erlang学习(2)

李小红 lixiaohong@gmail.com
黄冬 huangdong@gmail.com

你将会学到

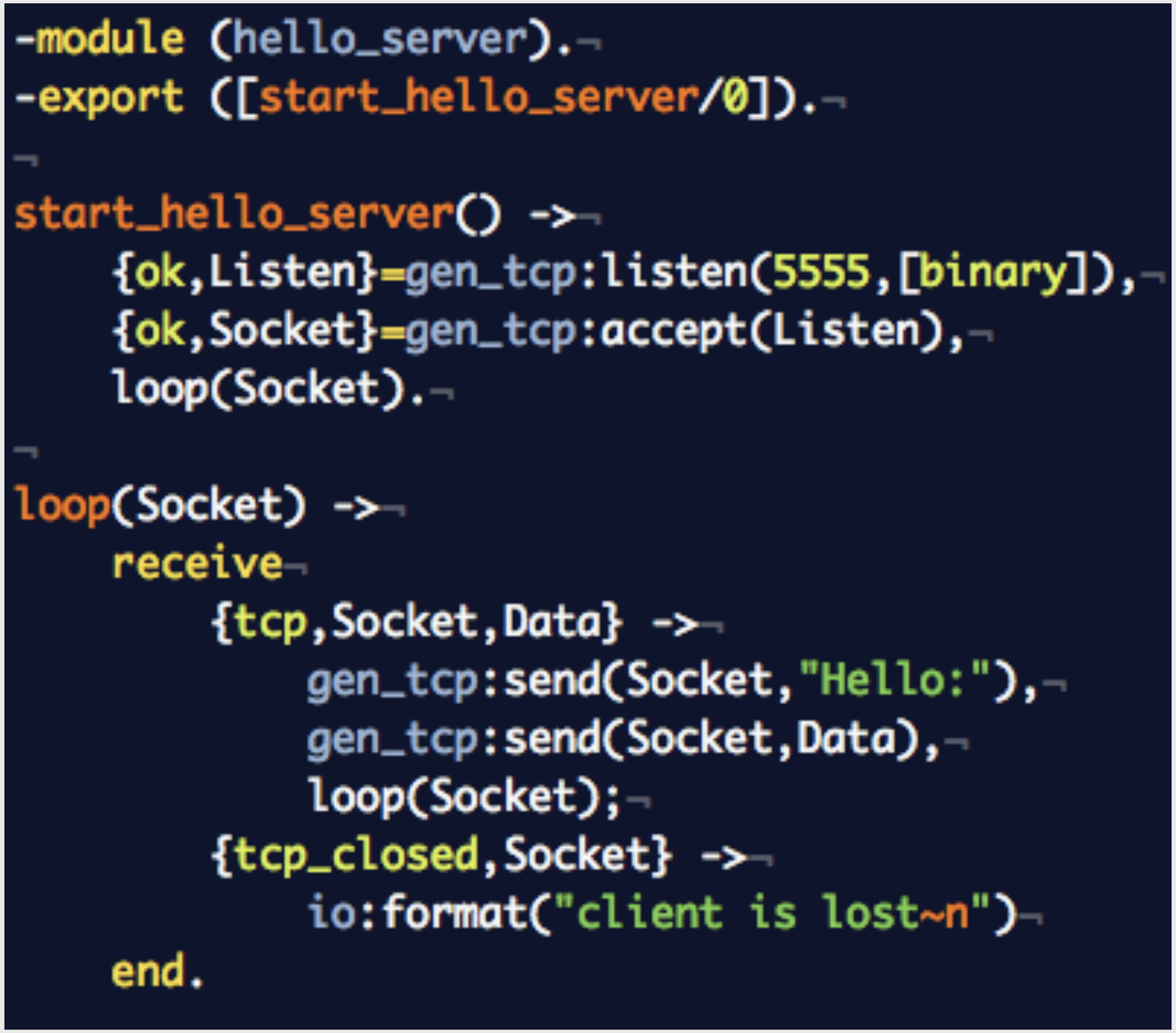
- 一个简单的Hello Server
- Table的并发消息机制
- Table的简化版Server

最简单的TCP Server

需求

- 监听5555端口
- 向每一个到来的消息say Hello

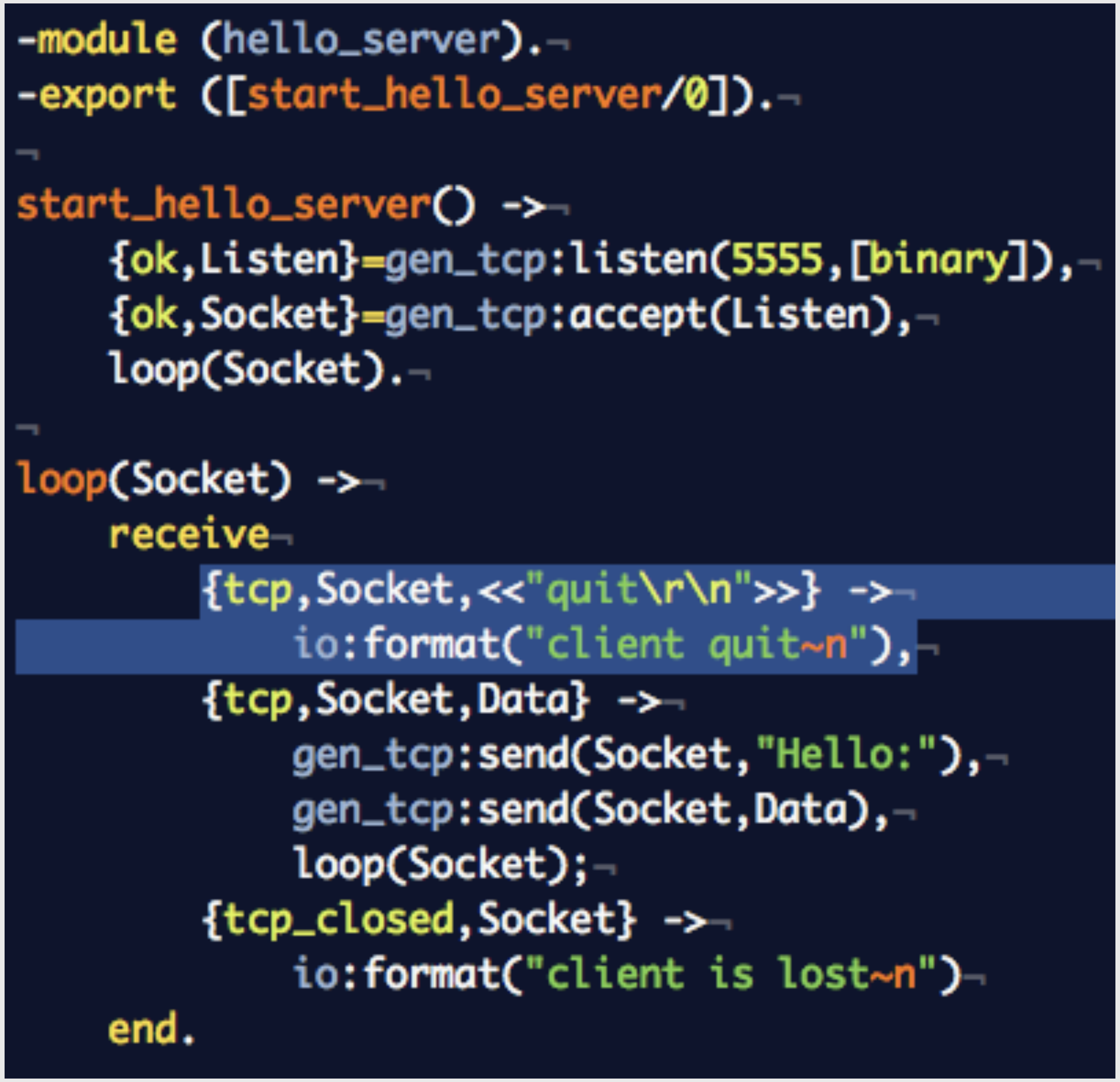
代码


```
-module (hello_server).
-export ([start_hello_server/0]).

start_hello_server() ->
    {ok,Listen}=gen_tcp:listen(5555,[binary]),
    {ok,Socket}=gen_tcp:accept(Listen),
    loop(Socket).

loop(Socket) ->
    receive
        {tcp,Socket,Data} ->
            gen_tcp:send(Socket,"Hello:"),
            gen_tcp:send(Socket,Data),
            loop(Socket);
        {tcp_closed,Socket} ->
            io:format("client is lost~n")
    end.
```

运行并继续

- 编译
- 运行
- 加功能：
  - 输入quit，断开连接
  - 自由发挥

增加server的处理

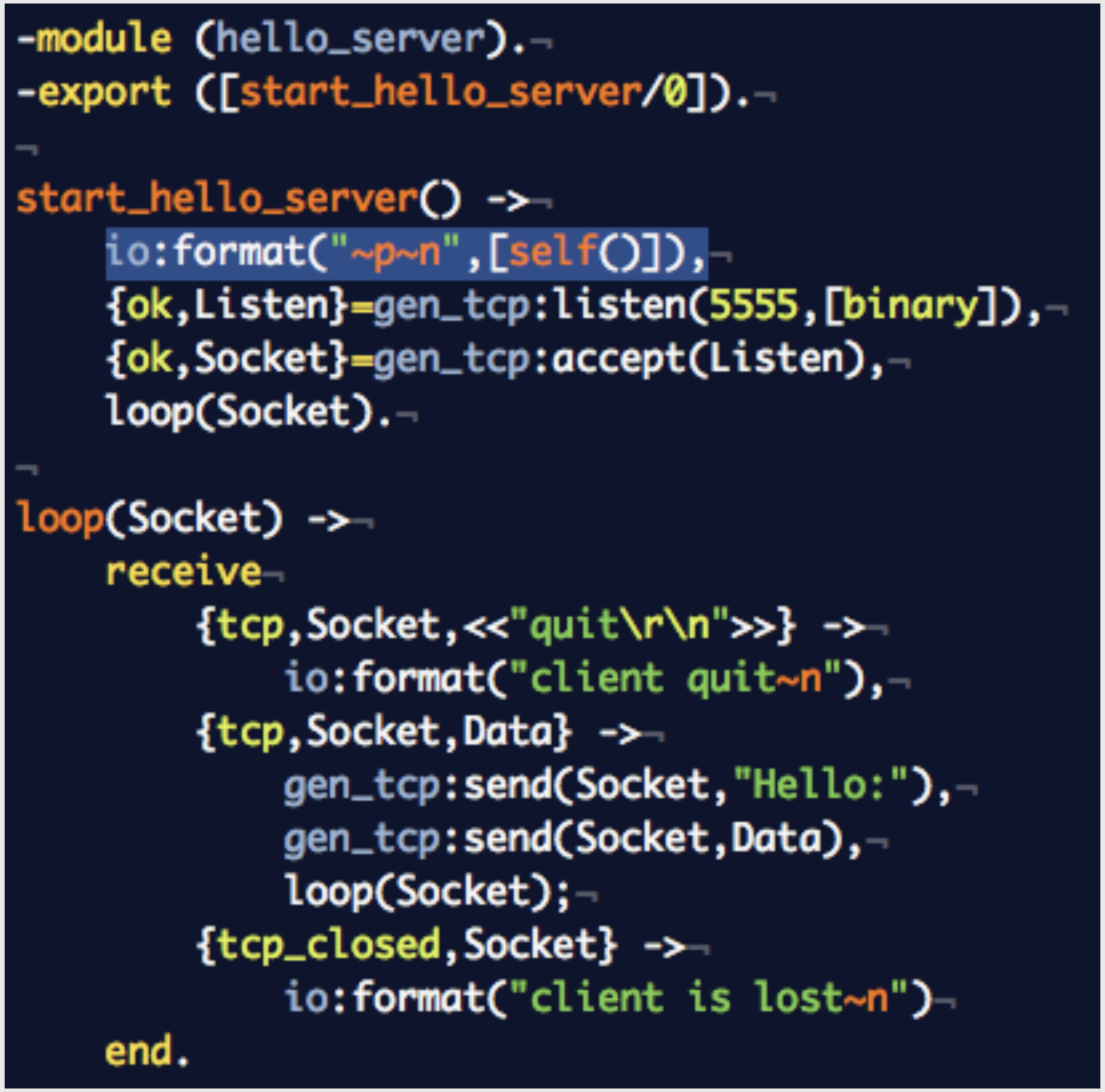


```
-module (hello_server).
-export ([start_hello_server/0]).

start_hello_server() ->
    {ok,Listen}=gen_tcp:listen(5555,[binary]),
    {ok,Socket}=gen_tcp:accept(Listen),
    loop(Socket).

loop(Socket) ->
    receive
        {tcp,Socket,<<"quit\r\n">>} ->
            io:format("client quit~n"),
        {tcp,Socket,Data} ->
            gen_tcp:send(Socket,"Hello:"),
            gen_tcp:send(Socket,Data),
            loop(Socket);
        {tcp_closed,Socket} ->
            io:format("client is lost~n")
    end.
```

运行并测试

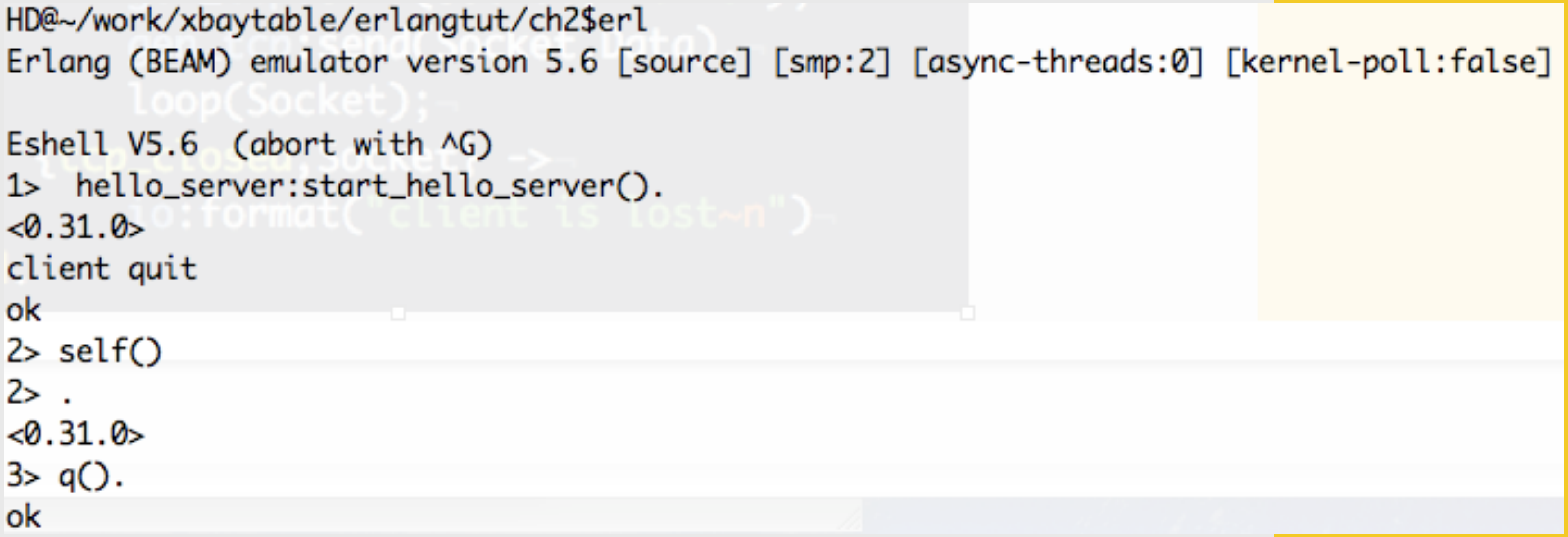
- client输入quit后
  - 服务器完成
  - client没有断开
- 为什么?

加上代码看看



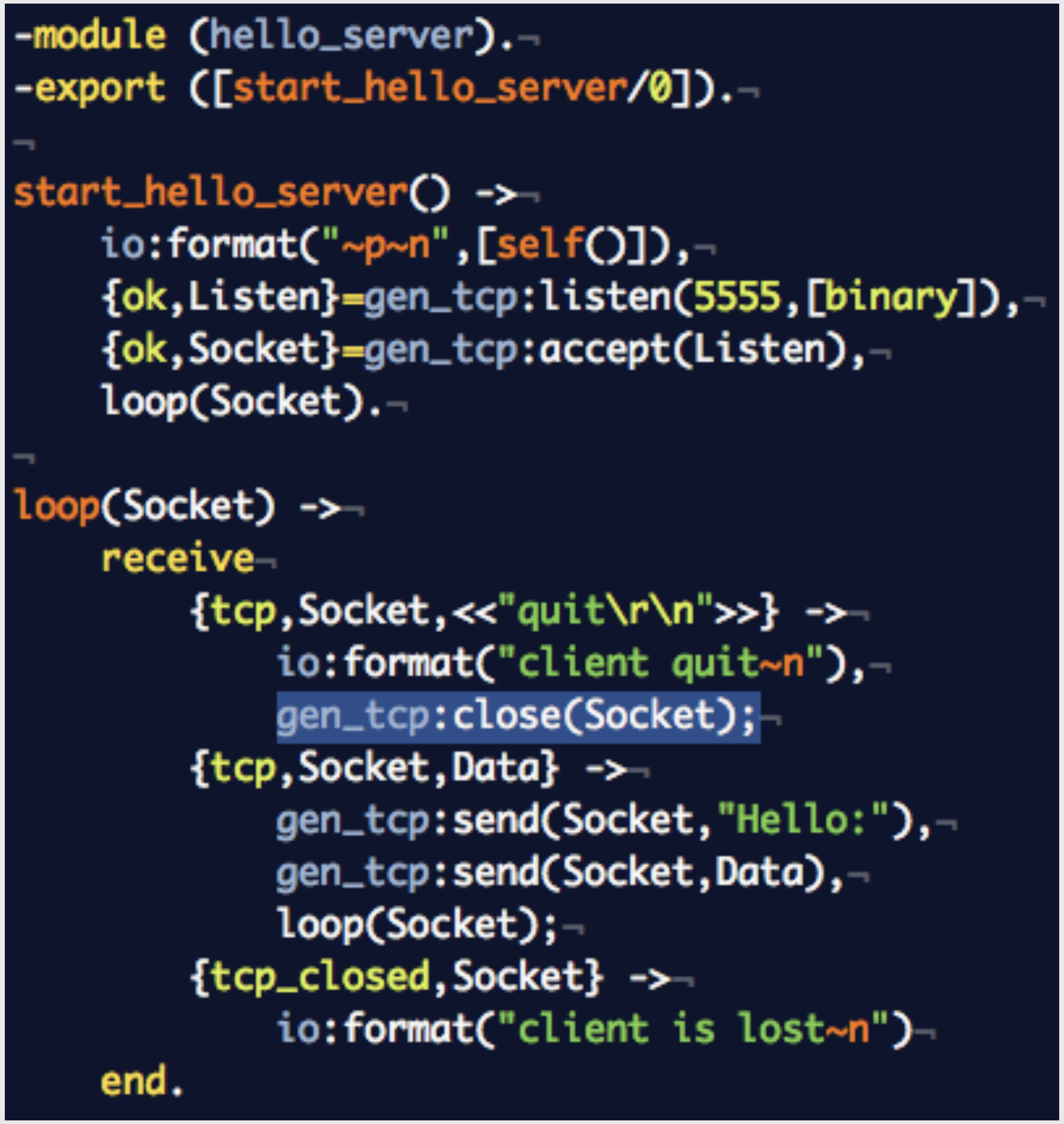
```
-module (hello_server).
-export ([start_hello_server/0]).

start_hello_server() ->
    io:format("~p~n",[self()]),
    {ok,Listen}=gen_tcp:listen(5555,[binary]),
    {ok,Socket}=gen_tcp:accept(Listen),
    loop(Socket).

loop(Socket) ->
    receive
        {tcp,Socket,<<"quit\r\n">>} ->
            io:format("client quit~n"),
        {tcp,Socket,Data} ->
            gen_tcp:send(Socket,"Hello:"),
            gen_tcp:send(Socket,Data),
            loop(Socket);
        {tcp_closed,Socket} ->
            io:format("client is lost~n")
    end.
```

原来self()没有消失



```
HD@~/work/xbaytable/erlangtut/ch2$erl
Erlang (BEAM) emulator version 5.6 [source] [smp:2] [async-threads:0] [kernel-poll:false]

Eshell V5.6  (abort with ^G)
1>  hello_server:start_hello_server().
<0.31.0>
client quit
ok
2> self()
2> .
<0.31.0>
3> q().
ok
```

好好处理尾巴



```
-module (hello_server).
-export ([start_hello_server/0]).

start_hello_server() ->
    io:format("~p~n",[self()]),
    {ok,Listen}=gen_tcp:listen(5555,[binary]),
    {ok,Socket}=gen_tcp:accept(Listen),
    loop(Socket).

loop(Socket) ->
    receive
        {tcp,Socket,<<"quit\r\n">>} ->
            io:format("client quit~n"),
            gen_tcp:close(Socket);
        {tcp,Socket,Data} ->
            gen_tcp:send(Socket,"Hello:"),
            gen_tcp:send(Socket,Data),
            loop(Socket);
        {tcp_closed,Socket} ->
            io:format("client is lost~n")
    end.
```

Table Server原型

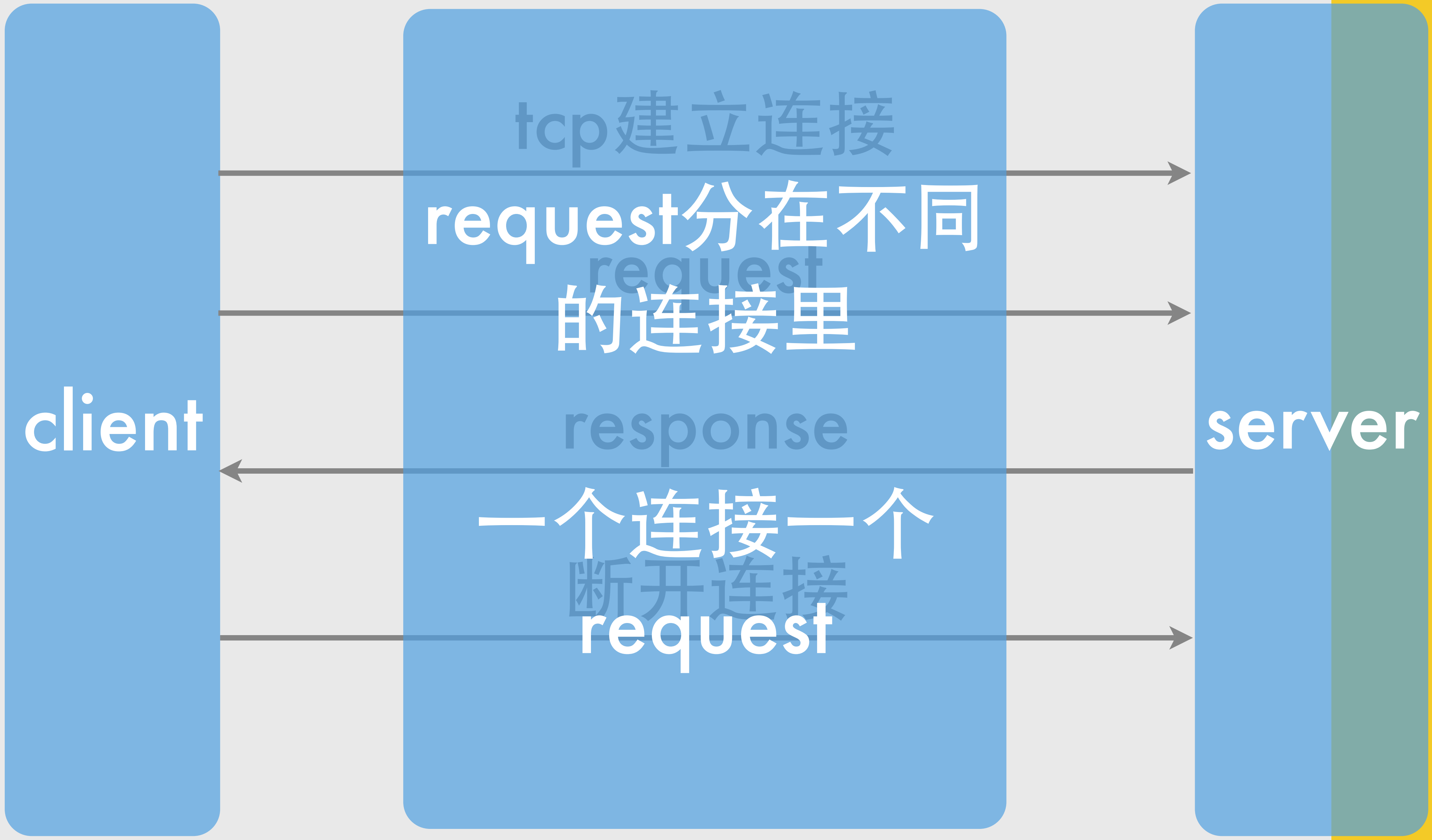
需求

- 监听5555端口
- 服务器接收消息
  - {set,Name,Value}
  - {get,Name}
  - {delete,Name}

如何组织数据?

- 如果在一个连接里发送多个数据包，如何从一个二进制流中取出各个数据体
- 数据指令的编码和解码方法（序列化和反序列化）

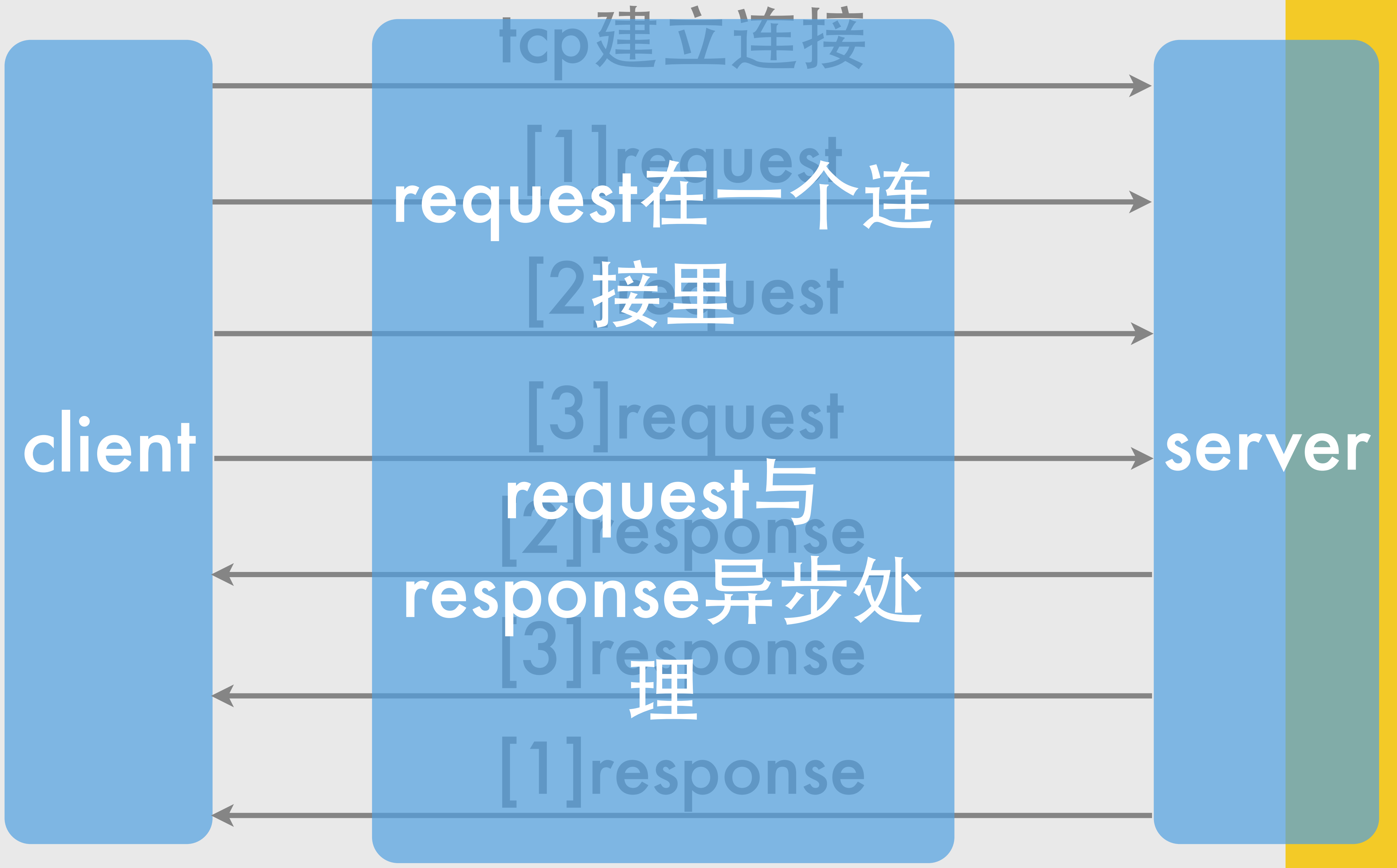
短连接



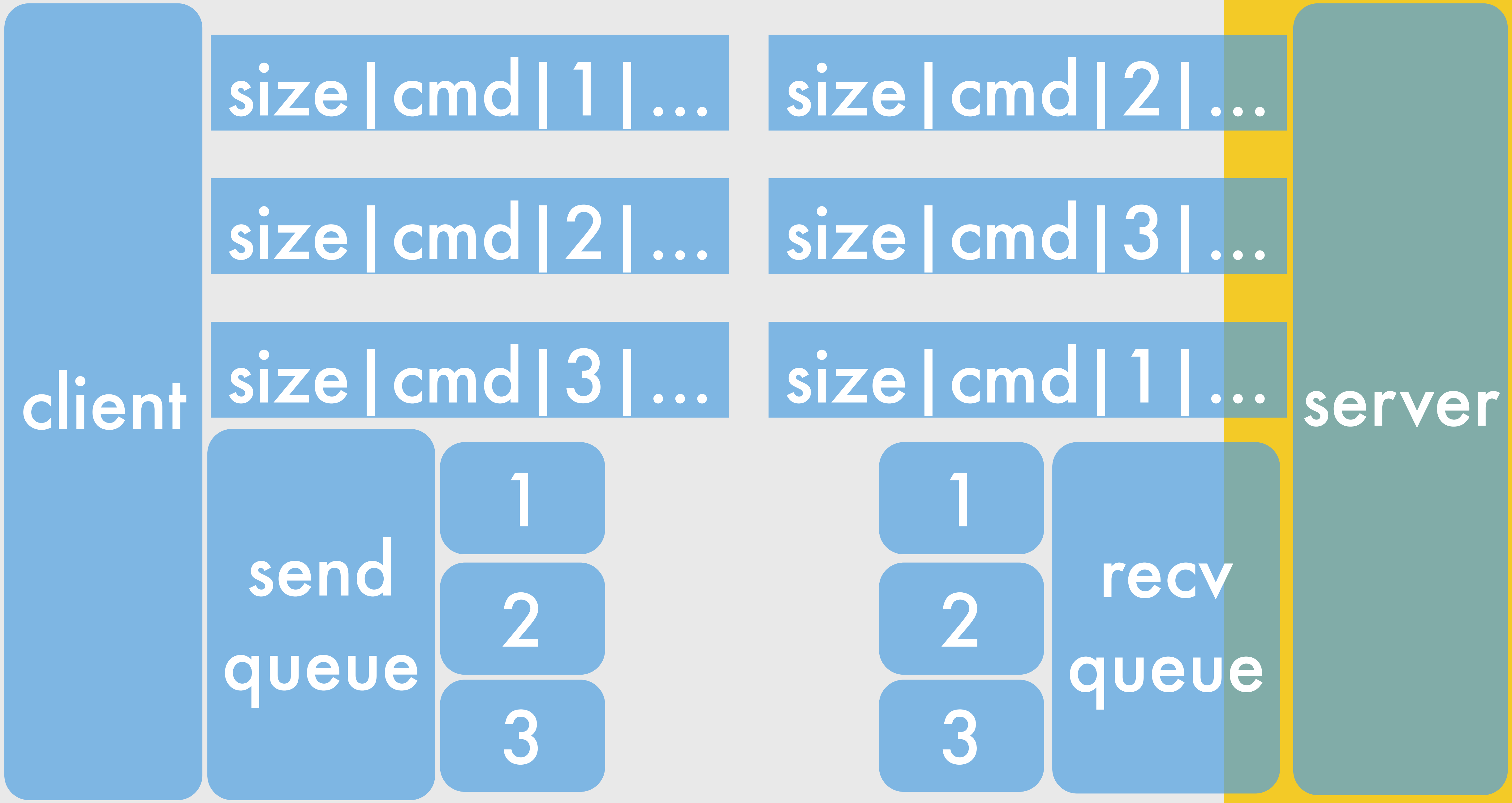
request分在不同的连接里

一个连接一个

长连接


tcp建立连接
[1]request
[2]request
[3]request
[2]response
[3]response
[1]response
client
server
request在一个连接里
request与response异步处理

长连接的处理细节

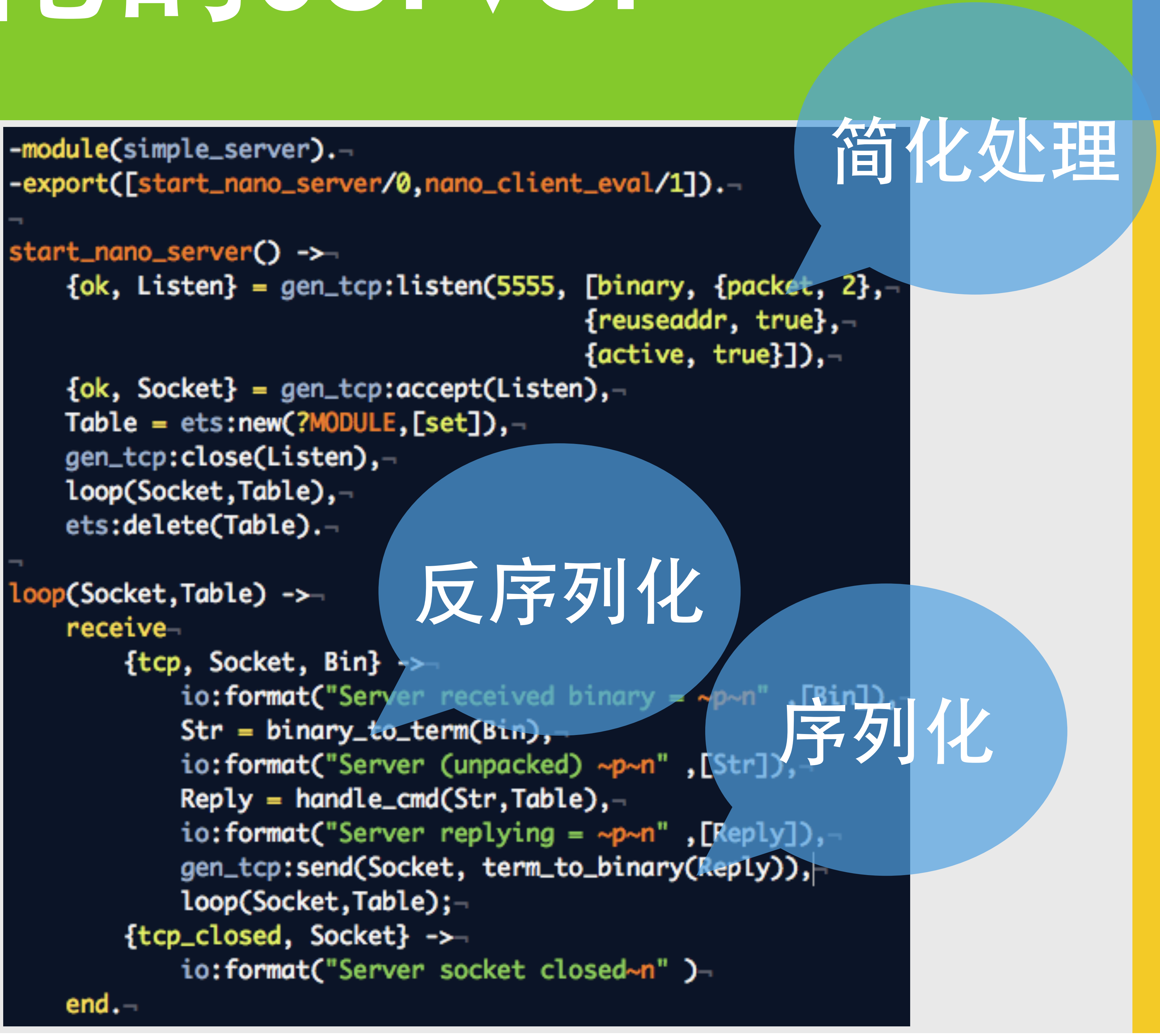

client
size | cmd | 1 | …
size | cmd | 2 | …
size | cmd | 3 | …
send queue
1
2
3
size | cmd | 2 | …
size | cmd | 3 | …
size | cmd | 1 | …
1
2
3
recv queue
server

消息组成

size	cmd	id	content
2	2	2	

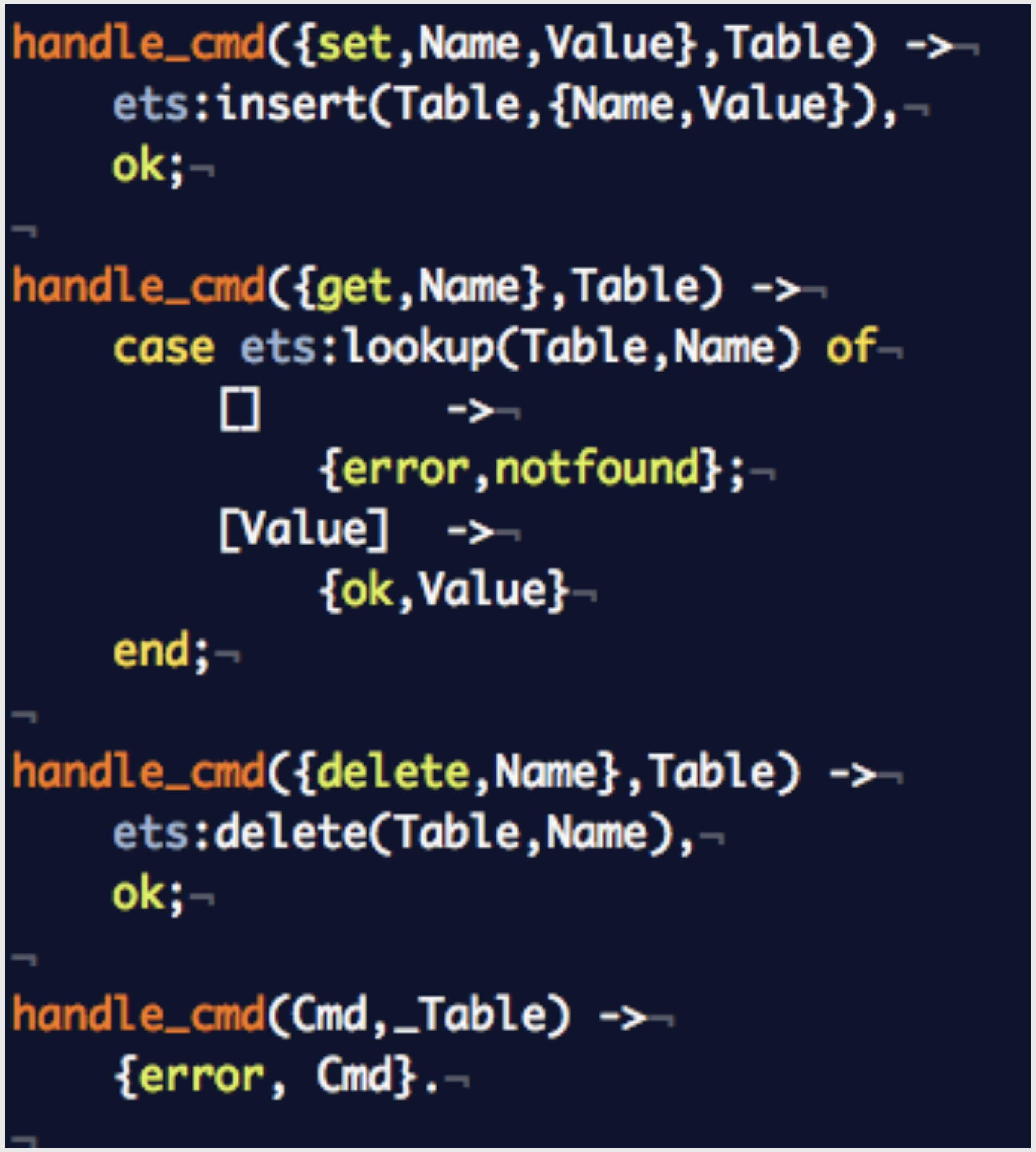
http://code.google.com/p/xbaytable/wiki/Specification

简化的server



```
-module(simple_server).
-export([start_nano_server/0,nano_client_eval/1]).

start_nano_server() ->
    {ok, Listen} = gen_tcp:listen(5555, [binary, {packet, 2},
                                         {reuseaddr, true},
                                         {active, true}]),
    {ok, Socket} = gen_tcp:accept(Listen),
    Table = ets:new(?MODULE,[set]),
    gen_tcp:close(Listen),
    loop(Socket,Table),
    ets:delete(Table).

loop(Socket,Table) ->
    receive
        {tcp, Socket, Bin} ->
            io:format("Server received binary = ~p~n" ,[Bin]),
            Str = binary_to_term(Bin),
            io:format("Server (unpacked) ~p~n" ,[Str]),
            Reply = handle_cmd(Str,Table),
            io:format("Server replying = ~p~n" ,[Reply]),
            gen_tcp:send(Socket, term_to_binary(Reply)),
            loop(Socket,Table);
        {tcp_closed, Socket} ->
            io:format("Server socket closed~n" )
    end.
```

简化的命令处理

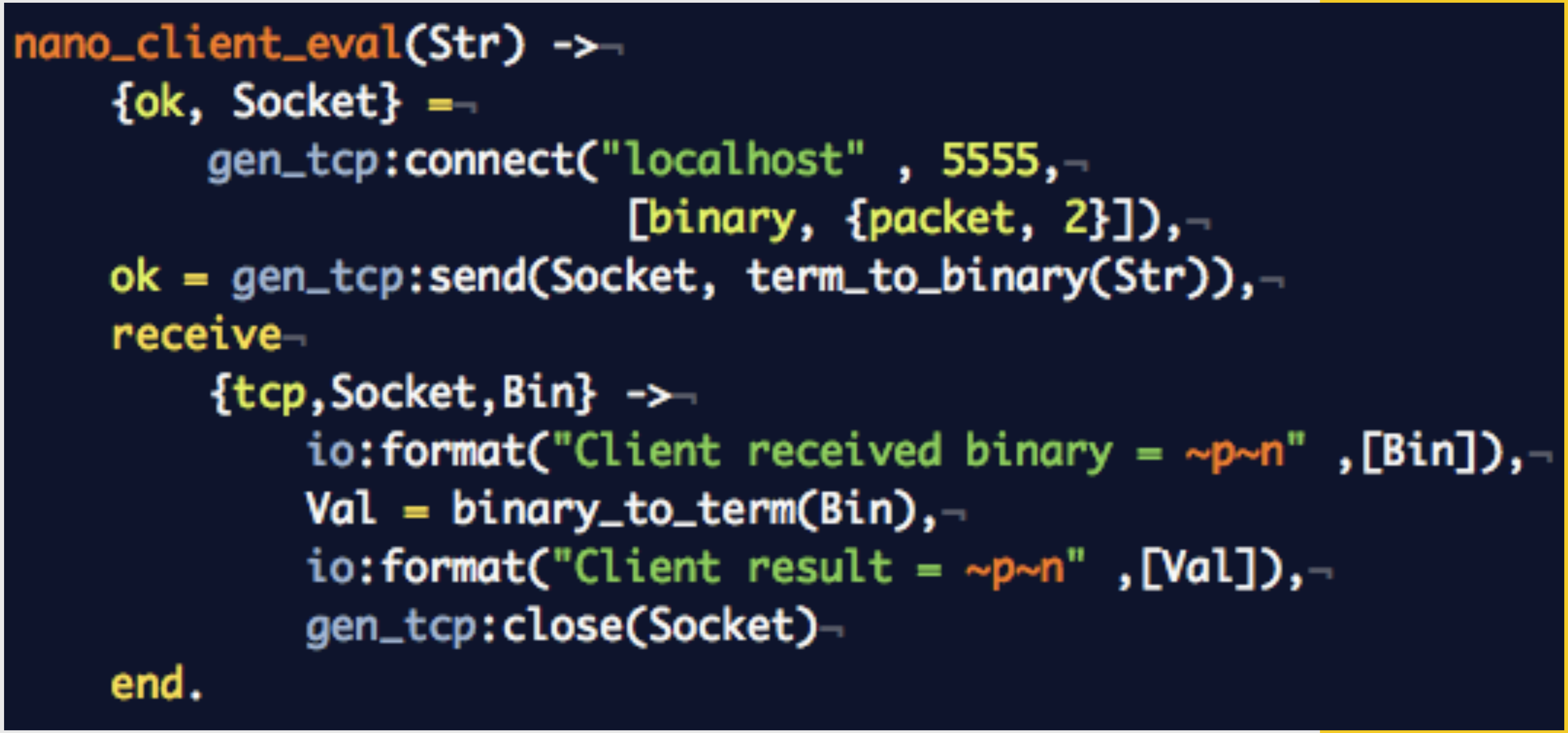


```
handle_cmd({set,Name,Value},Table) ->
    ets:insert(Table,{Name,Value}),
    ok;

handle_cmd({get,Name},Table) ->
    case ets:lookup(Table,Name) of
        []        ->
            {error,notfound};
        [Value]   ->
            {ok,Value}
    end;

handle_cmd({delete,Name},Table) ->
    ets:delete(Table,Name),
    ok;

handle_cmd(Cmd,_Table) ->
    {error, Cmd}.
```

尝试一个客户端

- 连接到 5555
- 发出我们需要的发出的指令
- 断开连接

测试客户端



```
nano_client_eval(Str) ->
    {ok, Socket} =
        gen_tcp:connect("localhost" , 5555,
                                     [binary, {packet, 2}]),
    ok = gen_tcp:send(Socket, term_to_binary(Str)),
    receive
        {tcp,Socket,Bin} ->
            io:format("Client received binary = ~p~n" ,[Bin]),
            Val = binary_to_term(Bin),
            io:format("Client result = ~p~n" ,[Val]),
            gen_tcp:close(Socket)
    end.
```

测试

```
[xiaohong@localhost test]$ erl
Erlang (BEAM) emulator version 5.6 [source] [async-threads:0] [hipe] [kernel-poll:false]

Eshell V5.6  (abort with ^G)
1> simple_server:start_nano_server().
```

```
[xiaohong@localhost test]$ erl
Erlang (BEAM) emulator version 5.6 [source] [async-threads:0] [hipe] [kernel-poll:false]

Eshell V5.6  (abort with ^G)
1> simple_server:nano_client_eval({set,"xiaohong","passwd"}).
Client received binary = <<131,100,0,2,111,107>>
Client result = ok
ok
```

```
[xiaohong@localhost test]$ erl
Erlang (BEAM) emulator version 5.6 [source] [async-threads:0] [hipe] [kernel-poll:false]

Eshell V5.6  (abort with ^G)
1> simple_server:start_nano_server().
Server received binary = <<131,104,3,100,0,3,115,101,116,107,0,8,120,105,97,
                           111,104,111,110,103,107,0,6,112,97,115,115,119,100>>
Server (unpacked) {set,"xiaohong","passwd"}
Server replying = ok
Server socket closed
true
```

两步两个脚印