2.2 tcp 服务器实现流程 (二)- 建立监听队列与接收连接_物联网 / 嵌入式工程师 - 慕课网

> 慕课网慕课教程 2.2 tcp 服务器实现流程 (二)– 建立监听队列与接收连接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。

- 在服务器绑定 ip 地址与端口号之后, 则需要让服务器 socket 套接字设置成被动监听状态，并创建监听队列，这里需要调用 listen 函数
- listen 函数的详细信息如下:

函数头文件 #include <sys/types.h>

#include <sys/socket.h>

函数原型

int listen(int sockfd,int backlog)

函数功能 设置套接字状态为被动监听，并创建监听队列

函数参数 sockfd : 套接字文件描述符

backlog : 监听队列的长度

函数返回值 成功 : 返回 0

失败 : 返回 –1, 并设置 errno

示例

在 tcp 服务器中 设置套接字为监听状态, 并创建监听队列

- 当 客户端 发出连接请求之后, 则需要调用 accept 函数建立连接
- accept 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

int accept(int sockfd,struct sockaddr *addr,socklen_t *addrlen)

函数功能 接受来自于其他 socket 的连接请求，并建立连接

函数参数 sockfd : 套接字文件描述符

addr : 网络地址结构的指针（输出参数，用于保存发送请求端的地址信息）

addrlen : 网络地址结构长度的指针 (输出参数，但是需要进行初始化)

函数返回值 成功 : 返回新的文件描述符

失败：–1 , 并设置 errno

示例:

设计一个服务器程序，并和客户端建立连接，并打印客户端的 ip 地址和端口号

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

#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;


    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));


    close(cfd);
    close(sfd);

    return 0;
}
```

- 测试方法:
    - 运行服务器，等待连接
        - step 1 : 查看当前系统的 ip 地址或者使用 127.0.0.1

```
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>
        inet6 240e:45e:cd0:c8ed:4f7b:c5b8:aa7d:5d  prefixlen 64  scopeid 0x0<global>
        inet6 240e:45e:cd0:c8ed:ece0:5e1a:56f3:f7bd  prefixlen 64  scopeid 0x0<globa
        ether 00:0c:29:16:35:99  txqueuelen 1000  (Ethernet)
        RX packets 745863  bytes 928501095 (928.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 276819  bytes 22765210 (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 92465  bytes 5230293 (5.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 92465  bytes 5230293 (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        ben@ubuntu: ~/class/week15/codes/part1
ben@ubuntu:~/class/week15/codes/part2$ ls
A01tcp_socket_bind.c  A02listen_accept.c  a.out
ben@ubuntu:~/class/week15/codes/part2$ ./a.out 10.226.42.58 8888
                                                  端口号
                                          ip 地址
```

- step 3：运行客户端程序

```
ben@ubuntu: ~/class/week15/codes/part2        ben@ubuntu: ~/class/week1
ben@ubuntu:~/class/week15/codes/part1$ ./a.out 10.226.42.58 8888
```

- step 4：运行结果如下:

```
ben@ubuntu:~/class/week15/codes/part2$ ./a.out 10.226.42.58 8888
ip : 10.226.42.58,port : 47500
ben@ubuntu:~/class/week15/codes/part2$
```

完成示例程序

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