**yum -y install net-tools：获取安装包网络命令**

**[root@localhost ~]# netstat -anp|grep 9502**

**tcp服务器**：

//创建Server对象，监听 127.0.0.1:9501端口

$serv **=** **new** **swoole\_server**("127.0.0.1", 9501);

//监听连接进入事件

/\*\*

\*$fd：客户端连接唯一标识

\*$from\_id：线程id

\*/

$serv**->**on('connect', **function** ($serv, $fd,$from\_id) {

echo "Client: Connect.\n";

});

//监听数据接收事件

$serv**->**on('receive', **function** ($serv, $fd, $from\_id, $data) {

$serv**->**send($fd, "Server: "**.**$data);

});

//监听连接关闭事件

$serv**->**on('close', **function** ($serv, $fd) {

echo "Client: Close.\n";

});

//启动服务器

$serv**->**start();

**启动服务：**[root@localhost server]# php tcp.php **，**卡住不动就启动了

**查看是否启动**

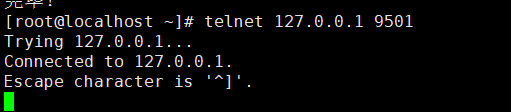
**方案1**：[root@localhost ~]# netstat -anp|grep 9501



**方案2：**

客户端连接

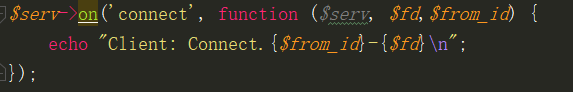
[root@localhost ~]# telnet 127.0.0.1 9501



服务端监听：//客户端连接上时

[root@localhost server]# php tcp.php

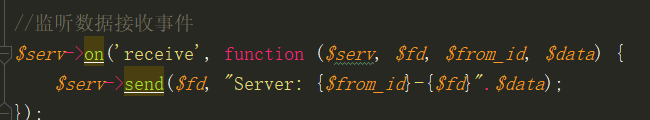
Client: Connect.0-1



**客户端输出内容：**

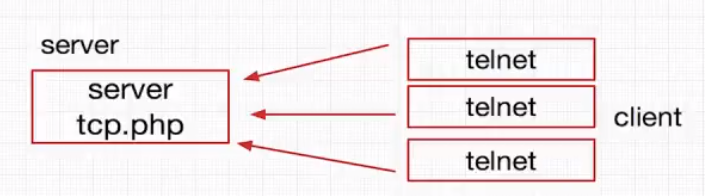
1

Server: 0-11//服务端监听到的

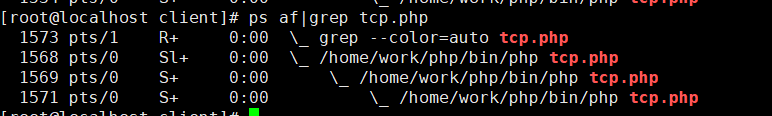


服务端关闭服务，Ctrl+c

客户端监听：Connection closed by foreign host.



查看当前跑的进程：[root@localhost client]# ps af|grep tcp.php



设置再大也不会超过可以使用数，单核是0

**tcp客户端**

<?php  
//连接tcp服务  
*$client* = new swoole\_client(*SWOOLE\_SOCK\_TCP*);  
if(!*$client*->connect('127.0.0.1',9501)){  
 echo "连接失败";  
 exit();  
}  
//php cli 常量  
*fwrite*(*STDOUT*,'请输入消息');  
*$msg* = *trim*(*fgets*(*STDIN*));  
//发送消息给tcp server  
*$client*->send(*$msg*);  
//接受server的数据  
*$result* = *$client*->recv();  
echo *$result*;

**UDP服务器：**

<?php  
//创建Server对象，监听 127.0.0.1:9502端口，类型为SWOOLE\_SOCK\_UDP  
*$serv* = new swoole\_server('127.0.0.1',9502,*SWOOLE\_PROCESS*,*SWOOLE\_SOCK\_UDP*);  
//监听数据接收事件  
*$serv*->on('Packet', function (*$serv*, *$data*, *$clientInfo*) {  
 *$serv*->sendto(*$clientInfo*['address'], *$clientInfo*['port'], "Server ".*$data*);  
 *var\_dump*(*$clientInfo*);  
});  
//启动服务器  
*$serv*->start();

进入目录运行

注意：UDP服务器与TCP服务器不同，UDP没有连接的概念。启动Server后，客户端无需Connect，直接可以向Server监听的9502端口发送数据包。对应的事件为onPacket

[root@localhost ~]# netstat -anp|grep 9502

udp 0 0 127.0.0.1:9502 0.0.0.0:\* 91329/php

**upd客户端测试**：

测试命令安装：[root@localhost ~]# yum -y install nmap-ncat.x86\_64 yum search 包

测试：[root@localhost ~]# nc -u 127.0.0.1 9502

输入发送的信息

web服务器：

<?php  
*$http* = new swoole\_http\_server("0.0.0.0", 9501);  
*$http*->on('request', function (*$request*, *$response*) {  
 *var\_dump*(*$request*->*get*, *$request*->*post*);  
 *$response*->header("Content-Type", "text/html; charset=utf-8");  
 *$response*->end("<h1>Hello Swoole. #".*rand*(1000, 9999)."</h1>");  
});  
*$http*->start();

进入目录运行

注意：Http服务器只需要关注请求响应即可，onRequest事件，触发此事件，回调函数有2个参数，一个是$request【GET/POST】，response，对request的响应可以通过操作response对象来完成。$response->end()方法表示输出一段HTML内容，并结束此请求

1、0.0.0.0 表示监听所有IP地址，一台服务器可能同时有多个IP，如127.0.0.1本地回环IP、192.168.1.100局域网IP、210.127.20.2 外网IP，这里也可以单独指定监听一个IP

2、9501 监听的端口，如果被占用程序会抛出致命错误，中断执行。

HTTP服务：

<http://127.0.0.1:9501>

Chrome 请求两次问题：会产生额外的一次请求，/favicon.ico，可以在代码中响应404错误

*$http*->on('request', function (*$request*, *$response*) {  
 if (*$request*->*server*['path\_info'] == '/favicon.ico' || *$request*->*server*['request\_uri'] == '/favicon.ico') {  
 return *$response*->end();  
 }  
 *var\_dump*(*$request*->*get*, *$request*->*post*);  
 *$response*->header("Content-Type", "text/html; charset=utf-8");  
 *$response*->end("<h1>Hello Swoole. #".*rand*(1000, 9999)."</h1>");  
});

URL 路由：应用程序可以根据$request->server['request\_uri']实现路由

*$http*->on('request', function (*$request*, *$response*) {  
 list(*$controller*, *$action*) = *explode*('/', *trim*(*$request*->*server*['request\_uri'], '/'));  
 //根据 $controller, $action 映射到不同的控制器类和方法  
 (new *$controller*)->*$action*(*$request*, *$response*);  
});



启动：[root@localhost server]# php http\_server.php 卡住启动成功

测试连接：

方案1：[root@localhost client]# curl 127.0.0.1:8811

输出内容：<h1>HTTPSERVER</h1>

方案2：域名访问输出内容

<?php  
*$http* = new swoole\_http\_server("0.0.0.0",8811);  
/\*\*  
 \* $request：url上的请求数据  
 \* $response：返回的响应数据  
 \*/  
*$http*->on('request',function (*$request*,*$response*){  
 //print\_r($\_GET);//获取不到数据  
 *print\_r*(*$request*->*get*);  
 *$response*->end("<h1>HTTPSERVER</h1>");//end只能调用一次，如果需要分多次向客户端发送数据，请使用write方法,打印到浏览器需要放到这里  
});  
*$http*->start();