

在Erlang程序里处理状态 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/461/2021_2022__E5_9C_A8Erlang_E7_c104_461635.htm 从题外话说起：据我亲身经历，很多令人郁结的程序员最大的问题不是不熟悉语言和类库，不是不了解算法，不是不会用工具，而是对程序里的信息流没有概念弄不清哪些信息应该在什么地方、信息从哪里来、经过怎样的转换、到哪里去。而顺序化编程语言（C、C++、Java、C#……）很大程度上加重了这个毛病：在一些不那么漂亮的代码里经常可以看到被滥用的static方法和变量，说到底还是不恰当的全局变量的延续，说到底还是没弄清楚哪些信息应该在哪些位置出现。而Erlang的编程练习对此很有帮助。没有全局变量，变量赋值后就不能改变。于是一些常见的bad smell自然而然地就不会出现了，一些常用的重构手法自然而然地就用不上了。不过呢，这个世界毕竟是有状态的。比如说一个Stomp server就需要记住哪个client订阅了哪个频道。于是当你认真思考“什么信息应该在什么地方”这个问题时，Erlang的几种选择就显得很有意思了。参数传递 只有当你认真思考的时候，你才会发现原来很多信息都是很容易得到的。要控制一个函数的行为，最简单也最常用的办法就是改变传递给它的参数。如果这个函数需要一种新的状态，也许那意味着给它增加一个参数。1、进程字典 调用put和get方法可以把信息放入一个“每个进程一个实例”的字典。例如random在字典里放了一个名叫random_seed的变量，用来生成伪随机数。2、ETS 同样是一张二维表，ETS里的信息是所有进程都能访问的。例如Stomperl 需要记录哪个client订阅哪

个mailer进程，显然所有监听socket的进程都需要了解这个订阅信息，才能正确分发消息。于是订阅信息就应该（至少）在ETS里保存。

3、DETS ETS只在内存中存在，这意味着两件事：第一，程序结束数据就消失；第二，数据只能在一个节点共享。DETS的API和ETS相似，但它是基于文件的，所以持久保存和多节点共享都是题中应有之义。注意，ETS和DETS保存的数据都必须是tuple。

4、Mnesia 这是一个真正的数据库。功能齐备，并且仍然软实时。

以上四种方式的排列不是随机的。应该首先考虑靠前的手段，如果有明确的理由表明一种手段不能满足需要时才可以考虑比较靠后的手段。这很费脑子，有时让人沮丧。但经过深思熟虑的程序好过不假思索的程序，发现自己犯错好过犯错而不自知。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com