

<<Clojure编程>>

图书基本信息

书名：<<Clojure编程>>

13位ISBN编号：9787121197185

10位ISBN编号：7121197189

出版时间：2013-3-26

出版时间：电子工业出版社

作者：Chas Emerick,Brian Carper,Christophe Grand

译者：徐明明,杨寿勋

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Clojure编程>>

内容概要

Clojure是一种实用的通用语言，它是传奇语言LISP的方言，可与Ruby、Python等动态语言相媲美，更以无缝Java库、服务，以及拥有JVM系统得天独厚的资源优势而胜出。

本书既可以用来熟悉Clojure基础知识与常见例子，也可了解其相关的实践领域与话题，更可以看到这一JVM平台上的LISP如何帮助消除不必要的复杂性，为大家在编程实践中解决最具挑战性的问题开辟新的选择——更具灵活性，更适于Web编程和操作数据库，可以应付更为苛刻的应用程序安全要求，更有效的并发性和并行处理、数据分析能力，以及在未来云环境下的更大的发展潜力。

<<Clojure编程>>

作者简介

Chas Emerick是软件公司Snowtide informatics的创始人。

自2008年以来，致力于Clojure语言本身以及一些开源项目的开发；长期在cemerick.com撰写Clojure、软件开发实践、创业精神，及其他主题的文章。

Brian Carper是一位心理学研究领域的专业程序员，用Clojure从事数据分析与web开发。

他开发过一个把Clojure代码转化为CSS代码的编译器以及一个操作关系型数据库的类库，经常在briancarper.net上发表Clojure文章。

Christophe Grand是一个独立顾问，现居法国里昂。

他主要从事Clojure培训与编码，参与了Clojure语言的开发，撰写过Enlive和 Moustache类库，也是Counterclockwise——Clojure IDE for Eclipse的贡献者，经常在clj-me.cgrand.net上发表有关Clojure的文章。

<<Clojure编程>>

书籍目录

前言 第1章进入Clojure仙境 为什么要选择Clojure ?

获取Clojure Clojure REPL 不!

括号真的不会让你瞎了眼 表达式、操作符、语法以及优先级 同像性 Clojure Reader 标量字面量 注释 空格和逗号 集合字面量 reader的一些其他语法糖 命名空间 符号求值 特殊形式 阻止求值: quote 代码块 :do 定义Var: def 本地绑定: let 解构 (let, 第2部分) 定义函数: fn 条件判断: if 循环: loop和recur 引用var: var 和Java的互操作: 和new 异常处理: try和throw 状态修改: set!

锁的原语: monitor.enter和monitor—exit 小结 eval 这只是开始 第1部分函数式编程以及并发 第2章函数式编程 所谓函数式编程, 到底意味着什么?

谈谈值的重要性 关于值 值与可变对象的比较 一个关键性的选择 作为头等公民的函数以及高阶函数 Apply, Partial 函数(功能)的组合 编写高阶函数 利用可组合的高阶函数构建一个日志系统 纯函数 为什么纯函数很有意思?

现实生活中的函数式编程 第3章集合类与数据结构 抽象优于实现 集合: collection 序列: Sequence Associative 索引集合: Indexed 栈: stack set 有序集合: sorted 访问集合元素的简洁方式 习惯用法 集合、key以及高阶函数 数据结构的类型 列表: List vector set map 不可变性和持久性 持久性与结构共享 易变集合 元数据 用Clojure的集合来小试牛刀 标识符和循环引用 思维方式的改变: 从命令式到函数式 遍历、更新以及Zipper 总结 第4章多线程和并发 计算在时间和空间内的转换 delay future promise 简单的并行化 状态和标识 Clojure的引用类型 第2部分构建抽象 第3部分工具、平台以及项目 第4部分实战 第5部分杂项 索引

章节摘录

版权页：插图：I/O、事务以及嵌套的Send跟ref和atom不同的是，可以非常安全地利用agent来协调I/O或者其他类型的阻塞操作。

这使得agent成为任何使用ref以及Clojure的STM来维持状态的程序中不可或缺的一个组件。

而且由于agent独特的语义使得它们成为简化涉及异步I/O操作的理想组件——即使你不使用ref。

因为agent会串行执行所有发送给它的action，因此它给有副作用的操作提供一个很自然的同步点。你可以建立一个agent来保持比如一个指向文件或者网络socket的OutputStream、一个数据库的连接，或者消息队列的连接等。

你可以确定发送到agent的每个action在执行期间都能独占这个连接。

这使得它非常容易融合到Clojure的环境中来——包括ref和atom，它们的目标就是最小化它们跟周围环境的副作用。

你可能在想，agent怎么能用在STM的事务中呢？

发送一个agent action本身就是一个有副作用的操作，那么看起来似乎不能把它放在一个事务里面去执行，因为会重试。

进而可能会导致副作用发生多次，从而导致最终结果不对。

不过实际上不是这样的。

agent是跟Clojure的STM实现紧密融合的，在一个事务内通过send和send-off发送的action会被保持到事务成功提交。

这意味着如果一个事务被重试100次，被发送给agent的action也在事务成功提交之后被执行一次。

类似的，在一个agent action内部，对send和send-off的调用称为“嵌套发送”——也会被保持到直到这个action完成。

在这两种情况下，被发送的action都可能因为一个校验器的失败而被全部抛弃掉。

为了展示这些语义并且看看它们到底能给我们带来什么，让我们通过几个例子看看如何使用agent跟ref以及STM合作，在一个高度并行化、高IO负载的情况下简化对I/O操作的协调。

利用agent来记录引用状态的变更日志在181页“对Ref进行修改的细节”一节开发的游戏使用ref来保持角色的状态，证实了Clojure的STM在这种多人并发场景下的能力。

但是任何跟这个类似的游戏，特别是那种多个玩家同时玩的游戏，都会对玩家的行为进行跟踪和保存，以及它们的角色所受的影响（加血、减血）。

当然我们不会把这种类型日志、持久化以及其他I/O相关的东西写进核心游戏引擎：任何想做的持久化最终会由于事务重启而不一致。

<<Clojure编程>>

编辑推荐

《Clojure编程》基于新版，提供大量优良的样本代码，可让读者在快速入门、提高的同时，并不遗漏必要的初级概念。

抛开Clojure不谈，这也是一本值得推荐的编程语言书籍。

或许你现在还不能充分理解Clojure和函数式编程的精髓，或许目前的日常工作还用不到Clojure，但这绝对是一个值得你重视的主题，本书据此精心设计。

作者在这一领域有足够的发言权，有能力引导读者从下载/安装开始学起，直到掌握用起进行web开发。

《Clojure编程》写作精良，平易近人，具有非常好的“信噪比”，可谓最实用而全面的Clojure图书，对Clojure的状态、标识及并发理念，函数式编程以及Clojure生态系统都有完整阐述。

如果有人问你“我应该读什么书上的Clojure”，你可以坚定地指向它！

名人推荐

“ O ’ Reilly Radar 博客有口皆碑。

” ——Wired “ O ’ Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。

” ——Business 2.0 “ O ’ Reilly Conference 是聚集关键思想领袖的绝对典范。

” ——CRN “ 一本O ’ Reilly 的书就代表一个有用、有前途、需要学习的主题。

” ——Irish Times “ Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra 的建议去做了：‘ 如果你在路遇到岔路口，走小路（岔路）。

’ 回顾过去Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。

” ——Linux Journal

<<Clojure编程>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>