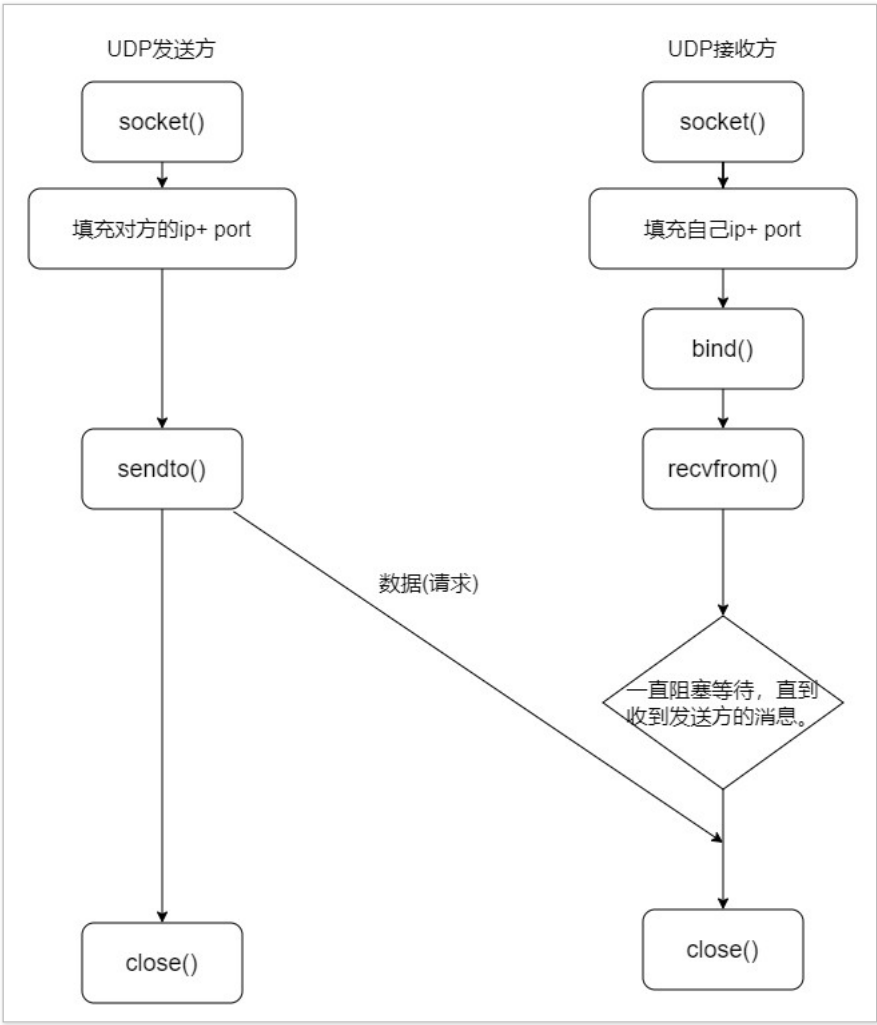


8.2 UDP 服务器代码实战_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 8.2 UDP 服务器代码实战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。



udp_recv.c

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

void recv_data(int sockfd)
{
    int n = 0;
    char buf[1024] = {0};
    struct sockaddr_in client_addr;
    socklen_t len = sizeof(client_addr);

    while(1)
    {
        memset(buf,0,sizeof(buf));
```

```

n = recvfrom(sockfd,buf,sizeof(buf),0,(struct sockaddr *)&client_addr,&len);
if(n < 0)
{
    perror("Fail to recvfrom");
    exit(EXIT_FAILURE);
}

printf("=====\n");
printf("Recv from IP : %s\n",inet_ntoa(client_addr.sin_addr));
printf("Recv fromt port : %d\n", ntohs(client_addr.sin_port));
printf("Recv %d bytes : %s\n",n,buf);

if(strncmp(buf,"quit",4) == 0)
    break;
}
return ;
}

int main(int argc, const char *argv[])
{
    int sockfd = 0;
    struct sockaddr_in my_addr;
    socklen_t len = sizeof(my_addr);
    if(argc < 3)
    {
        fprintf(stderr,"Usage : %s ip port!\n",argv[0]);
        exit(EXIT_FAILURE);
    }

    sockfd = socket(AF_INET,SOCK_DGRAM ,0);
    if(sockfd < 0)
    {
        perror("Fail to socket!");
        exit(EXIT_FAILURE);
    }

    memset(&my_addr,0,sizeof(my_addr));
    my_addr.sin_family = AF_INET;
    my_addr.sin_port = htons(atoi(argv[2]));
    my_addr.sin_addr.s_addr = inet_addr(argv[1]);

    if(bind(sockfd,(struct sockaddr *)&my_addr,len) < 0)
    {
        perror("Fail to bind");
        return -1;
    }

    recv_data(sockfd);

    close(sockfd);

    return 0;
}

```

udp_send.c

```

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

void send_data(int sockfd,struct sockaddr_in *addr,int len)
{
    int n = 0;
    char buf[1024] = {0};

    while(1)
    {
        putchar('>');
        memset(buf,0,sizeof(buf));
        fgets(buf,sizeof(buf),stdin);
        buf[strlen(buf) - 1] = '\0';

        n = sendto(sockfd,buf,strlen(buf),0,(struct sockaddr *)addr,len);
    }
}

```

```

        if(n < 0)
        {
            perror("Fail to sendto");
            exit(EXIT_FAILURE);
        }

        if(strncmp(buf,"quit",4) == 0)
            break;
    }
    return ;
}

int main(int argc, const char *argv[])
{
    int sockfd = 0;
    struct sockaddr_in peer_addr;
    socklen_t len = sizeof(peer_addr);
    if(argc < 3)
    {
        fprintf(stderr,"Usage : %s ip port!\n",argv[0]);
        exit(EXIT_FAILURE);
    }

    sockfd = socket(AF_INET,SOCK_DGRAM ,0);
    if(sockfd < 0)
    {
        perror("Fail to socket!");
        exit(EXIT_FAILURE);
    }

    memset(&peer_addr,0,sizeof(peer_addr));
    peer_addr.sin_family = AF_INET;
    peer_addr.sin_port = htons(atoi(argv[2]));
    peer_addr.sin_addr.s_addr = inet_addr(argv[1]);

    send_data(sockfd,&peer_addr,len);

    close(sockfd);

    return 0;
}

```

• 测试运行

- 开启两个不同的终端，A 终端执行命令
 - linux@linux:~/Class/network/udp\$ gcc udp_send.c -o send
 - linux@linux:~/Class/network/udp\$ gcc udp_send.c -o recv
 - linux@linux:~/Class/network/udp\$./recv 127.0.0.1 9090
- B 终端执行命令
 - ./send 127.0.0.1 9090

三. 课后任务

练习:

1. 新建一个 udp_client 和 udp_server 的文件夹
2. 在 udp_client 文件夹中新建一个 log.txt 文件，然后写入以下数据
 1. 然后编写一个 udp_file_client.c 的代码，代码中设计一个结构体。

typedef struct

```

{

char file_name[20]; // 文件名

char file_data[100]; // 文件内容

int file_size; // 文件内容大小

```

```
}packet_t;
```

要求读取 log.txt 文件的内容，并把文件名、文件内容和文件内容大小分别填充到上述结构体中。然后把结构体发送给服务器。

4. 最后编写一个 udp_file_server.c 的代码，接收客户端发送过来的结构体。然后再 udp_server 目录下，根据结构体名字，文件内容，文件大小。创建一个一样的文件。

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

