



网络编程入门篇

利用 socket 实现 TCP 客户端

目录

- 基础知识
- 具体示例
- 示例代码讲解



基础知识

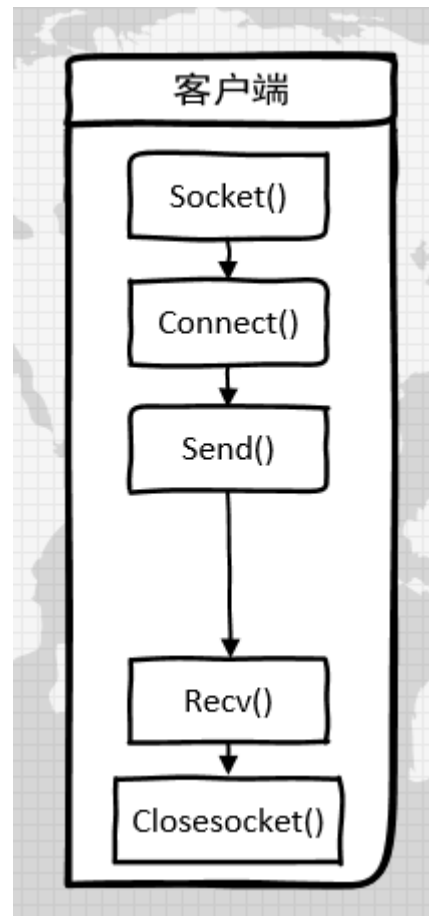
基础知识

- **socket** 编程一般采用**客户端-服务器**模式（即由客户进程向服务器进程发出请求，服务器进程执行请求的任务并将执行结果返回给客户进程的模式。）
- 今天我们要讲的就是如何利用 **socket** 编程实现基于 **TCP** 协议通信的客户端。
- 首先我们先向大家展示 **socket** 编程的流程，然后再给出具体的实例，最后讲解一下示例代码。

基础知识

TCP 客户端的 socket 编程流程如右图所示:

- 创建 socket
- 建立连接
- 通信
- 关闭 socket





具体示例

工程配置

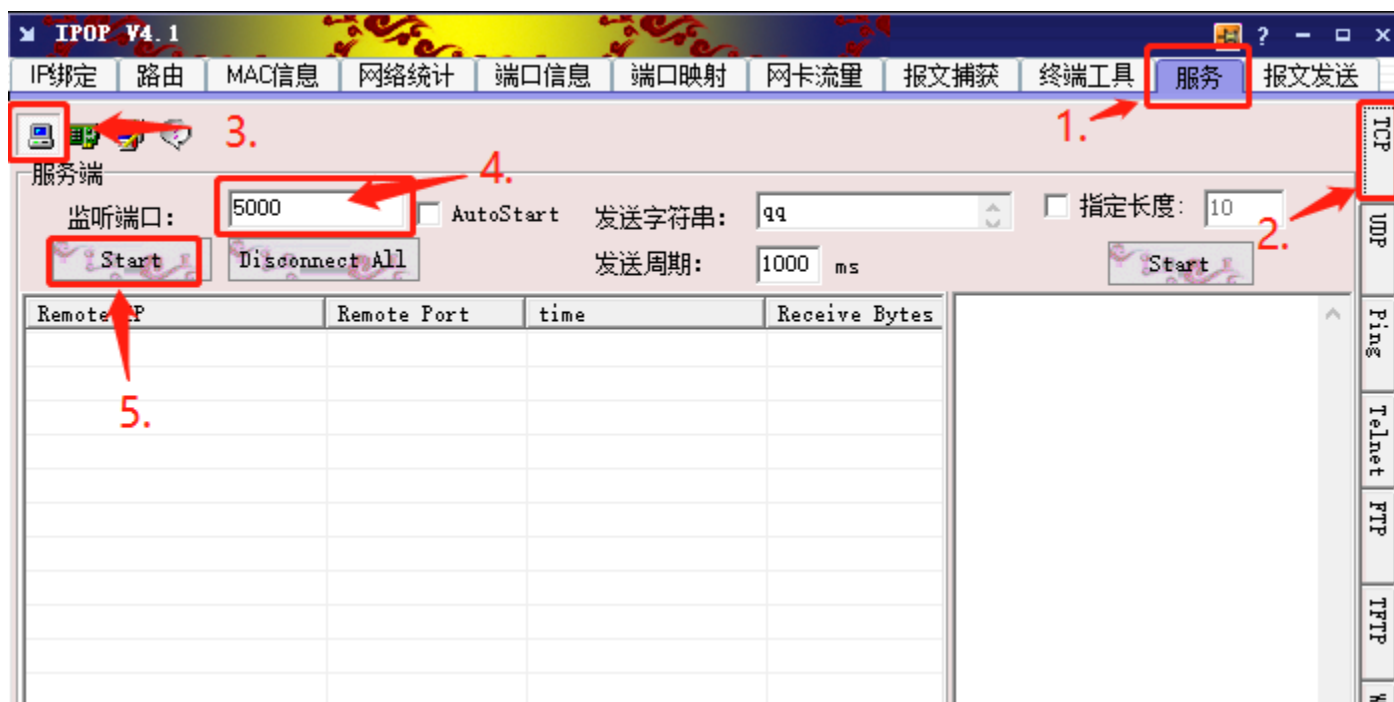
- RT-Thread samples 软件包中已有一份该示例代码 `tcpclient.c`，可以通过 `env` 配置将示例代码加入到项目中。
- 按照下面的路径即可开启 `tcp client` 的示例代码

```
RT-Thread online packages --->
  miscellaneous packages --->
    samples: RT-Thread kernel and components samples --->
      network sample options --->
        [*] [network] tcp client
```

- 保存并更新软件包 `pkgs --update`
- 编译工程 `scons`
- 然后运行 `qemu`

开启TCP服务器

- 在运行示例代码之前需要先在电脑上开启一个 TCP 服务器，这里以网络调试助手 IPOPOP 为例。



查看本机ip地址

- 在windows系统中打开命令提示符，输入ipconfig即可查看本机ip

```
管理员: 命令提示符
C:\WINDOWS\system32>
C:\WINDOWS\system32>ipconfig

Windows IP 配置

以太网适配器 以太网 2:

    连接特定的 DNS 后缀 . . . . . :
    IPv6 地址 . . . . . : 2001:470:d:aa9:a945:38e7:1ee2:1119
    临时 IPv6 地址. . . . . : 2001:470:d:aa9:14fb:d705:f25a:fb26
    本地链接 IPv6 地址. . . . . : fe80::a945:38e7:1ee2:1119%7
    IPv4 地址 . . . . . : 192.168.12.44
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : fe80::ce2d:e0ff:fe05:3b5a%7
                       192.168.10.1

以太网适配器 tap:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址. . . . . : fe80::d462:3cb4:193:1998%3
    IPv4 地址 . . . . . : 192.168.137.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . :
```

运行示例代码

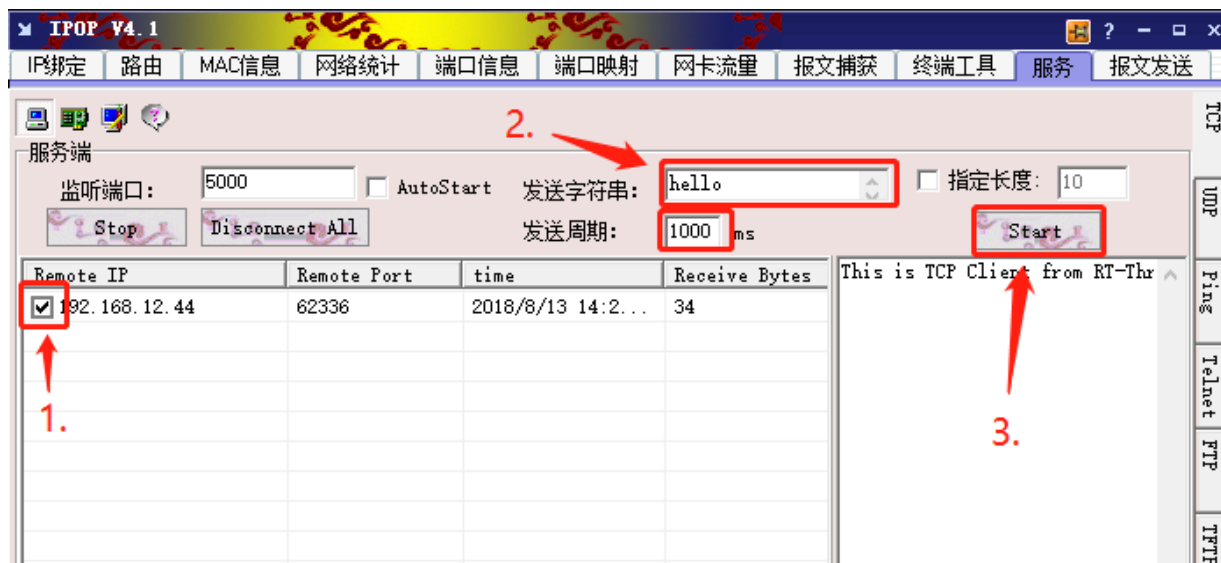
- 在qemu运行起来后，在 msh 命令行下输入下面的命令即可让示例代码运行。

```
msh> tcpclient 192.168.12.44 5000
```

- tcpclient 有两个参数 URL PORT
- 其中：
- URL 是目标服务器的网址或 IP 地址
- PORT 是目标服务器的端口号

```
msh />tcpclient 192.168.12.44 5000  
Received data = Welcome to TcpSrv  
█
```

从服务器发消息给客户端



发送字符 'q' 即可断开连接

```
Received data = hello
Received data = hello
Received data = hello
got a 'q' or 'Q',close the socket.
msh />
```

注意事项

- 电脑需要关闭防火墙



示例代码讲解

示例代码讲解

```
#include <rtthread.h>
#include <sys/socket.h> /* 使用BSD socket, 需要包含socket.h头文件 */
#include <netdb.h>
#include <string.h>
#define BUFSZ 1024

static const char send_data[] = "This is TCP Client from RT-Thread."; /* 发送用到的数据 */
void tcpclient(int argc, char **argv)
{
    int ret;
    char *recv_data;
    struct hostent *host;
    int sock, bytes_received;
    struct sockaddr_in server_addr;
    const char *url;
    int port;

    if (argc < 3)
    {
        rt_kprintf("Usage: tcpclient URL PORT\n");
        rt_kprintf("Like: tcpclient 192.168.12.44 5000\n");
        return ;
    }
}
```

示例代码讲解

```
url = argv[1];
port = strtoul(argv[2], 0, 10);

/* 通过函数入口参数url获得host地址（如果是域名，会做域名解析） */
host = gethostbyname(url);

/* 分配用于存放接收数据的缓冲 */
recv_data = rt_malloc(BUFSZ);
if (recv_data == RT_NULL)
{
    rt_kprintf("No memory\n");
    return;
}

/* 创建一个socket，类型是SOCKET_STREAM，TCP类型 */
if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == -1)
{
    /* 创建socket失败 */
    rt_kprintf("Socket error\n");

    /* 释放接收缓冲 */
    rt_free(recv_data);
    return;
}
```

示例代码讲解

```
/* 初始化预连接的服务端地址 */
server_addr.sin_family = AF_INET;
server_addr.sin_port = htons(port);
server_addr.sin_addr = *((struct in_addr *)host->h_addr);
rt_memset(&(server_addr.sin_zero), 0, sizeof(server_addr.sin_zero));

/* 连接到服务端 */
if (connect(sock, (struct sockaddr *)&server_addr, sizeof(struct sockaddr)) == -1)
{
    /* 连接失败 */
    rt_kprintf("Connect fail!\n");
    closesocket(sock);

    /* 释放接收缓冲 */
    rt_free(recv_data);
    return;
}

while (1)
{
    /* 从sock连接中接收最大BUFSZ - 1字节数据 */
    bytes_received = recv(sock, recv_data, BUFSZ - 1, 0);
    if (bytes_received < 0)
    {

```


示例代码讲解

```
/* 接收失败，关闭这个连接 */
closesocket(sock);
rt_kprintf("\nreceived error,close the socket.\r\n");

/* 释放接收缓冲 */
rt_free(recv_data);
break;
}
else if (bytes_received == 0)
{
    /* 默认 recv 为阻塞模式，此时收到0认为连接出错，关闭这个连接 */
    closesocket(sock);
    rt_kprintf("\nreceived error,close the socket.\r\n");

    /* 释放接收缓冲 */
    rt_free(recv_data);
    break;
}

/* 有接收到数据，把末端清零 */
recv_data[bytes_received] = '\0';
if (strncmp(recv_data, "q", 1) == 0 || strncmp(recv_data, "Q", 1) == 0)
{
```

示例代码讲解

```
/* 如果是首字母是q或Q, 关闭这个连接 */
closesocket(sock);
rt_kprintf("\n got a 'q' or 'Q',close the socket.\r\n");

/* 释放接收缓冲 */
rt_free(recv_data);
break;
}
else
{
    /* 在控制终端显示收到的数据 */
    rt_kprintf("\nReceived data = %s ", recv_data);
}

/* 发送数据到sock连接 */
ret = send(sock, send_data, strlen(send_data), 0);
if (ret < 0)
{
    /*发送失败, 关闭这个连接 */
    closesocket(sock);
    rt_kprintf("\nsend error,close the socket.\r\n");

    rt_free(recv_data);
    break;
}
```

示例代码讲解

```
    else if (ret == 0)
    {
        /* 打印send函数返回值为0的警告信息 */
        rt_kprintf("\n Send warning,send function return 0.\r\n");
    }
}
return;
}
MSH_CMD_EXPORT(tcpclient, a tcp client sample);
```