

廈門大學



信息学院软件工程系

《计算机网络》实验报告

题 目 实验七 代理服务器软件

班 级 软件工程 2019 级 4 班

姓 名 陈栋

学 号 22920192204171

实验时间 2021 年 5 月 28 日

2021 年 5 月 28 日

填写说明

- 1、本文件为 Word 模板文件，建议使用 Microsoft Word 2019 打开，在可填写的区域中如实填写；
- 2、填表时，勿破坏排版，勿修改字体字号，打印成 PDF 文件提交；
- 3、文件总大小尽量控制在 1MB 以下，勿超过 5MB；
- 4、应将材料清单上传在代码托管平台上；
- 5、在学期最后一节课前按要求打包发送至 cni21@qq.com。

1 实验目的

通过完成实验，掌握基于 RFC 应用层协议规约文档传输的原理，实现符合接口且能和已有知名软件协同运作的软件

2 实验环境

Windows 10; vs2019

3 实验结果

由于系统环境原因对实验代码进行了阅读修改，没有实际运行成功原代码

服务器的设计思路：

1、通过 socket()函数，建立基本的套接字

```
/*
 * 返回socket编号
 * int socket(int domain, int type, int protocol);
 * domain:AF_INET表明使用tcp/ip协议模型
 * type:套接字类型stream对应tcp协议
 * protocol:协议类型。
 */
if ((sock_fd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
    log_message("socket()");
    exit(1);
}
```

2、通过 bind()函数，绑定本地地址与端口

```
/*用来给参数sockfd的socket一个名称
 * int bind(int sockfd, struct sockaddr * my_addr, int addrlen);
 * my_addr指向一sockaddr结构，对于不同的socket domain定义了一个通用的结构体
 */
if (bind(sock_fd, (struct sockaddr*)&local, sizeof(local)) < 0) {
    log_message("bind()");
    exit(1);
}
```

3、通过 listen()函数，监听，并设置并发数

```
//创建一个套接口，并监听申请的连接，允许最大的排队连接队列为25
if (listen(sock_fd, 25) < 0) {
    log_message("listen()");
    exit(1);
}
```

4、通过 accept()接受连入的套接字，并通过 pthread()新建线程处理请求

```
while (1) {
    //获取连接
    if ((net_fd =
        accept(sock_fd, (struct sockaddr*)&remote,
        &remotelen)) < 0) {
        log_message("accept()");
        exit(1);
    }

    int one = 1;
    /*设置socket状态
    * int setsockopt(int s, int level, int optname, const void * optval, socklen_t optlen);
    * s: 需要设置的socket
    * level: 指定socket的协议类型
    * optname: 选项名称    TCP_NODELAY 不使用Nagle算法
    * optval: 指向存放选项值的缓冲区
    * optvalen: 缓冲区的长度
    */
    setsockopt(sock_fd, SOL_TCP, TCP_NODELAY, &one, sizeof(one));
    if (pthread_create(&worker, //线程标识符
        NULL, //线程属性
        &app_thread_process, //线程函数的起始地址
        (void*)&net_fd //传递给start_routing的参数//传递给线程函数的参数
    ) == 0) { //返回0成功
        pthread_detach(worker); //将子进程与主进程分离，分离后资源自动回收
    }
    else {
        log_message("pthread_create()");
    }
}
```

5、在处理请求时，读取一个字节，判断 socks 版本号是 4 还是 5，交由相应的模块处理

```
char methods = socks_invitation(net_fd, &version);

switch (version) {
    case VERSION5: { ... }
    case VERSION4: { ... }
}
```

6、Socks5，判断命令是针对 IP 还是域名的处理

```

case VERSION5: { //socks5
    socks5_auth(net_fd, methods);
    int command = socks5_command(net_fd);
    //对报文中的信息进行处理
    if (command == IP) { ... }
    else if (command == DOMAIN) { ... }
    else {
        app_thread_exit(1, net_fd);
    }
}

```

7、建立对目标 IP 和端口的连接，读取内容，并发送到客户端上

```

socks5_ip_send_response(net_fd, ip, p); //对客户端发送应答信息
socks5_domain_send_response(net_fd, address, size, p); //绑定成功，发送应答信息

```

8、Socks4，则是址对 ip 地址或端口建立连接

```

if (socks4_is_4a(ip)) { ... }
else {
    log_message("Socks4: connect by ip & port");
    inet_fd = app_connect(IP, (void*)ip, ntohs(p));
}

```

当端口连接成功后，监听端口，并从端口中获取报文信息等内容，实现信息的交换与转发。

```

ret = select(maxfd + 1, &rd_set, NULL, NULL, NULL); //对maxfd进行度操作
if (ret < 0 && errno == EINTR) {
    continue;
}
//用于测试指定的文件描述符是否在该集合中。
if (FD_ISSET(fd0, &rd_set)) {
    nread = recv(fd0, buffer_r, BUFSIZE, 0);
    if (nread <= 0)
        break;
    send(fd1, (const void*)buffer_r, nread, 0);
}
//用于测试指定的文件描述符是否在该集合中。
if (FD_ISSET(fd1, &rd_set)) {
    nread = recv(fd1, buffer_r, BUFSIZE, 0);
    if (nread <= 0)
        break;
    send(fd0, (const void*)buffer_r, nread, 0);
}

```

4 实验代码

本次实验的代码已上传于以下代码仓库：[cd888888/network: report \(github.com\)](https://github.com/cd888888/network-report)

5 实验总结

理解了客户端的编写步骤的函数调用

在客户端运行时，合理的多线程的调用管理

Socks 代理的相关内容，以及连接特点，socks4 主要是运用运用 tcp 协议，socks5 两种协议 tcp 和 udp 均可以运用，在编写客户端时，要注意区分 socks 的版本，以及版本内的内容