服务器篇:

服务器端

HTcpServer

链接端(客户端)

HConnect

------------------------------------------------------------------------------------

HTcpServer ID=ip

std::map<ip,HConnect> list;存储所有连接

包含多个HConnect连接(每个连接HConnect ID=ip)

HNetApp

std::map<ip,HTcpServer\*> list; 存储所有服务器端

这样的优点就是从任何客户端过来的数据，可以解析后再从原来的链路上发回去。

报文采用统一显示窗口，所有报文都在同一个窗口显示

正常报文

发出报文格式收到报文用color1，发出报文用color2颜色

(2018-4-11 00:00:00)A网(心跳)>> 85 00 00 00 00

(2018-4-11 00:00:00)A网(全遥信)<<86 00 00 00…..

2018-4-11 00:00:00 from:198.120.0.111 <<

如果解析勾上(之前的报文是不能解析的，只能解析后面):

(2018-4-11 00:00:00)A网(心跳)>> 85(标识码:心跳) 00 00(报文长度) 00(类型) 00(监控)

错误报文(不解析红色)

2018-4-11 00:00:00)A网(错误报文)>> 85(标识码:心跳) 00 00(报文长度) 00(类型) 00(监控)

ui->textEdit->setHtml("这个是<span style=' color:#ff0000;'>红色</span>");

"\([\(]from[:0-9.]+[\)]\)"; from:198.120.0.111

"\\b[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}\\b"; 2018-00-00 19:00:23

"\([\(]from[:0-9.\u4e00-\u9fa5]+[\)]\)"; (from:198.120.0.111:心跳)

"2018-11-20 19:18:12 (from:198.120.0.111) (type:心跳) << 85 00 00 00 "; ----1

"[\u4e00-\u9fa5a-zA-Z0-9\_ :<-\(]+<<[0-9 ]+"; --对应上面的