book is an in-depth introduction to Erlang, a programming language ideal for any situation where concurrency, fault tolerance, and fast response is essential. Erlang is gaining widespread adoption with the advent of multi-core processors and their new scalable approach to concurrency. With this guide you'll learn how to write complex concurrent programs in Erlang, regardless of your programming background or experience.

Written by leaders of the international Erlang community -- and based on their training material --

Erlang Programming focuses on the language's syntax and semantics, and explains pattern matching, proper lists, recursion, debugging, networking, and concurrency.

This book helps you:

Understand the strengths of Erlang and why its designers included specific features

Learn the concepts behind concurrency and Erlang's way of handling it

Write efficient Erlang programs while keeping code neat and readable

Discover how Erlang fills the requirements for distributed systems

Add simple graphical user interfaces with little effort

Learn Erlang's tracing mechanisms for debugging concurrent and distributed systems

Use the built-in Mnesia database and other table storage features

Erlang Programming provides exercises at the end of each chapter and simple examples throughout the book.

作者介绍:

Francesco Cesarini

14年来一直向学生、开发人员、测试人员、项目和技术经理教授Erlang/OTP技术。他协助在爱尔兰、美国和英国建立起了Erlang开发中心。

Simon

Thompson是一位肯特大学计算机实验室的逻辑和计算学教授，在过去25年里，他在那里教授本科生和研究生的计算学课程。

目录: Chapter 1 Introduction

Why Should I Use Erlang?

The History of Erlang

Erlang's Characteristics

Erlang and Multicore

Case Studies

How Should I Use Erlang?

Chapter 2 Basic Erlang

Integers

The Erlang Shell

Floats

Atoms

Booleans

Tuples

Lists

Term Comparison

Variables

Complex Data Structures

Pattern Matching

Functions

Modules

Exercises

Chapter 3 Sequential Erlang

Conditional Evaluations

Guards

Built-in Functions

Recursion

Runtime Errors

Handling Errors

Library Modules
The Debugger
Exercises
Chapter 4 Concurrent Programming
Creating Processes
Message Passing
Receiving Messages
Registered Processes
Timeouts
Benchmarking
Process Skeletons
Tail Recursion and Memory Leaks
A Case Study on Concurrency-Oriented Programming
Race Conditions, Deadlocks, and Process Starvation
The Process Manager
Exercises
Chapter 5 Process Design Patterns
Client/Server Models
A Process Pattern Example
Finite State Machines
Event Managers and Handlers
Exercises
Chapter 6 Process Error Handling
Process Links and Exit Signals
Robust Systems
Exercises
Chapter 7 Records and Macros
Records
Macros
Exercises
Chapter 8 Software Upgrade
Upgrading Modules
Behind the Scenes
Upgrading Processes
The .erlang File
Exercise
Chapter 9 More Data Types and High-Level Constructs
Functional Programming for Real
Funs and Higher-Order Functions
List Comprehensions
Binaries and Serialization
References
Exercises
Chapter 10 ETS and Dets Tables
ETS Tables
Dets Tables
A Mobile Subscriber Database Example
Exercises
Chapter 11 Distributed Programming in Erlang
Distributed Systems in Erlang
Distributed Computing in Erlang: The Basics
The epmd Process
Exercises
Chapter 12 OTP Behaviors
Introduction to OTP Behaviors
Generic Servers
Supervisors
Applications
Release Handling
Other Behaviors and Further Reading
Exercises
Chapter 13 Introducing Mnesia
When to Use Mnesia

Configuring Mnesia
Transactions
Partitioned Networks
Further Reading
Exercises
Chapter 14 GUI Programming with wxErlang
wxWidgets
wxErlang: An Erlang Binding for wxWidgets
A First Example: MicroBlog
The MiniBlog Example
Obtaining and Running wxErlang
Exercises
Chapter 15 Socket Programming
User Datagram Protocol
Transmission Control Protocol
The inet Module
Further Reading
Exercises
Chapter 16 Interfacing Erlang with Other Programming Languages
An Overview of Interworking
Interworking with Java
C Nodes
Erlang from the Unix Shell: erl_call
Port Programs
Library Support for Communication
Linked-in Drivers and the FFI
Exercises
Chapter 17 Trace BIFs, the dbg Tracer, and Match Specifications
Introduction
The Trace BIFs
Tracing Calls with the trace_pattern BIF
The dbg Tracer
Match Specifications: The fun Syntax
Match Specifications: The Nuts and Bolts
Further Reading
Exercises
Chapter 18 Types and Documentation
Types in Erlang
TypEr: Success Types and Type Inference
Documentation with EDoc
Exercises
Chapter 19 EUnit and Test-Driven Development
Test-Driven Development
EUnit
The EUnit Infrastructure
Testing State-Based Systems
Testing Concurrent Programs in Erlang
Exercises
Chapter 20 Style and Efficiency
Applications and Modules
Processes and Concurrency
Stylistic Conventions
Coding Strategies
Efficiency
And Finally...
Appendix Using Erlang
Getting Started with Erlang
Tools for Erlang
Where to Learn More
Colophon
• • • • • [\(收起\)](#)

标签

Erlang

Programming

编程

并发

Concurrency

O'Reilly

并行计算

erlang

评论

普通。

着重看了socket那块的编程，是一本好书~适合入门。网上有一些翻译<http://gashero.yeax.com/?p=55#id11>

结构还算比较清晰。

2年多以前写过半年的erlang，看完这本迅速捡了起来。不过第一次读的话，还是推荐erlang otp in action。

比较系统，比Armstrong的书

囫圇吞枣地看了一遍（完整的，哈哈），估计以后还要经常翻看。

涵盖了Erlang的各方面，比Joa的那本Erlang程序设计要详细。

个人感觉比erlang programming 那本书更系统一些

introduction to erlang programming

彻底的分布式。我的理解是Erlang有一个专业的虚拟机提供分布式的解决方案，它尝试从正面解决分布式的问题，而不是尝试搞成和单机一样，这一点是无奈的正视，也是正确的方向，一切忽略分布式的特长的办法都不可能解决分布式的问题。自己去从头做一个分布式系统，Erlang的思路可以借鉴，在特定的领域可以做的不一样，Erlang作为一个比较通用的方案已经做的很好了。分布式的问题依旧在，比如顺序问题，即使用Erlang，也一样避不开，抓住本质，才能做好分布式开发，没有银弹。

作为Programming Erlang 辅助书很不错，偏细节

算是语法书，比作者的那本讲的多。最后一章应该多读读。英文很简单，我都能读懂。

相对于《Erlang and OTP in Action》来说，更泛一点，也更偏入门一点。

其实erlang也不是很难嘛，估计是还没入门

入了kindle，第一本打算看它 从图书馆借了本实体书，每读一遍都有收获

相当不错，跟joe的互为补充不错，来自业界的经验

入门可看，可粗略浏览完后查阅细节

哈哈。当时和这个作者聊了下，人挺好玩的。我有签名版，谁要？

书不错，erlang这玩意，语言优美，思路简单，理解困难。

[Erlang Programming 下载链接1](#)

书评

这本书刚从卓越定的，还没有开始看。国内关于Erlang的书及其他资源实在是太少了。也没有看具体的介绍，希望这本书不要让人失望。国内那个Erlang社区好像已经没人更新了，我是一个新手
哎想找点资料都难

如果没有joe老头的原书中文版就罢了，有了，就不要再想着另起炉灶，搞出什么，基元，项元这些别扭的名词来，直接叫 原子和元组要死啊？
第5-9章是说看着怎么跟前几章风格不同，很别扭，一看果然不是一个人翻译的。有些内容我怀疑校对者到底看过没有。翻译不是把单词译成中文就...

o'reilly的素质没什么可以怀疑的，和armstrong的不同之处是此书似乎偏重实战，而不全是语言的描述。更难得的是国内已经开始翻译了，erlang能在国内竟有如此待遇。。

各种语句翻译得和机器翻译一样不明所以，关键还有大量的翻译错误，造成理解上的困难（尤其对于erlang新手）。
同时和以前即有的amstrong的翻译版的术语有些不统一，看得出来译者很没有诚意。
强烈不推荐看此翻译版！勉强看完一次，已再次入影印版。影印版推荐。

误买的英文影印版，到手后有点害怕。想退，但发现网友说中文版翻译不好，而且又贵了10几块，所以取消了退货。现在在看英文原版，原来真没有语言上的难度。贵在坚持。
看英文原版，不用担心译者将“atom”译成项元。不用担心译者那不顺畅的言语组织。

随着多内核处理器的流行，如果开发高效的并程序成了程序设计语言开发者们头疼的事情，Erlang作为一门不新的语言在这个时候成了计算机科学界的宠儿，其面对函数的程序设计思想，无side-effect的变量运用，高效和安全的消息传递以及模块热插拔的概念，使这门诞生于80年代的语言...

[Erlang Programming_下载链接1](#)