

廈門大學



信息学院软件工程系

《计算机网络》实验报告

题 目 实验五 利用 Socket API 实现许可认证软件

班 级 软件工程 2019 级 2 班

姓 名 袁佳哲

学 号 11920192203642

实验时间 2021 年 6 月 3 日

2021 年 6 月 3 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、材料清单上传在代码托管平台上；
- 5、在学期末将报告按“CNI-E1-00020190000000-张三”的命名后（其中 E1 是 Experiment 1 的缩写），压缩为 zip 文件作为附件，以“计算机网络-实验报告-00020190000000-张三”为主题发送至 cni21@qq.com。

1 实验目的

通过完成实验,掌握应用层文件传输的原理;了解传输过程中传输层协议选 用、应用层协议设计和协议开发等概念。

2 实验环境

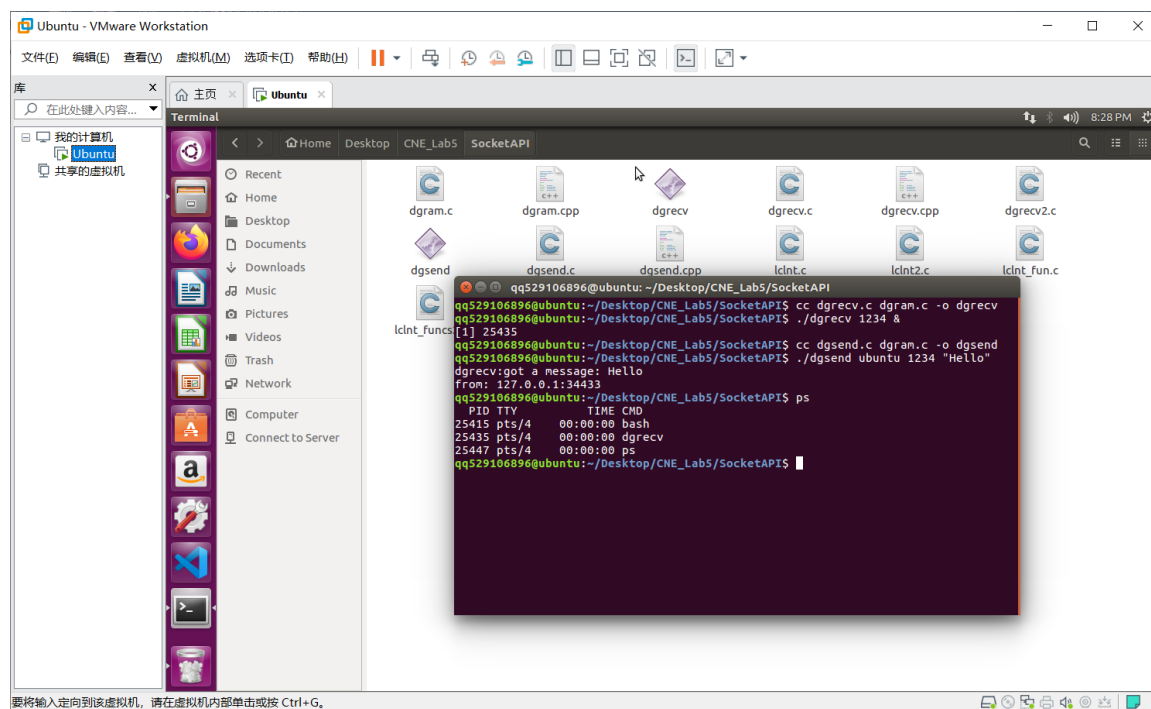
Linux 操作系统, C 语言

3 实验结果

编译与测试 UDP

编译服务器,并启动它,使得它监听端口 1234,然后编译和运行客户,使得客户发送字符串到端口 1234。服务器接收消息,打印消息,并且打印消息的返回地址。客户 socket 拥有主机地址和端口号,内核随机地给它分配了一个端口 34433。

利用 ps 指令可以看到服务器正在 25435 端口运行



服务器与客户端测试:

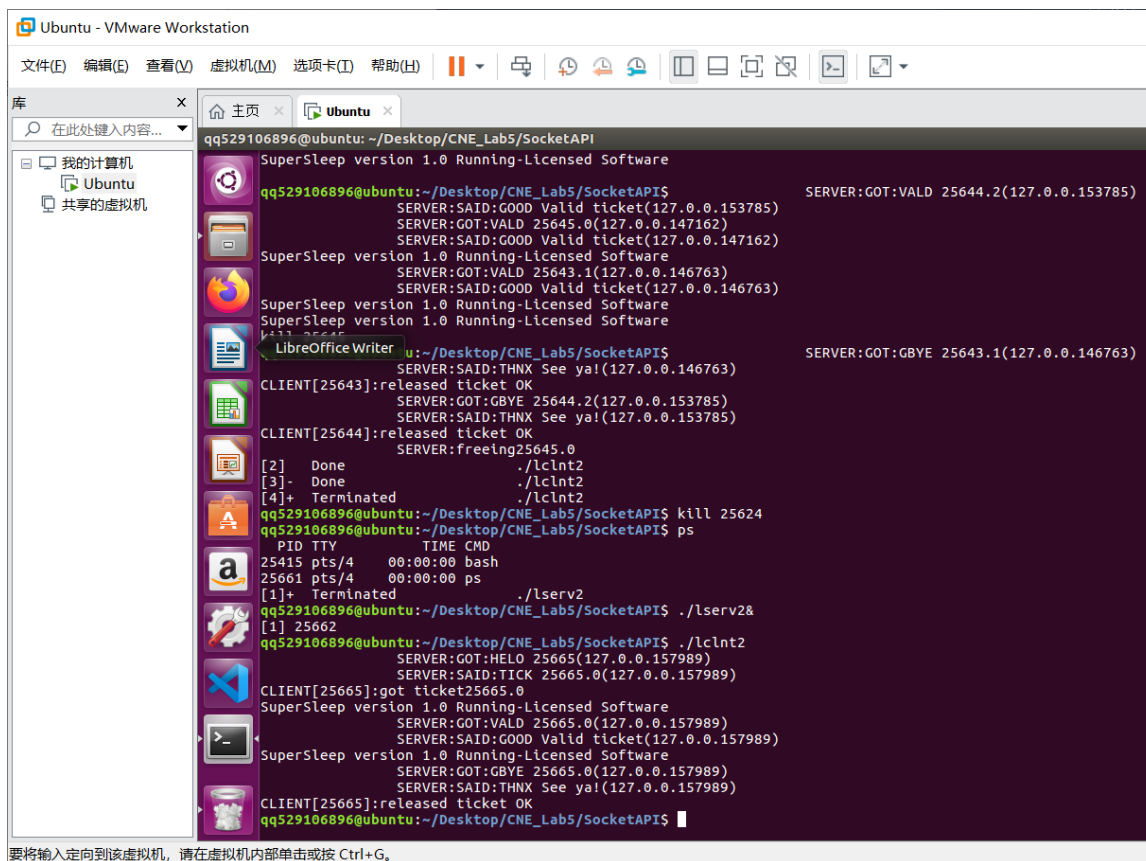
启动一个服务器,服务器在 25624 上运行;然后启动三个客户端,分别在 25643、25644、25645 上运行,可以看到服务器与客户端的交互过程,中间杀死客户 25645,可以看到最终 25654 的状态为 Terminated 而非 Done

The image shows a Windows 10 desktop with a VMware Workstation application open. The main window displays a virtual machine named 'Ubuntu'. The terminal window within the VM is active, showing a series of commands and outputs. The commands are executed in a shell, and the outputs show the results of the commands, including server responses and ticket requests. The left sidebar of the VMware Workstation shows the 'My Computer' view, with the 'Ubuntu' virtual machine selected. The terminal window shows the following commands and outputs:

```
qq529106896ubuntu:~/desktop/CNE_Labs/SocketAPI$ cc lcInt2.c lcInt_funcs2.c dgram.c -o lcInt2
qq529106896ubuntu:~/desktop/CNE_Labs/SocketAPI$ ./serv28
[1] 25624
[2] 25643
[3] 25644
[4] 25645
qq529106896ubuntu:~/desktop/CNE_Labs/SocketAPI$ SERVER:GOT:HELO 25645.(127.0.0.147162)
SERVER:SAID:TICK 25645.0(127.0.0.147162)
SERVER:GOT:HELO 25643.(127.0.0.146763)
SERVER:SAID:TICK 25643.1(127.0.0.146763)
SERVER:GOT:HELO 25644.(127.0.0.153785)
SERVER:SAID:TICK 25644.2(127.0.0.153785)
CLIENT[25644]:got ticket25644.2
SuperSleep version 1.0 Running-Licensed Software
CLIENT[25645]:got ticket25645.0
SuperSleep version 1.0 Running-Licensed Software
CLIENT[25643]:got ticket25643.1
SuperSleep version 1.0 Running-Licensed Software
qq529106896ubuntu:~/desktop/CNE_Labs/SocketAPI$ SERVER:GOT:VALD 25644.2(127.0.0.153785)
SERVER:SAID:GOOD Valid ticket(127.0.0.153785)
SERVER:GOT:VALD 25645.0(127.0.0.147162)
SERVER:SAID:GOOD Valid ticket(127.0.0.147162)
SuperSleep version 1.0 Running-Licensed Software
SERVER:GOT:VALD 25643.1(127.0.0.146763)
SERVER:SAID:GOOD Valid ticket(127.0.0.146763)
SuperSleep version 1.0 Running-Licensed Software
SuperSleep version 1.0 Running-Licensed Software
kill 25645
qq529106896ubuntu:~/desktop/CNE_Labs/SocketAPI$ SERVER:GOT:GBYE 25643.1(127.0.0.146763)
SERVER:SAID:THNX See ya!(127.0.0.146763)
CLIENT[25643]:released ticket OK
SERVER:GOT:GBYE 25644.2(127.0.0.153785)
SERVER:SAID:THNX See ya!(127.0.0.153785)
CLIENT[25644]:released ticket OK
SERVER:freelnt25645.0
[2] Done ./lcInt2
[3] Done ./lcInt2
[4]+ Terminated ./lcInt2
qq529106896ubuntu:~/desktop/CNE_Labs/SocketAPI$
```

关闭服务器，然后重新启动，再添加一个新的客户。因客户端被设置为建立——发送验证——验证成功即关闭，所以之前的 25643、25644、25645 客户端在此处不再显示。

具体阐述：每建立一个客户端与服务器的连接，客户端就申请一个许可证，服务器做出回应，给客户端一个许可证，客户端收到之后验证这个许可证，服务器端回复验证，客户端归还许可证，服务端接收这个返还之后，即撤销客户端与服务器连接，客户端也完成了自己的任务，而服务器仍然在运行

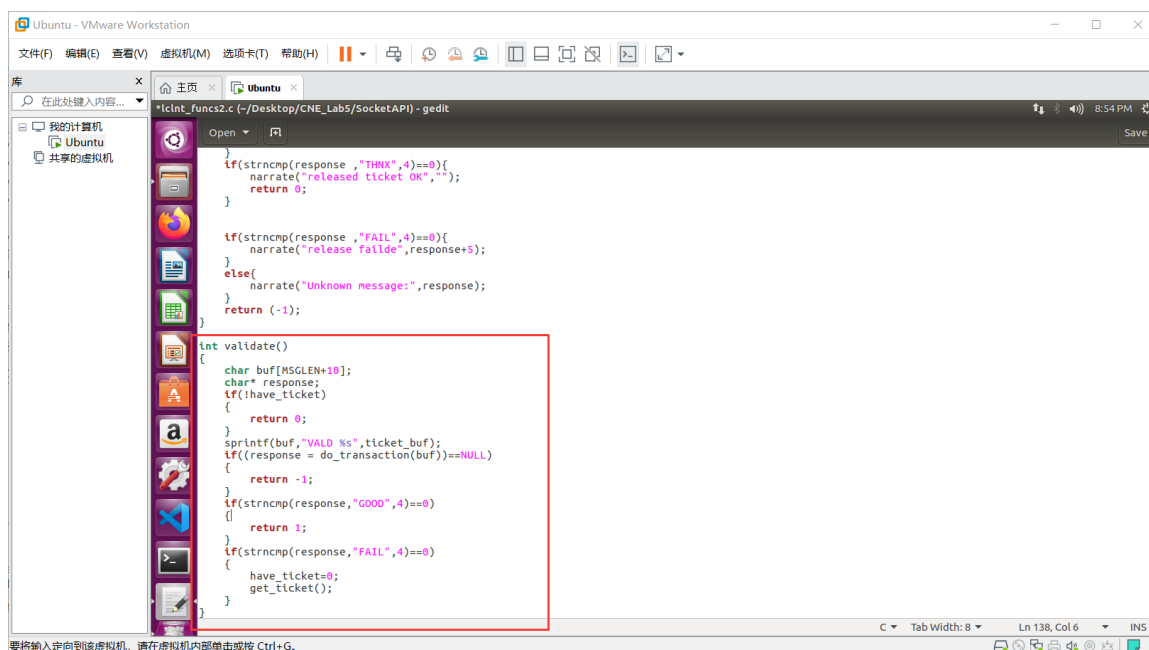


```
Ubuntu - VMware Workstation
文件(E) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
库
  在此处键入内容...
  我的计算机
    Ubuntu
  共享的虚拟机
  主页
  Ubuntu
qq529106896@ubuntu: ~/Desktop/CNE_Lab5/SocketAPI
SuperSleep version 1.0 Running-Licensed Software
qq529106896@ubuntu:~/Desktop/CNE_Lab5/SocketAPI$ SERVER:SAID:GOOD Valid ticket(127.0.0.153785) SERVER:GOT:VALD 25644.2(127.0.0.153785)
SERVER:GOT:VALD 25645.0(127.0.0.147162)
SERVER:SAID:GOOD Valid ticket(127.0.0.147162)
SuperSleep version 1.0 Running-Licensed Software
SERVER:GOT:VALD 25643.1(127.0.0.146763)
SERVER:SAID:GOOD Valid ticket(127.0.0.146763)
SuperSleep version 1.0 Running-Licensed Software
SuperSleep version 1.0 Running-Licensed Software
qq529106896@ubuntu:~/Desktop/CNE_Lab5/SocketAPI$ SERVER:SAID:THNX See ya!(127.0.0.146763) SERVER:GOT:GBYE 25643.1(127.0.0.146763)
CLIENT[25643]:released ticket OK
SERVER:GOT:GBYE 25644.2(127.0.0.153785)
SERVER:SAID:THNX See ya!(127.0.0.153785)
CLIENT[25644]:released ticket OK
SERVER:freelng25645.0
[2] Done ./lclnt2
[3] Done ./lclnt2
[4]+ Terminated ./lclnt2
qq529106896@ubuntu:~/Desktop/CNE_Lab5/SocketAPI$ kill 25624
qq529106896@ubuntu:~/Desktop/CNE_Lab5/SocketAPI$ ps
PID TTY TIME CMD
25415 pts/4 00:00:00 bash
25661 pts/4 00:00:00 ps
[1]+ Terminated ./lserv2
qq529106896@ubuntu:~/Desktop/CNE_Lab5/SocketAPI$ ./lserv2&
[1] 25662
qq529106896@ubuntu:~/Desktop/CNE_Lab5/SocketAPI$ ./lclnt2
SERVER:GOT:HELO 25665(127.0.0.157989)
SERVER:SAID:TICK 25665.0(127.0.0.157989)
CLIENT[25665]:got ticket25665.0
SuperSleep version 1.0 Running-Licensed Software
SERVER:GOT:VALD 25665.0(127.0.0.157989)
SERVER:SAID:GOOD Valid ticket(127.0.0.157989)
SuperSleep version 1.0 Running-Licensed Software
SERVER:GOT:GBYE 25665.0(127.0.0.157989)
SERVER:SAID:THNX See ya!(127.0.0.157989)
CLIENT[25665]:released ticket OK
qq529106896@ubuntu:~/Desktop/CNE_Lab5/SocketAPI$
```

要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。

核心部分代码如图中红框所示，为客户端添加验证功能，服务端添加相应的处理验证请求与返回信息的功能

具体阐释：每隔一段时间就进行验证，如果验证成功，即收到 **GOOD** 消息，继续使用当前票据；如果收到 **FAIL** 消息，即验证失败，需要重新向服务端申请票据



```
*clnt_funcs2.c (-/Desktop/CNE_Lab5/SocketAPI) - gedit
Open
Save

}
if(strncmp(response, "THX", 4) == 0){
    narrate("released ticket OK", "");
    return 0;
}

if(strncmp(response, "FAIL", 4) == 0){
    narrate("release failde", response+5);
}
else{
    narrate("Unknown message:", response);
}
return (-1);
}

int validate()
{
    char buf[MSGLEN+10];
    char* response;
    if(!have_ticket)
    {
        return 0;
    }
    sprintf(buf, "VALD %s", ticket_buf);
    if((response = do_transaction(buf)) == NULL)
    {
        return -1;
    }
    if(strncmp(response, "GOOD", 4) == 0)
    {
        return 1;
    }
    if(strncmp(response, "FAIL", 4) == 0)
    {
        have_ticket = 0;
        get_ticket();
    }
}
```

4 实验代码

本次实验的代码已上传于以下代码仓库：
https://www.gitee.com/yjz6666774/e3642/tree/master/E5_3642

5 实验总结

1. 可以观察到服务端和客户端在进行通信时，连接基本是都是秒建立的，所以用的是 **UDP** 协议。如果是 **TCP**，在建立连接时会有三次握手，每次撤销连接会有四次挥手。不过考虑到仅仅是进行测试，不太要求连接质量，**UDP** 即可满足要求。
2. 所有通信基于数据报（Datagram）

3. 服务端和客户端的大致分工如下：

客户端：申请——接收服务端返还消息——根据消息做出判断，执行下一步操作

服务端：接收客户端申请——检查当前状态，如剩余的票据数量、当前票据是否合法——对客户端做出回应