厦門大學

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　郑志豪

学　　号　22920192204336

实验时间　2021 年 6 月 09 日

2021 年 6 月 09 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

由于系统不支持，未能实现新的代理软件，故决定对附录二程序做分析。

服务器设计思路：

1. 通过 socket()函数建立套接字

AF_INET 是 IPv4 的 Internet 地址族格式

SOCK_STREAM 类型可将用户数据报协议（TCP）用于 Internet 地址系列。

```
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

sock_fd 是标识未绑定套接字的描述符。

local 是指向要分配给绑定套接字的本地地址的 sockaddr 结构的指针。

第三个参数是该指针指向的值的长度

```
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

```
pthread_t worker;
while (1) {
    if ((net_fd =
        accept(sock_fd, (struct sockaddr*)&remote,
            &remotelen)) < 0) {
        log_message("accept()");
        exit(1);
    }
    int one = 1;
    setsockopt(sock_fd, SOL_TCP, TCP_NODELAY, &one,
        sizeof(one));
    if (pthread_create
    (&worker, NULL, &app_thread_process,
        (void*)&net_fd) == 0) {
        pthread_detach(worker);
    }
    else {
        log_message("pthread_create()");
    }
}
```

5. 处理请求时需要判断 socks 版本号，通过 switch 来实现分支处理

```
    char methods = socks_invitation(net_fd, &version);
    switch (version) {
    case VERSION5: { ... }
    case VERSION4: {
        if (methods == 1) {
            char ident[255];
```

6. socks5 中判断命令针对 IP 还是域名

```
case VERSION5: {
    socks5_auth(net_fd, methods);
    int command = socks5_command(net_fd);
    if (command == IP) { ... }
    else if (command == DOMAIN) { ... }
    else {
        app_thread_exit(1, net_fd);
    }
}
```

建立对目标 IP 和端口的连接后读取内容并将其发送到客户端上

```
socks5_ip_send_response(net_fd, ip, p);
```

```
socks5_domain_send_response(net_fd, address,
    size, p);
```

7. socks4 中则只对 IP 或者只对端口建立连接

```
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

```
while (1) {
    FD_ZERO(&rd_set);
    FD_SET(fd0, &rd_set);
    FD_SET(fd1, &rd_set);
    ret = select(maxfd + 1, &rd_set, NULL, NULL, NULL);
    if (ret < 0 && errno == EINTR) {
        continue;
    }
    //用于测试
    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);
    }
    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：https://github.com/zzh221/E7

5 实验总结

理解了客户端的编写步骤的函数调用，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

。