## Recon-Erlang线上系统诊断工具

源码在[这里](https://github.com/ferd/recon)。Recon, 它的文档非常清晰，见[这里](http://ferd.github.io/recon/index.html)。

Recon主要包括三个模块：

recon  
Main module, contains basic functionality to interact with the recon application. It includes functions to gather information about processes and the general state of the virtual machine, ports, and OTP behaviours running in the node. It also includes a few functions to facilitate RPC calls with distributed Erlang nodes.

recon\_lib  
Regroups useful functionality used by recon when dealing with data from the node. Would be an interesting place to look if you were looking to extend Recon’s functionality

recon\_alloc  
Regroups functions to deal with Erlang’s memory allocators, or particularly, to try to present the allocator data in a way that makes it simpler to discover the presence of possible problems.

和一系列脚本用于在发生crashdump的时候帮助用户分析到原因, 设计的时候充分考虑到了对系统的最小影响，在线上使用是很安全的。

其中最有价值的是 recon\_alloc， 基本上把内存分配器的细节和复杂都屏蔽起来，用户可以很好的看到内存工作的效率.

<https://github.com/zhongwencool/observer_cli>

[**Linux/Unix笔记本**](http://www.cnblogs.com/ggjucheng/archive/2012/08/18/2645321.html)

<http://www.cnblogs.com/ggjucheng/archive/2012/08/18/2645321.html>

关于send/recv/accept

gen\_tcp:send

<http://blog.yufeng.info/archives/1581>

<http://blog.yufeng.info/archives/336>

当我们调用gen\_tcp:send的时候, kernel模块会根据gen\_tcp socket的类型决定调用相应的模块. 这个模块要么是inet\_tcp, 要么是inet6\_tcp. 这个模块会把发送请求委托给prim\_inet模块. prim\_inet模块首先检查Socket是否合法, 如果合法然后调用erlang:port\_command把系统推到ERTS运行期.  
这个推的结果有2个: 1. 成功, 进程挂起等待运行期的反馈. 2. 失败,立即返回.什么情况下会失败呢?  
1. 驱动不支持soft\_busy, 但是我们用了force标志  
2. 驱动已经busy了, 但是我们不允许进程挂起.

inet\_drv内部每个socket都有个消息队列, 保持着上层推来的消息. 这个消息队列有上下水位线的. 当消息的字节数目超过了高水位线的时候, inet\_drv就把socket标志为busy. 这个busy要到队列的字节数少于低水位线的时候才解除.

如果一条消息成功的推到协议栈, 那么tcp 驱动会给调用者进程发送{inet\_reply,S,Status}消息, 反馈结果. 这时候调用者进程也就是tcp:send返回, 完成了整个流程.

总结起来就是，gen\_tcp数据的发送需要占用宿主进程的reds,这也可能造成宿主进程被挂起，在设的时候尽量避免一个进程拥有太多的port. 更好的做法是: 手工把gen\_tcp的2个步骤分开做:  
1. 不停的erlang:port\_command(S, Data, OptList)  最好加上force标志   
2. 被动等待{inet\_reply,S,Status} 消息.

3. 并且使用inet:setopts(Socket, Options)设置socket的{send\_timeout, Integer} {send\_timeout\_close, Boolean}选项