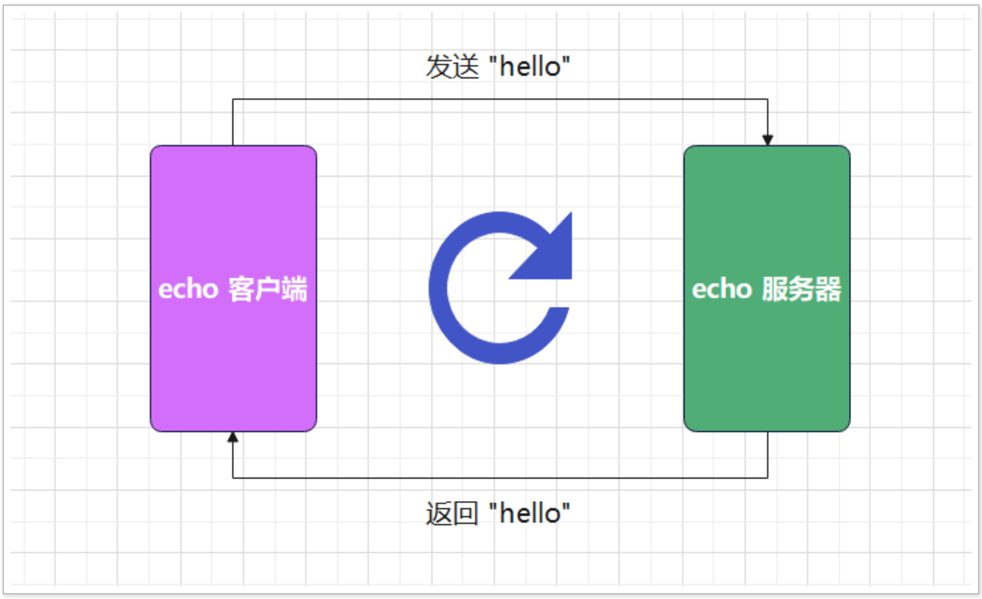
2.3 tcp 服务器实现流程 (三)- 数据接收与发送_物联网 / 嵌入式工程师 - 慕课网

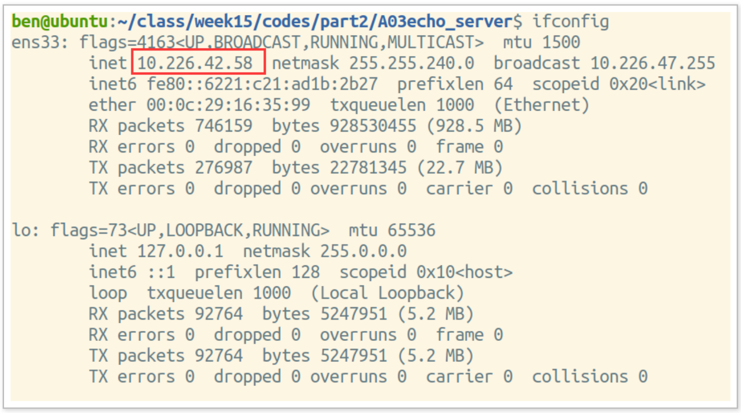
> 慕课网慕课教程 2.3 tcp 服务器实现流程（三）– 数据接收与发送涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

- tcp 服务器数据接收与发送都是使用 send 函数与 recv 函数
- 示例：实现 echo 服务器
-
    - echo 服务器: 将客户端发送的数据再重新发送给客户端



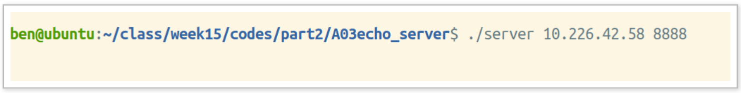
```
#define LISTEN_SZ 10

int main(int argc,char *argv[])
{
    if (argc != 3){
        fprintf(stderr,"usage : %s < ip > < port >.\n",argv[0]);
        exit(EXIT_FAILURE);
    }

    int sfd,ret,cfd;
    struct sockaddr_in  svr_addr,cli_addr;

    ssize_t sbytes,rbytes;
    char buffer[1024] = {0};

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

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


    ret = listen(sfd,LISTEN_SZ);
    if (ret == -1){
        perror("[ERROR] listen(): ");
        close(sfd);
        exit(EXIT_FAILURE);
    }

    socklen_t len = sizeof(struct sockaddr_in);
    bzero(&cli_addr,sizeof(struct sockaddr));
    cfd = accept(sfd,(struct sockaddr *)&cli_addr,&len);
    if (cfd == -1){
        perror("[ERROR] accept(): ");
        exit(EXIT_FAILURE);
    }

    printf("ip : %s,port : %d\n",inet_ntoa(cli_addr.sin_addr),ntohs(cli_addr.sin_port));


    rbytes = recv(cfd,buffer,sizeof(buffer),0);
        if (rbytes == -1){
            perror("recv(): ");

            exit(EXIT_FAILURE);
        }else if (rbytes == 0){
            printf("The client is offline.\n");

            exit(EXIT_FAILURE);

        }else if (rbytes > 0){

            sbytes = send(cfd,buffer,sizeof(buffer),0);
            if (sbytes == -1){
                perror("[ERROR] send(): ");

                exit(EXIT_FAILURE);
            }
        }

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

- 测试方法如下:
    - 运行服务器
        - step 1 : 查看当前系统的 ip 地址



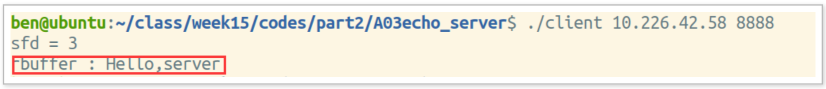
```
ben@ubuntu:~/class/week15/codes/part2/A03echo_server$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.226.42.58  netmask 255.255.240.0  broadcast 10.226.47.255
        inet6 fe80::6221:c21:ad1b:2b27  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:16:35:99  txqueuelen 1000  (Ethernet)
        RX packets 746159  bytes 928530455 (928.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 276987  bytes 22781345 (22.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 92764  bytes 5247951 (5.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 92764  bytes 5247951 (5.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

        - step 2 : 运行服务器程序



```
ben@ubuntu:~/class/week15/codes/part2/A03echo_server$ ./server 10.226.42.58 8888
```

    - 运行客户端



```
ben@ubuntu:~/class/week15/codes/part2/A03echo_server$ ./client 10.226.42.58 8888
sfd = 3
buffer : Hello,server
```

将示例程序的 echo 服务器修改成 能够循环接收客户端信息，客户端也需要支持循环发送，输入 quit 则退出

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