

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 数字媒体技术 2023 级 1 班

姓 名 张琳

学 号 37220232203919

实验时间 2025 年 12 月 14 日

2025 年 12 月 14 日

填写说明

- 1、本文件为 Word 模板文件，建议使用 Microsoft Word 2024 打开，在可填写的区域中如实填写；
- 2、填表时勿改变字体字号，保持排版工整，打印为 PDF 文件提交；
- 3、文件总大小尽量控制在 1MB 以下，最大勿超过 5MB；
- 4、在实验课结束 14 天内，按实验报告提交到我校课程网站的指定位置，源代码等主要材料上传在公开的代码托管平台上。
- 5、鼓励同学之间探讨，鼓励合理使用人工智能平台，提升效率，但不应滥用相关资源，如抄袭代码和代写作业。

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

Windows11

3 实验结果

需求说明：

- 1、 某组织管理员在购买许可证时，输入用户名、口令和许可证类型，许可证程序返回一个由 10 个数字组成的序列号。
- 2、 该组织的用户第一次使用软件 A 时，输入序列号。
- 3、 该组织用户运行软件时，向许可证服务器发送验证。
- 4、 许可证服务器查询得到该序列号的使用人数，如果未到达上限，则返回授权指令；否则，返回拒绝指令。
- 5、 软件 A 得到授权指令，允许用户使用软件。否则，提示用户稍后再试，退出程序。

当软件 A 或非正常退出（崩溃被其它程序中止）时，许可证服务器应在扣除使用人数时剔除它。可以使软件 A 定期（如：30 分钟）向服务器报告其状态，超过一定时间没有收到报告时，则认定崩溃。

当许可证服务器崩溃时，软件 A 应能在重新启动时恢复。许可证服务器重启后，如果新的软件 A 前来连接，服务器不可以因接受其连接而拒绝已认证用户的连接。

分析，分成两个部分：

1. 获取许可证，输入字符串用户、口令，返回一个新的序列号，服务器记录该序列号已分发

2.

- 启动阶段：

- 服务器启动时读取存储的连接表，并认为此前为崩溃，将连接表所有用户心跳时间设为现在，确保旧用户优先重连；

- 客户端连接时，先判断是否为合法序列号，再判断人数上限；

- 连接阶段：

- 服务器每次成功连接时、客户端断连时更新存储表；

- 服务器每间隔一段时间检查连接表，如果心跳超时则认为该客户端断连；

- 客户端每间隔一段时间向服务器发送心跳，若发送失败，则等待重连，多次重连失败回退到启动；

实现：

使用 flask 作为前端界面，用于获取序列号和显示当前在线列表，其中前端 html 代码为 ai 生成



当前页面：申请许可证 | [前往监控后台](#) →

获取软件许可证

请输入用户信息以生成唯一序列号

用户名

口令 (任意)

立即生成

生成成功！您的序列号是：
5076495300
请复制并在客户端软件中使用

[← 返回申请页](#) 许可证并发监控 (Max=10) 刷新数据

序列号 (SN)	在线人数	状态	当前接入客户端 (IP:Port)
2023003919	0 / 3	空闲	None
5076495300	0 / 3	空闲	None

使用 tkinter 作为客户端软件，用 uuid 标识每个客户端

Client ddc0db29

请输入许可证序列号(10位):

验证并运行

状态: 未运行

连接后客户端间隔发送心跳，服务端间隔检查上次心跳时间

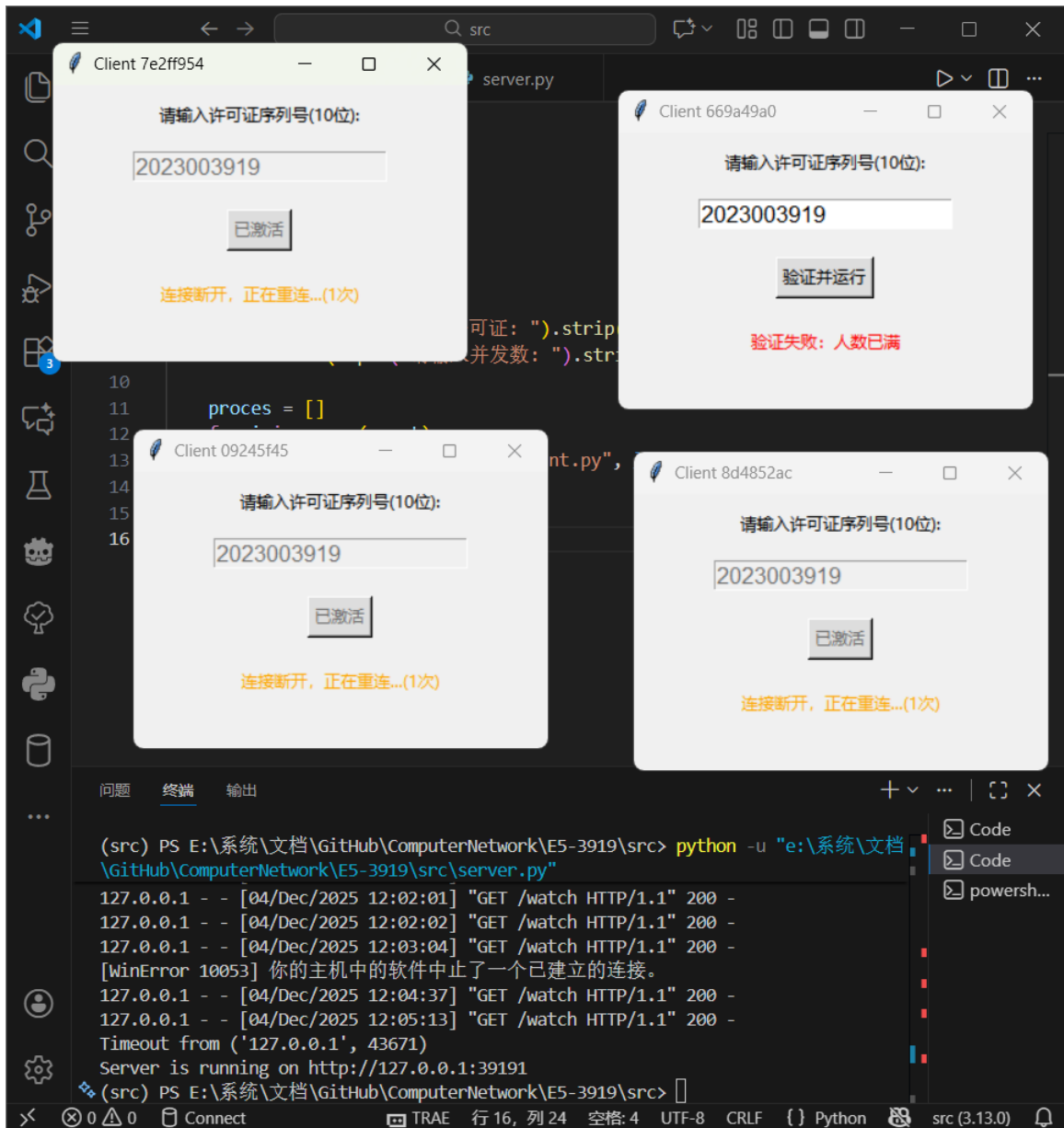


序号 (SN)	在线人数	状态	当前接入客户端 (IP:Port)
2023003919	1 / 3	使用中	127.0.0.1:50837

连接数量超出时拒绝连接



服务器崩溃时客户端间隔尝试重连



重连时优先连接旧用户（演示前右上点击了“验证并运行”）



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/lzydroper/ComputerNetwork_Homework/tree/main/E5-3919。(注意：

建议使用码云，并设置公开权限；本学期暂不推荐使用 GitHub；如使用厦门大学私有 Git 服务，应将 whuang@xmu.edu.cn 加入项目成员备查，本段话删除。） |

5 课后思考题

|（注明题号和题目文字，逐个回答课后思考题。如无，填写无。本段话删除。） |

6 实验总结

|本来是被 ai 骗去用 nodejs 的，后面发现那个用的是 WebSocket，想用原生 Socket 实验有点本末倒置，后面只得跑去用 flask 了，最初是希望客户端也是个网页界面的，可惜。

|