10.2 局域网聊天室之群发服务器设置\_物联网/嵌入式工程师-慕课网 2023/2/10 17:10 纹程 Ξ 免费课 实战课 体系课 慕课教程 专栏 手记 企业服务 我自 Q 索引目录♣ - .概述 底层核心技 从所有教程的词条中查询· ☑动高级1 ..实现思路 三.代码总体思路 首页 > 慕课教程 > 物联网/嵌入式工程师 > 10.2 局域网聊天室之群发服务器设置 底层核心技 弘高级2 大白老师・更新于 2022-12-21 ◆ 上一节 10.1 局域网聊天... 10.3 局域网聊天... 下一节 ▶ **人工智能必** 一.概述 • 说明

**居项目实战之** 

**居项目实战之** 

**居项目实战之** 

'嵌入式项目

需求

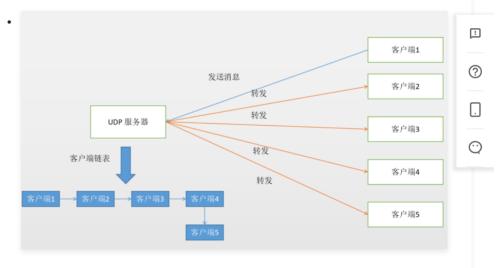
• 项目需求

局域网群发服务器的功能,后期在对代码的架构进行扩展

• 服务端: 实现一个服务器群发的功能,服务端设计一个链表,用来存储用户信息,当链表中 某个用户发送数据的时候,链表中其他用户也可以收到。

局域网聊天室整体代码量较多。客户端和服务器相关代码都比较多。这里我们根据需求先实现一个

• 客户端: 发送数据和接收数据应该是异步的。双方互不影响



# 二.实现思路

- 1. 整体想法
  - 1. 客户端
    - 创建多线程, 把客户端的收发分开。主线程发送数据, 子线程接收数据
  - 2. 服务器
    - 创建1个链表
    - 链表中保存用户的ip + port。

╱ 意见反馈 ♡ 收藏教程 □ 标记书签 2023/2/10 17:10 纹程 ≔ 底层核心技 ☑动高级1 底层核心技 ☑动高级2 **人工智能必 居项目实战之 居项目实战之 居项目实战之** '嵌入式项目

2. 部门代码思路

1. 客户端多线程设计

```
一.概述
<> 代码块
                                                                           :现思路
                                                                        三.代码总体思路
    void *recv_message(void *arg)
1
     {
3
     ;
4
     }
5
6
    //创建子线程接收
    pthread_create(&tid,NULL,recv_message,&sockfd);
7
8
9
    //主线程发送
    while(1)
10
11
         //发送数据
12
13
         sendto();
14
```

索引目录

- 3. 服务器设计
  - 链表节点设计

```
<> 代码块
     typedef struct sockaddr_in datatype_t;
2
3
     typedef struct node
4
                                                                                 ⊡
5
         datatype_t data;
         struct node *next;
6
                                                                                 ?
     }linknode_t;
                                                                                 \Box
```

• 服务器遍历链表发送数据

♡ 收藏教程

```
<> 代码块
                                                                         0
     //服务端收到消息
     recvfrom(sockfd,buf,sizeof(buf),0,&peer_addr,&addrlen);
 3
 4
     //判断该地址及端口是否在链表中已经存放了,不存在则把该地址
5
     //填充到链表中去
     insert_data_linklist(head,peer_addr);
 6
7
8
     //遍历链表,把消息发送给每一个客户端
9
     int find_linklist(linknode_t*head,datatype_t data)
10
11
        struct node *p;
12
13
        for(p = head->next; p != NULL; p = p->next)
14
15
            //结构体类型数据不能直接比较
            //if(p->data == data)
16
            //
                     return 1;
17
18
            if(memcmp(&(p->data),&data,sizeof(datatype_t)) == 0)
19
                   return 1;
20
21
22
        }
23
         return 0;
24
                                                                      不
```

□ 标记书签

教程 :=

底层核心技 ☑动高级1

底层核心技 ☑动高级2

**人工智能必** 

**居项目实战之** 

**居项目实战之** 

**居项目实战之** 

'嵌入式项目

memcmp函数介绍

索引目录

• memcmp简介,类似于strncmp。但是strncmp只能比较字符串。但是memcmp 函数能够按照内存字节来比较。

二.实现思路

⊡

?

 $\Box$ 

0

```
:码总体思路
<> 代码块
1
    #include <string.h>
2
3
    void *memcmp(const void *s1,const void *s2,size_t n);
    功能: memcmp是比较s1和s2所指向的内存前nbyte是否相同。(逐个字节的比较)
 4
 5
          第一个比较字符数组的收地址
6
    @s2
         第二个比较字符数组的收地址
         比较前n个byte
7
    @n
8
    返回值:
    s1 > s2 , 返回1
    s1 == s2, 返回 0
10
    s1 < s2, 则返回 -1
11
```

# 三.代码总体思路

• 【bug】说明

由于此代码为项目的中间架构,此架构目前存在bug。所有登录的用户,需要先发送一个消息,用于接收数据,保持其IP地址和端口,之后才能收到数据。

- 代码设计
  - 服务端代码
  - · linklist.h

```
<> 代码块
     #ifndef __LINKLIST_H__
     #define __LINKLIST_H__
 3
     #include <stdio.h>
 4
     #include <string.h>
     #include <stdlib.h>
 6
     #include <sys/socket.h>
7
     #include <netinet/in.h>
9
     #include <arpa/inet.h>
10
11
     typedef struct sockaddr_in datatype_t;
     typedef struct node
13
14
15
         datatype_t data; //数据域保存有效数据
16
         struct node *next; //指针域保存下一个结点的地址
     }linknode_t;
17
18
     extern linknode_t *create_empty_linklist();
19
     extern void insert_head_linklist(linknode_t *head,datatype_t data);
20
     extern int find_linklist(linknode_t *head,datatype_t *data);
     extern void broadcast_message(int sockfd,linknode_t *head,char *msg,int ms
23
     #endif
```

linklist.c

不

教程 :Ξ

底层核心技 ☑动高级1

茋层核心技 〖动高级2

**人工智能必** 

**居项目实战之** 

**居项目实战之** 

**居项目实战之** 

′嵌入式项目

∮阶段−三大

```
1
2
     #include "linklist.h"
3
4
     //1.创建空的链表---为头结点在堆区分配空间
5
     linknode t *create empty linklist()
6
7
         linknode_t *head = NULL;
8
9
         //1.1 分配堆区空间
10
         head = (linknode_t *)malloc(sizeof(linknode_t));
11
12
         if(NULL == head)
13
         {
14
            printf("malloc is fail!\n");
15
            return NULL;
16
         }
17
18
         memset(head,0,sizeof(linknode_t));
19
         //head->next = NULL;
20
         //head->data = 0;
21
         return head;
22
23
24
     //2.头插法:每次都在头结点后插入数据。
25
     //特点:插入的顺序和输出的顺序是相反的。
26
27
     void insert_head_linklist(linknode_t *head,datatype_t data)
28
29
         //2.1 为结点在堆区申请空间
30
         linknode_t *temp = (linknode_t *)malloc(sizeof(linknode_t));
31
         if(NULL == temp)
32
33
            printf("malloc is fail!\n");
34
            return ;
35
         }
36
37
         //2.2 插入数据
38
         temp->data = data;
39
40
         //2.3 连接结点
41
         temp->next = head->next;
42
         head->next = temp;
43
44
         return :
45
     }
46
47
     //在链表种查找该客户端是否已经存在用户信息,存放返回1,否则返回0
48
     int find_linklist(linknode_t *head,datatype_t *data)
49
50
         linknode_t *p = head;
51
52
         while(p->next != NULL)
53
         {
54
            //说明链表中有该用户信息, 老用户
55
            if(memcmp(\&(p->data),data,sizeof(datatype_t)) == 0)
56
                return 1;
57
58
            p = p->next;
59
60
         //若是循环结束,上面的memcmp都没有执行,说明是新用户,返回0
61
         return 0;
62
     }
63
 ╱ 意见反馈
                 ♡ 收藏教程
                                 □ 标记书签
```

索引目录

一.概述

二.实现思路

三.代码总体思路

⊡

?

 $\Box$ 

### 10.2 局域网聊天室之群发服务器设置\_物联网/嵌入式工程师-慕课网

教程 :☰

底层核心技 ☑动高级1

底层核心技 ☑动高级2

**人工智能必** 

**居项目实战之** 

**居项目实战之** 

**居项目实战之** 

′嵌入式项目

∮阶段−三大

```
void broadcast_message(int sockfd,linknode_t *head,char *msg,int msg_len)
65
66
                                                                          索引目录
67
         linknode_t *p = head;
68
                                                                              --.概述
69
         while(p->next != NULL)
                                                                              二.实现思路
70
             sendto(sockfd,msg,msg\_len,0,(struct sockaddr *)(&(p->next->data))代码总体思路
71
72
             p = p->next;
73
         }
74
         return ;
     }
```

udp\_server.c

```
1
     #include <stdio.h>
3
                                    /* See NOTES */
     #include <sys/types.h>
     #include <sys/socket.h>
     #include <netinet/in.h>
 6
     #include <arpa/inet.h>
7
     #include <string.h>
8
     #include <stdlib.h>
9
     #include <unistd.h>
10
     #include "linklist.h"
11
12
     void recv_data(int sockfd,linknode_t *head)
13
14
         int n = 0;
15
         char buf[1024] = {0};
         struct sockaddr_in client_addr;
17
         int len = sizeof(client addr);
18
19
         while(1)
20
         {
21
             memset(buf,0,sizeof(buf));
22
             n = recvfrom(sockfd,buf,sizeof(buf),0,(struct sockaddr *)&client_
23
             if(n < 0)
24
25
                 perror("Fail to recvfrom");
26
                 exit(EXIT_FAILURE);
             }
28
29
             //若是新用户, ip + port可以到链表中
             //若是链表中存在该用户信息,返回1
31
             //否则,返回0
32
             if(!find_linklist(head,&client_addr))
33
             {
                 insert_head_linklist(head,client_addr);
35
             }
36
37
             //把接收到的消息,发送給所有的客户端
38
             broadcast_message(sockfd,head,buf,n);
39
         }
40
         return ;
41
     }
42
43
     int init_socket(const char *ip,const char *port)
44
45
         int sockfd;
46
         struct sockaddr_in my_addr;
47
         int len = sizeof(my_addr);
                                                                                不
```

□ 标记书签

╱ 意见反馈

♡ 收藏教程

⊡

②

 $\Box$ 

#### 纹程 Ξ

底层核心技 ☑动高级1

底层核心技 ☑动高级2

**人工智能必** 

**居项目实战之** 

**居项目实战之** 

**居项目实战之** 

'嵌入式项目

```
sockfd = socket(AF_INET,SOCK_DGRAM,0);
50
         if(sockfd < 0)
51
                                                                          索引目录
         {
52
             perror("Fail to socket!");
                                                                             一.概述
54
             return -1;
                                                                             二.实现思路
         }
55
                                                                             三.代码总体思路
56
         //2.填充服务器自己的ip + port
         memset(&my_addr,0,sizeof(my_addr));
5.8
         my_addr.sin_family = AF_INET;
59
         my_addr.sin_port = htons(atoi(port));
         my_addr.sin_addr.s_addr = inet_addr(ip);
61
62
         //3.把ip + port与socket绑定
63
64
         if(bind(sockfd,(struct sockaddr *)&my_addr,len) < 0)</pre>
65
             perror("Fail to bind");
66
67
             return -1;
         }
68
69
         printf("wait recv data!\n");
70
72
         return sockfd;
     }
73
74
75
     //./a.out ip port
     int main(int argc, const char *argv[])
76
77
78
         int sockfd;
79
         struct sockaddr_in my_addr;
         int len = sizeof(my_addr);
80
81
         linknode_t *head = NULL;
         if(argc != 3)
82
         {
83
             fprintf(stderr,"Usage : %s ip port!\n",argv[0]);
84
             exit(EXIT_FAILURE);
86
87
         //1. 初始化udp操作,获得套接字
89
         sockfd = init_socket(argv[1],argv[2]);
90
         //2.创建空的链表
91
         head = create_empty_linklist();
93
         //3.接收数据
94
95
         recv_data(sockfd,head);
96
         //4.关闭文件描述符
97
98
         close(sockfd);
         return 0;
```

Makefije

```
<> 代码块
     .PHONY : clean
1
3
    CC := gcc
    INCLUDE_DIR := -I .
4
    OBJ := linklist.o udp_server.o
```

□ 标记书签

♡ 收藏教程

不

⊡

?

 $\Box$ 

教程 :=

底层核心技 〖动高级1

茋层核心技 ☑动高级2

**人工智能必** 

**居项目实战之** 

**居项目实战之** 

**居项目实战之** 

′嵌入式项目

∮阶段−三大

```
9
     $(TARGET):$(OBJ)
10
                                                                       索引目录
11
         gcc $^ -o $@
                                                                          一.概述
     $(OBJ) : %.o : %.c
13
                                                                          二.实现思路
         $(CC) -c $< -o $@
14
                                                                          三.代码总体思路
15
16
     clean:
        rm -rf *.o server
17
```

- 客户端代码
- client.c

```
1
    #include <stdio.h>
    #include <sys/types.h>
                                   /* See NOTES */
    #include <sys/socket.h>
    #include <netinet/in.h>
    #include <arpa/inet.h>
    #include <string.h>
    #include <stdlib.h>
    #include <unistd.h>
10
    #include <pthread.h>
11
12
    void *recv_message(void *arg)
13
14
        int sockfd = *(int *)arg;
15
16
        int n = 0;
17
        char buf[1024] = {0};
18
        struct sockaddr_in client_addr;
19
        int len = sizeof(client_addr);
20
21
        while(1)
22
23
            memset(buf,0,sizeof(buf));
24
            //接收服务器转发的消息
25
            n = recvfrom(sockfd,buf,sizeof(buf),0,(struct sockaddr *)&client_a
26
            if(n < 0)
27
            {
28
                perror("Fail to recvfrom");
29
                exit(EXIT_FAILURE);
30
31
32
            if(strncmp(buf,"quit",4) == 0)
33
                break;
34
35
            printf("recv %d bytes : %s\n",n,buf);
36
37
38
        exit(EXIT_SUCCESS);
39
40
41
    void send_data(int sockfd,struct sockaddr_in *addr,int len)
42
43
        int n = 0;
44
        char buf[1024] = {0};
45
46
        while(1)
47
        {
48
            putchar('>');
49
                                                                                不
                 ot(buf A circof(buf)).
                 ♡ 收藏教程
                                  □ 标记书签
```

⊡

?

 $\Box$ 

纹程 Ξ

底层核心技 ☑动高级1

底层核心技 ☑动高级2

**人工智能必** 

**居项目实战之** 

**居项目实战之** 

**居项目实战之** 

'嵌入式项目

```
buf[strlen(buf) - 1] = '\0'; //'\n'--->'\0'
 51
 52
                                                                          索引目录
             if(strncmp(buf,"quit",4) == 0)
 53
                 break;
                                                                             一.概述
55
             n = sendto(sockfd,buf,strlen(buf),0,(struct sockaddr *)addr,len);
 56
                                                                             三.代码总体思路
             if(n < 0)
 57
 58
                  perror("Fail to sendto"):
 59
                 exit(EXIT_FAILURE);
 60
             }
 61
         }
 62
         return ;
 63
 64
 65
     //./a.out ip port
 66
     int main(int argc, const char *argv[])
 67
 68
 69
         int sockfd;
         struct sockaddr_in peer_addr;
70
         int len = sizeof(peer_addr);
 71
         int ret = 0;
         pthread_t tid;
 73
         if(argc != 3)
 74
 75
         {
             fprintf(stderr,"Usage : %s ip port!\n",argv[0]);
76
             exit(EXIT_FAILURE);
 77
 78
         }
 79
         //1.通过socket创建文件描述符
 80
         sockfd = socket(AF_INET,SOCK_DGRAM,0);
 81
 82
         if(sockfd < 0)
 83
             perror("Fail to socket!");
 84
 85
             return -1;
 86
 87
         //2.填充服务器的ip + port
 88
 89
         memset(&peer_addr,0,sizeof(peer_addr));
         peer_addr.sin_family = AF_INET;
 90
         peer addr.sin port = htons(atoi(argv[2]));
 91
         peer_addr.sin_addr.s_addr = inet_addr(argv[1]);
 92
         //3. 创建接收线程,接收数据
 94
         ret = pthread_create(&tid,NULL,recv_message,(void *)&sockfd);
 95
 96
         if(ret != 0)
 97
         {
             fprintf(stderr,"Fail to pthread_create : %s\n",strerror(ret));
98
             exit(EXIT_FAILURE);
99
100
101
         //3.发送数据
102
103
         send_data(sockfd,&peer_addr,len);
104
         //4.关闭文件描述符
105
106
         close(sockfd);
107
         return 0;
     }
```

10.1 局域网聊天室项目概述 ◀ 上一节

下一节 ▶ 10.3 局域网聊天室之整体功能

不

⊡

?

 $\Box$ 

# 10.2 局域网聊天室之群发服务器设置\_物联网/嵌入式工程师-慕课网

▶ 我要提出意见反馈 ∷≡ 索引目录 一.概述 底层核心技 公司高级1 二.实现思路 企业服务 网站地图 网站首页 关于我们 联系我们 讲师招募 帮助中心 意见反馈 代码托管 Copyright © 2023 imooc.com All Rights Reserved | 京ICP备 12003892号-11 京公网安备11010802030151号 **人工智能必 居项目实战之 居项目实战之 居项目实战之** ′嵌入式项目 ? 0

 $\overline{\uparrow}$ 

╱ 意见反馈

♡ 收藏教程

□ 标记书签