

深圳大学期末考试试卷

开/闭卷 开卷

A/B 卷

课程编号 1503020001

课序号 01

课程名称 系统编程

学分 3

命题人(签字) 审题人(签字) 年 月 日

题号	一	二	三	四	五	六	七	八	九	十	基本题 总分	附加题
得分												
评卷人												

特别说明：所有作答过程建议在华为 Taishan 服务器+openEuler 上完成。

在作答之前，请首先确认你的实验环境满足以下要求：

1. 请以名为“学生姓名拼音+学号”的用户登录、完成作业，如学生名张三，学号 123456，则用户名为 zhangsan123456。请特别注意，以用户名 zhangsan123456 完成的作业记为 0 分。

2. 用户提示符需要包含用户名、主机名和工作目录。

参考样例 1) 名为 szu 的用户在她/他的虚拟机 taishan02-vm-10 下工作时，工作目录为 process，则终端提示符显示为（此时，环境变量 PS1='[\u@\h \W]\\$ '）：

```
[szu@taishan02-vm-10 process]$
```

参考样例 2) 用户 caichengru2020152021 在虚拟机 taishan01-sp-2023-3 的主工作（~）时，终端提示符显示为

```
caichenru2020152021 at taishan01-sp-2023-3 in ~
```

3. 请按题目要求完成编码和测试，按要求运行代码、截屏，填写 **2025 年度《系统编程》大作业答题纸**，在截止日期（**2025 年 6 月 27 日 16:30**）前通过 Blackboard 提交**答题纸电子版和代码**，**2025 年 6 月 27 日 18:00 前**

提交纸质版本(粤海校区下文山湖湖边粤海门广场实验楼 A 座 202)。没有赶上 Blackboard 提交截止日期的同学，可在 2025 年 6 月 30 日 23:30 前通过 QQ、微信、电子邮件等方式联系主讲老师提交电子版，并在 2025 年 7 月 01 日 17:30 提交纸质版，这之后不再接收任何答卷。特别注意：2025 年 6 月 27 日 16:30~2025 年 06 月 30 日 23:30 之间提交电子版大作业的计入迟交，迟交的同学期末总评最高分为 92，不会有 A+ 的机会。

4. 必须提交纸质答卷：1) 平时课堂上提交，最后一次课是 2025 年 6 月 23 日下午，致理楼 L2-409。2) 主讲老师办公室：粤海校区下文山湖湖边粤海门广场实验楼 A 座 202。
5. 大作业分数的评定将根据提交的答题纸上实验结果截屏的正确性、可读性、描述与分析有效性和逻辑性、代码的质量等。所有截屏终端提示符信息与
学生姓名学号不符的皆为 0 分。
6. 本次考核设置 3 道大题，任选 1 题作答即可。请根据个人兴趣与能力选答。其中选题(一)和(二)是独立完成，选题(三)是团队完成，难度依次递增。注意事项：
 - 1) 期末得分与选题无关
 - 2) 选题(一) “功能迭代”，研发“单机多用户聊天”系统软件，考核内容与授课内容密切相关，在平时实验基础上拓展功能或优化性能。
 - 3) 选题(二) “开源贡献”，需要独立阅读内核代码，发现问题、解决问题并提交到开源社区，A+需要有功能增强或性能提升相关的代码贡献；其所有开源贡献需要有网站或开源贡献收录邮件截图证明。

- 4) 选题(三) “应用创新”，以特色家禽智慧养殖系统为基座，针对其中特定模块与要求，应用本课程内容优化其性能优化，要结合代码和运行数据进行分析。

选题(一) 功能迭代

利用操作系统提供的系统调用、标准 C 库函数和 POSIX 线程库 pthread 函数设计并实现单机多用户聊天系统软件。并利用学过的知识深入分析问题

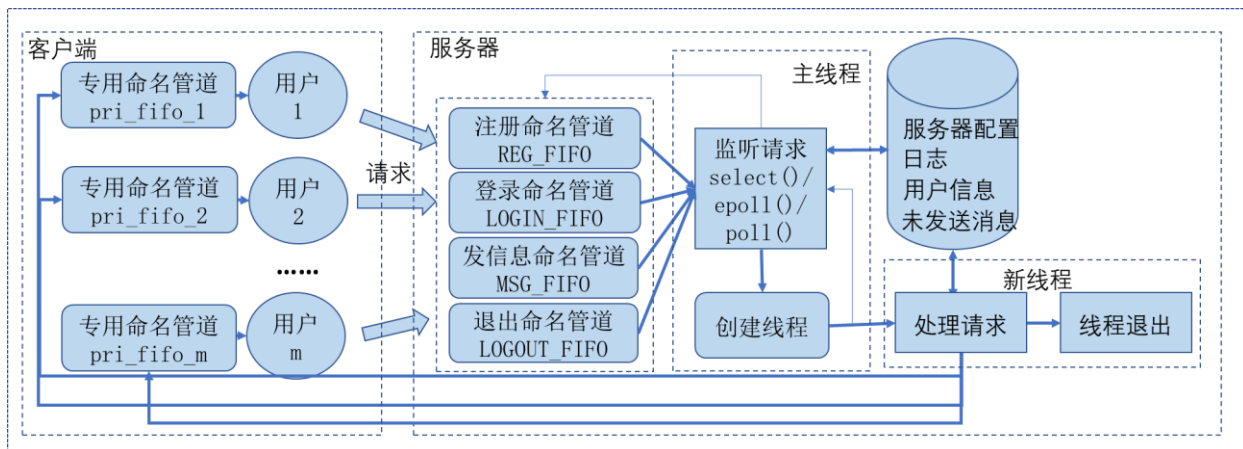


图 1. 单机多用户聊天系统参考架构图

1. 服务器初始化（40 分），见图 1，服务器是多线程守护进程。

1.1. 服务器配置文件设置以下信息：

1.1.1. 服务器名称和版本

服务器命名格式为“chatserver_学生姓名拼音首字母缩写_版本号”。首先，以张三为例，学生姓名拼音首字母缩写为“zs”。其次服务器的版本号设置由学生自定义。可以参考 Linux 内核版本号命名的方式。Linux 内核版本有两种：稳定版和开发版，Linux 内核版本号由 3 组数字组成：x.y.z。其中 x 代表目前发布的内核主版本；y 为偶数表示稳定版本，奇数表示开发中版本；z 表示错误修补的次数，此时版本号为：1.0.0。例如张三同学的服务器命名为：chatserver_zs_1.0.0

1.1.2. 众所周知的命名管道

- a) REG_FIFO : 命名为“学生姓名拼音首字母缩写+register”
- b) LOGIN_FIFO : 命名为“学生姓名拼音首字母缩写+login”
- c) MSG_FIFO : 命名为“学生姓名拼音首字母缩写+sendmsg”
- d) LOGOUT_FIFO : 命名为“学生姓名拼音首字母缩写+logout”

1.1.3. 日志文件管理：统一存放在 LOGFILES 文件夹下

LOGFILES=/var/log/chat-logs/。

a) 服务器日志文件 server.log：放在文件夹 LOGFILES_SERVER 下

LOGFILES_SERVER=/var/log/chat-logs/server/

b) 用户日志文件存放目录：放在文件夹 LOGFILES_USERS 下

LOGFILES_USERS=/var/log/chat-logs/users/

1.2. 服务器首先读取配置文件，通过配置文件约束行为。

1.2.1. 根据配置文件创建众所周知的命名管道，采用多路复用技术（select()、epoll()或poll()）监听并读取以上信息，分别启动线程处理用户的注册请求、登录请求、聊天请求和退出请求，并将结果返回给用户，并做日志记录。服务器要求运行期间忽略 SIGCHLD、SIGTERM 等信号。服务器运行日志文件启动和终止信息记录在以下服务器日志文件中：

/var/log/chat-logs/server/server.log。

1.2.2. 为每个注册成功的用户建立独立日志文件，统一放在前面 1.1.3 节设置的用户日志文件存放目录 LOGFILES_USERS 下。日志文件收集用户如下信息：

- 注册事件（用户名，注册，时间）
- 登录事件（用户名，登录，时间）
- 退出事件（用户名，退出，时间）
- 总在线时长（用户名，在线时间）
- 成功发送信息的次数（发送者，接收者，cnt），聊天时，超过 5 次在用户名后加“*”，表示重要朋友
- 未成功发送的信息（发送者，接收者，时间，false）

日志文件只有超级用户可以读、写、不可运行。同组用户和其他用

户一律取消读、写、运行等权限。如果有未成功发送的信息，当接收者重新登录时服务器主动推送这些信息，并更改相应的“成功发送信息的次数”（发送者，接收者，时间, cnt）。

请在答卷完成以下内容：

1) （6 分）请通过 Shell 命令 ps 获取服务器进程状态并附上运行结果截图，**结合进程状态信息和源代码实现说明服务器进程是守护进程**，例如忽略特定信号 SIGCHLD、SIGTERM 的代码确保进程运行的健壮性等。

注意服务器命名：chatserver_学生姓名拼音首字母缩写_版本号

2) （24 分）请截图显示配置文件内容，结合服务器端相应的处理代码和 Shell 命令结果截图答题。

a)通过 Shell 命令 ls 获取 4 个公共 FIFO 的信息并附上运行结果截图，结合 4 个公共 FIFO 创建、打开的代码，**解释 Shell 命令 ls 运行结果特征字段含义**，例如，显示文件是 FIFO 的特征字段（4 分）。

b)请截图显示日志文件生成代码，并结合以下 Shell 命令运行结果展示有效日志文件属性及其内容（20 分）。

● 运行以下 Shell 命令并截取运行结果：（5 分）

```
$cat /var/log/chat-logs/server/server.log
```

服务器名字包括版本号启动，退出等内容

● 运行以下 Shell 命令并截取运行结果：（5 分）

```
$ls -l /var/log/chat-logs/users/
```

显示现有已经注册的用户日志文件及其权限

● 用 Shell 命令“ cat”展示其中一个日志文件中用户注册、登录、退出、在线总时长、与好友成功发送消息次数和未成功发送消息事件记录的日志信息（5 分）

● 未成功发送信息的管理等（难点）（5分）。

3) (10分)展示多路复用监听请求的代码、解释运行过程 `select()`、`epoll()` 或 `poll()` 的参数文件描述符集合的变化。**特别提醒，代码质量将作为评分点。**

2. 基本聊天功能展示（40分）：用户可以通过客户端注册、登录、收发信息、查看服务器信息、退出。用户注册时提供用户名和密码，不允许不同用户使用同一用户名注册。客户端将为每个用户建立一个以用户名为名字的专用命名管道，用户 A 发送给用户 B 的信息首先发送给服务器，然后服务器通过 B 的专用命名管道发送给用户 B。

1) (5分)守护进程各线程通过共享内存共同维护和使用一个在线用户列表，包含用户名、密码及其私有命名管道。**请注意其中的互斥访问和线程安全，请截图并讨论你的线程安全实现。**

2) (5分)截图用户注册的线程函数，截屏显示以下过程：启动4个终端，分别用以下用户名依次注册：学生姓名拼音_1、学生姓名拼音_2、学生姓名拼音_3、学生姓名拼音_3。守护进程将注册不成功的信息返回给相应的用户。

3) (10分)截图客户端**登录**线程函数，截屏显示以下登录过程：启动4个终端，分别用以下用户名登录：学生姓名拼音_1、学生姓名拼音_2、学生姓名拼音_3、学生姓名拼音_4。守护进程将成功登录的用户信息及所有在线的总人数及用户名显示给所有的用户，不成功的信息仅返回给不成功的用户端。(5分)名为“学生姓名拼音_2”的用户退出，守护进程将其退出的信息及剩余在线用户的信息显示给剩余的用户。保持名为“学生姓名拼音_2”用户的退出状态做4) (5分)。

4) (20分)截图用户**消息发送**过程，一对一发送(10分)：“学生

姓名拼音_1”的用户给“学生姓名拼音_2”的用户发送信息“Hi, let's play badminton?”4次。截取以上5次显示结果。一对多发送（5分）：“学生姓名拼音_1”的用户给“学生姓名拼音_2”、“学生姓名拼音_3”的用户发送信息“Hi, let's play badminton?”，截屏三个用户的收发信息情况，包括成功和不成功的情况（可能较难），成功将显示两人总消息发送次数，超过5次在用户名后加“*”，表示重要朋友。最后名为“学生姓名拼音_2”的用户登录系统，收到前面的信息，展示发送的时间（5分）。

3. 基于线程池的多用户聊天服务器设计与实现（20分）（难点）：系统参考架构图见图2，请结合代码介绍线程池的管理。

a) （5分）修改服务器版本号为1.2.0；

请展示配置文件内容并运行以下Shell命令并截取运行结果：

```
$cat /var/log/chat-logs/server/server.log
```

b) （5分）在配置文件里新增线程池规模参数POOLSIZE，值为10，服务器启动时根据配置文件新建POOLSIZE个线程。请展示配置文件内容并服务器代码展示线程池规模。

c) （10分）运行期接收请求时分派线程，请求处理完后回收线程用于再分派，特别地，刚回收的线程倾向于下一次被分派。请给出线程池管理数据结构设计、线程池线程状态管理算法设计和实现等，并附上相应的运行展示。线程池中线程被分派和回收的时间记录在日志/var/log/chat-logs/server/threads.log。截图展示日志信息。

特别提醒：代码质量和是否按时提交是本题评分点之一，请结合程序的正确性、线程的安全性、用户信息的安全性等讨论，迟交的同学本题分数直

接减分，减分范围 0~20 分，以确保总评成绩不为 A+ 为准。即如果该生整体成绩为 A, 虽然迟交，不扣分。如果该学生平时成绩基本满分，期末答卷也接近满分，但迟交，则本题将扣至该生不拿 A+ 为止。

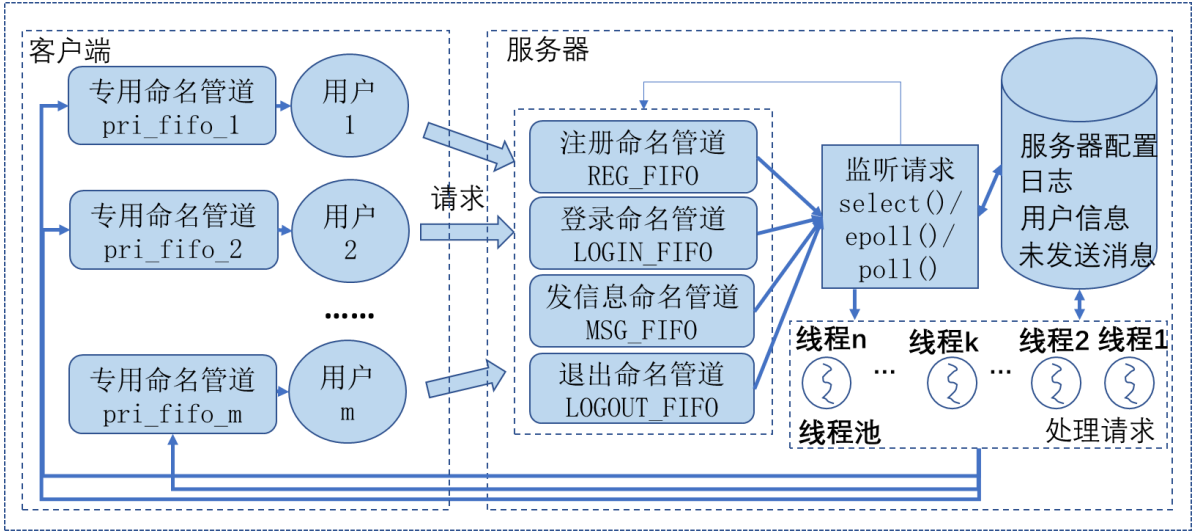


图 2. 基于线程池的单机多用户聊天系统参考架构图

选题(二) 开源贡献

国内外操作系统开源社区贡献,例如 Linux Kernel、openEuler。关于 Linux Kernel 开源社区贡献,可以阅读 2024 学生华为布道师、2019 级学长曹弈轩的经验¹:

<https://developer.huawei.com/home/program/advocates/member/7f07f1011f7d48b0bc989f8712919d79>

关于 openEuler 开源社区贡献,可以阅读 2024 学生华为布道师、2018 级学长黎浩然的经验²:

<https://developer.huawei.com/home/program/advocates/member/24a870748f094d1aafb8628f295b3d7b>

开源贡献思路获取(仅供参考,不局限): openEuler 开源实习官网,领取一项任务并完成,贡献社区后提交任务报告:

(<https://www.openeuler.org/zh/internship/>),

开源之夏-openEuler

<https://summer-ospp.ac.cn/org/orgdetail/b9770f4d-f586-4373-bdac-ce6389065946?lang=en>

选取一个点,提交一个开源贡献即可,不需要完成整个项目,也不需要参与项目。基于完成的开源贡献,完成答卷:

- 动机(问题定义)(40分)

源代码介绍,存在问题

- 解决方案(60分)

1) 详细介绍你的解决方案(10分)

¹ 2021-2022-1 “系统编程”课程学生,成绩 A+

² 2020-2021-1 “系统编程”课程学生,成绩 A+

- 2) 提交 PR: 截取 PR 提交的截屏; (10 分)
- 3) 截屏你的 PR 被接收的截屏 (10 分)
- 4) 展示修改后的代码 (10 分)
- 5) 修改后的代码的运行 (20 分)

选题(三) 创新实践

贵州赤水乌骨鸡是中国地理标志农产品，“养好一只鸡，带富一方人”，是地方经济与文化传承的载体。然而，乌骨鸡养殖仍然采用传统散养方式，存在养殖风险高、人工负荷重等问题。基于端-边-云协同、人工智能助力的智慧养殖平台服务可助农减负增收。目前贵州长庚农业智慧养殖基地部署了100多摄像头，为构建智慧养殖提供了必要的基础设施。乌骨鸡智慧养殖平台为养殖户提供了“查看鸡舍”等直播服务，让农户随时随地通过手机查看鸡舍，简化了巡检工作，极大减少了农民巡检的负担。但是它们也让每个养殖户暴露在视频中，没有了隐私。图3-1展示面向乌骨鸡智慧养殖的隐私保护“查看鸡舍”服务系统框架图，图3-2展示的是基于单线程/多线程进程模型实现隐私保护“查看鸡舍”服务。多线程和线程池技术可实现多用户并行服务，是提高系统性能的有效手段。图3-1中部分组件例示基于多线程和线程池技术提升系统性能，但有些组件的设计还需要进一步优化。

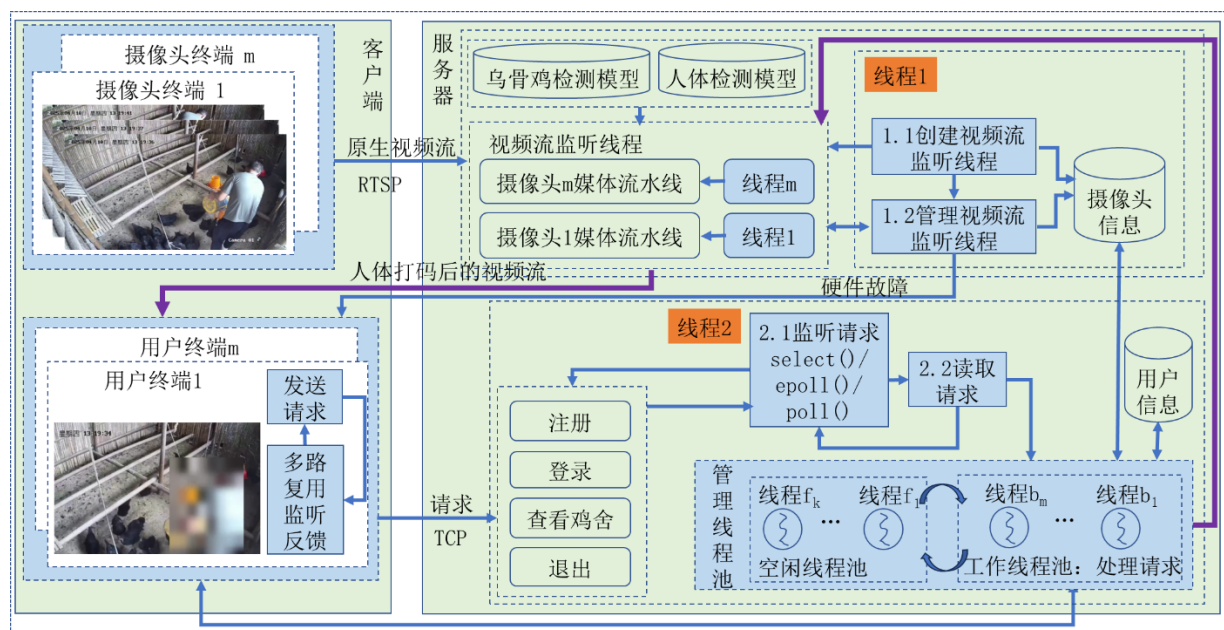


图 3-1. 面向乌骨鸡智慧养殖的隐私保护“查看鸡舍”服务系统框架图

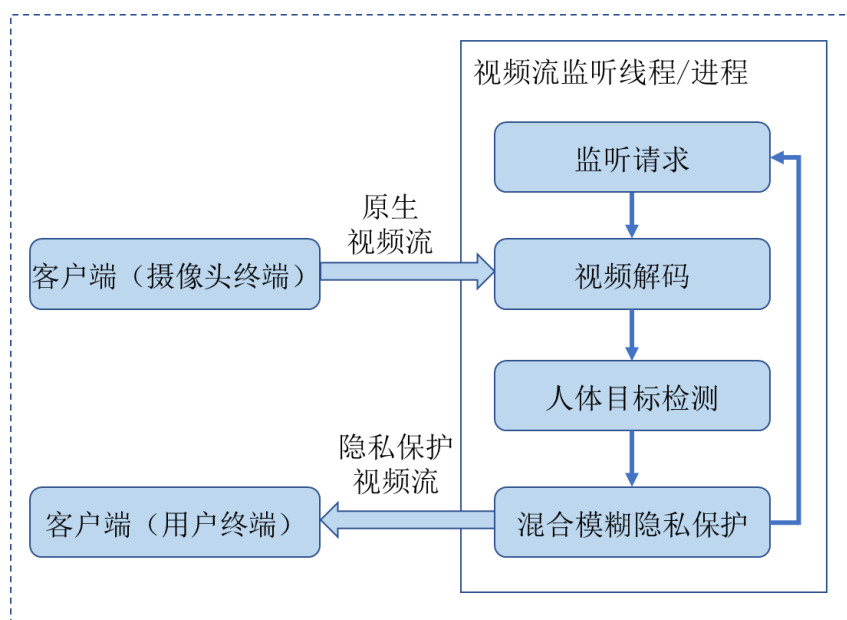


图 3-2. 基于单线程/单线程进程模型实现隐私保护的“查看鸡舍”服务



图 3-3. 黑色皮鞋、头发都会被乌骨鸡目标检测算法监测为乌骨鸡

项目组使用 Python 初步实现了面向乌骨鸡智慧养殖的隐私保护“查看鸡舍”服务的源代码，但采用的单线程进程模型，性能亟需提高；另外，目前的乌骨鸡监测和人体监测精度不高，混合模糊把附近的乌骨鸡也打码了。

以下初步小结图 3-2 中的视频流处理问题和优化方向：

- 1) （30 分）视频流串行处理，视频解码、人体目标监测、混合模糊隐私保护需要较长时，用户端视频展示卡顿严重，可采用缓冲区、多线程等技术优化人体检测和打码。
- 2) （50 分）目前人体区域全部打码，如果监测到乌骨鸡，乌骨鸡不应该

被打码。但因为乌骨鸡通体乌黑，给目标监测算法带来很大的挑战，如图 3-3 所示，基于 YOLOv8 目标检测，养殖户脚穿的黑色皮鞋，头上乌黑的头发都会被检测为乌骨鸡。请优化乌骨鸡目标检测算法，或设计不打码乌骨鸡的人体打码算法。

3) (20 分) 请将现有单线程进程实现的服务改为图 3-1 中的多线程+线程池服务，并参考以下规则设计服务器线程：

1) 线程的创建、销毁以及异常处理

2) 线程 1 的“管理视频流监听线程”组件，当某个摄像头的连接断开时，重新连接 k 次（配置文件设置），k 次失败发送硬件异常预警（可调用空函数，结合代码展示），回收相应线“视频流监听线程”程资源的回收。

3) 线程 1 “创建视频流监听线程”中，创建线程的数量按照实际部署摄像头数量：MAX_CAMERAS，该值在配置文件中配置。

请组队完成以上优化工作（最多 4 人），展示配置文件，运行结果和代码，并给出分工合作内容。4 人独立提交完整答卷。