

1.2 tcp 客户端发送与接收数据_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 1.2 tcp 客户端发送与接收数据涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

- 基于 socket 发送数据需要调用 send 函数, 下面是 send 函数的具体信息

函数头文件 #include <sys/types.h>

#include <sys/socket.h>

函数原型

ssize_t send(int sockfd,const void *buf,size_t len,int flags)

函数功能 基于套接字（建立连接）发送数据

函数参数 sockfd : 套接字文件描述符

buf : 发送缓冲区的地址

len : 发送数据的长度

flags : 发送标志位

函数返回值 成功 : 返回 成功发送的字节数

失败 : 返回 -1, 并设置 errno

示例

客户端发送数据给服务器

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>

#include <sys/types.h>
#include <sys/socket.h>

int main(int argc,char *argv[])
{
    int sfd,ret;
    ssize_t sbytes = 0,rbytes = 0;
    char sbuffer[1024] = {0};
    char rbuffer[1024] = {0};

    struct sockaddr_in svr_addr;
    if (argc != 3){
        fprintf(stderr,"Usage : %s < ip > < port >.\n",argv[0]);
        return -1;
    }

    sfd = socket(AF_INET,SOCK_STREAM,0);
    if (sfd == -1){
        perror("[ERROR] socket(): ");
        exit(EXIT_FAILURE);
    }

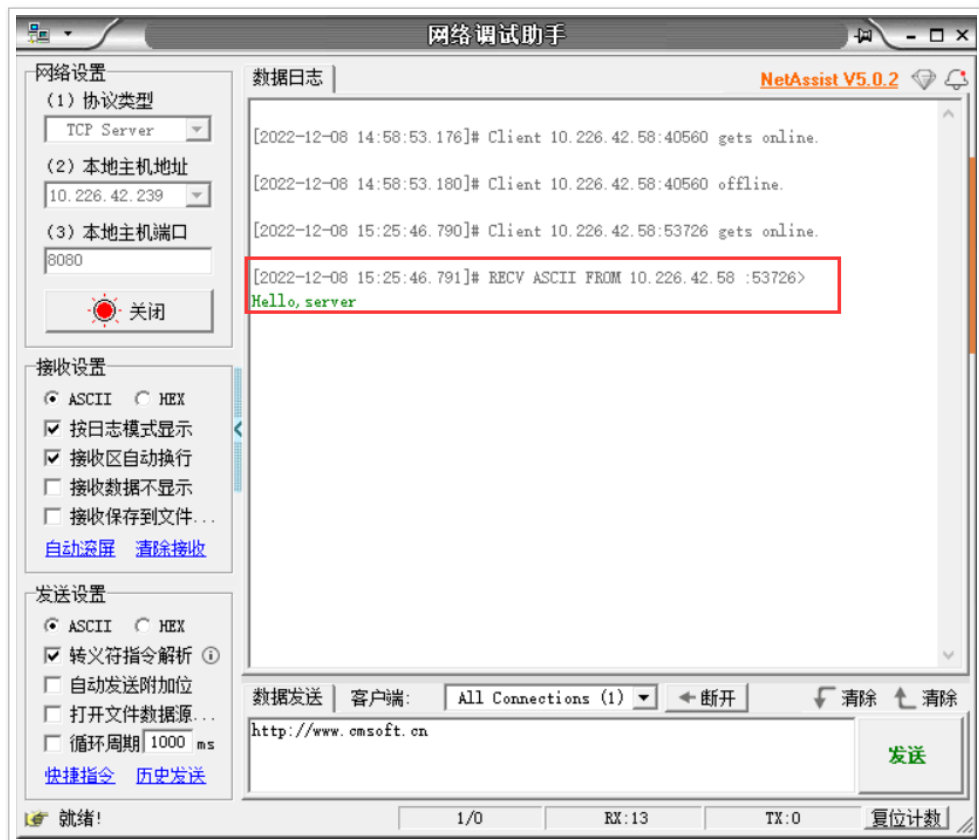
    printf("sfd = %d\n",sfd);
    bzero(&svr_addr,sizeof(svr_addr));
```

```
svr_addr.sin_family = AF_INET;
svr_addr.sin_port = htons(atoi(argv[2]));
svr_addr.sin_addr.s_addr = inet_addr(argv[1]);

ret = connect(sfd,(const struct sockaddr *)&svr_addr,sizeof(struct sockaddr));
if (ret == -1){
    perror("[ERROR] connect()");
    exit(EXIT_FAILURE);
}

strcpy(sbuffer,"Hello,server");
sbytes = send (sfd,sbuffer,strlen(sbuffer) + 1,0);
if (sbytes == -1){
    perror("[ERROR] send()");
    exit(EXIT_FAILURE);
}

close(fd);
}
```



- 基于 socket 接收数据需要调用 recv 函数, 具体信息如下:

函数头文件 #include <sys/types.h>

#include <sys/socket.h>

函数原型

ssize_t recv(int sockfd,void *buf,size_t len,int flags)

函数功能 基于套接字接收数据

函数参数 sockfd : 套接字文件描述符

buf : 接收缓冲区的地址

len : 接收数据最大长度

flags : 标志位

函数返回值 成功 : 返回 成功接收的字节数

失败：返回 -1, 并设置 errno

示例

服务器发送数据，客户端接收数据

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>

#include <sys/types.h>
#include <sys/socket.h>

int main(int argc, char *argv[])
{
    int sfd, ret;
    ssize_t sbytes = 0, rbytes = 0;
    char sbuffer[1024] = {0};
    char rbuffer[1024] = {0};

    struct sockaddr_in svr_addr;
    if (argc != 3){
        fprintf(stderr, "Usage : %s < ip > < port >.\n", argv[0]);
        return -1;
    }

    sfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sfd == -1){
        perror("[ERROR] socket(): ");
        exit(EXIT_FAILURE);
    }

    printf("sfd = %d\n", sfd);

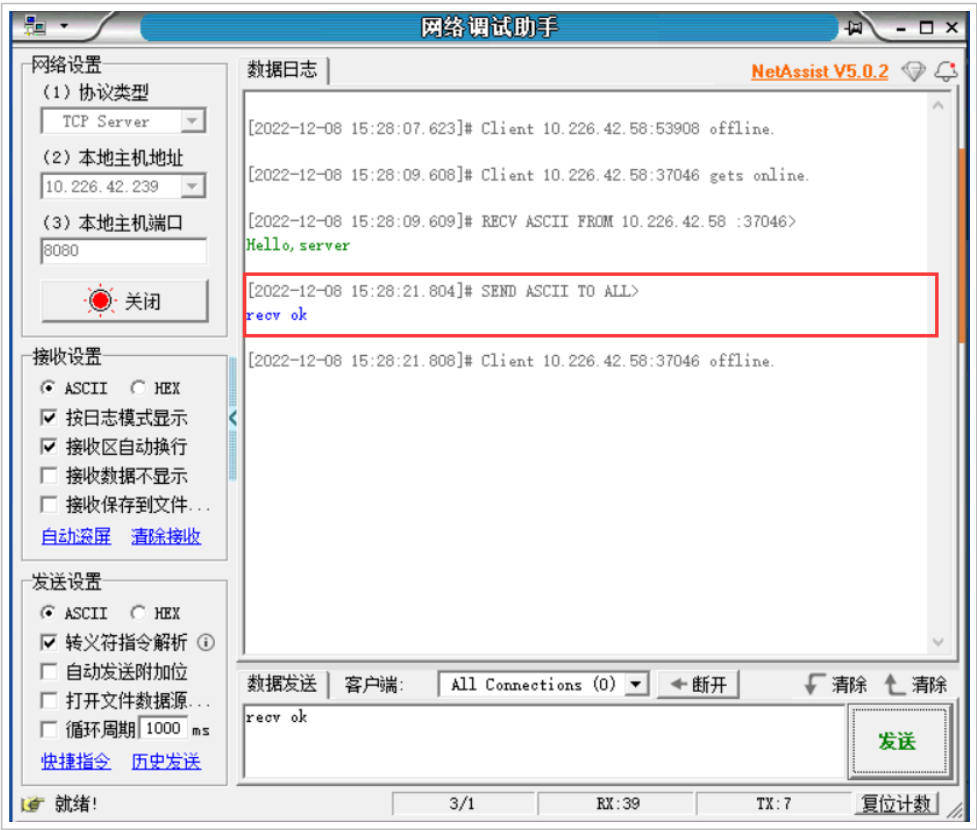
    bzero(&svr_addr, sizeof(svr_addr));
    svr_addr.sin_family = AF_INET;
    svr_addr.sin_port = htons(atoi(argv[2]));
    svr_addr.sin_addr.s_addr = inet_addr(argv[1]);

    ret = connect(sfd, (const struct sockaddr *)&svr_addr, sizeof(struct sockaddr));
    if (ret == -1){
        perror("[ERROR] connect():");
        exit(EXIT_FAILURE);
    }

    strcpy(sbuffer, "Hello, server");
    sbytes = send (sfd, sbuffer, strlen(sbuffer) + 1, 0);
    if (sbytes == -1){
        perror("[ERROR] send(): ");
        exit(EXIT_FAILURE);
    }

    rbytes = recv(sfd, rbuffer, sizeof(rbuffer), 0);
    if (rbytes == -1){
        perror("[ERROR] recv():");
        exit(EXIT_FAILURE);
    }else if (rbytes == 0){
        printf("server is offline.\n");
    }else if (rbytes > 0){
        printf("recv buffer : %s\n", rbuffer);
    }

    close(sfd);
    return 0;
}
```



```
ben@ubuntu:~/class/week15/codes/part1$ ./a.out 10.226.42.239 8080
sfd = 3
server is offline.
ben@ubuntu:~/class/week15/codes/part1$
```

- 设计一个客户端程序，从键盘接收用户输入后，输入 quit 退出客户端程序，将信息发送给服务器，使用调试助手作为 tcp 服务器，来进行接收

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

