

Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

Курсовой проект по курсу

«Операционные системы»

Группа: М8О-215Б-23

Студент: Кармишен Е.С.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 06.03.25

Москва, 2025

Постановка задачи

Вариант 22

Цель курсового проекта

1. Приобретение практических навыков в использовании знаний, полученных в течении курса
2. Проведение исследования в выбранной предметной области

Задание

Необходимо спроектировать и реализовать программный прототип в соответствии с выбранным вариантом. Произвести анализ и сделать вывод на основании данных, полученных при работе программного прототипа.

Необходимо создать собственный сервер сообщений. В качестве механизма передачи сообщений возможно использовать следующие:

1. Pipes
2. Sockets
3. Files/Shared memory

При работе с сервером сообщений необходимо предусмотреть следующие механизмы (в зависимости от варианта):

- Долговременное хранение сообщений
- Транзактивность
- Система имен очередей
- Возможность настройки переадресации сообщений по фильтрам
- Приоритеты сообщений
- И другие возможные, в зависимости от варианта

Основные операции, которые должен поддерживать сервер сообщений и библиотека по работе с ним:

- CreateQueue
- DeleteQueue
- ConnectToQueue
- Push (Send)
- Pop/Top (Receive)

Клиент-серверная система для передачи мгновенных сообщений. Базовый функционал должен быть следующим:

- Клиент может присоединиться к серверу, введя логин
- Клиент может отправить сообщение другому клиенту по его логину
- Клиент в реальном времени принимает сообщения от других клиентов

22 Необходимо предусмотреть возможность создания «групповых чатов». Связь между сервером и клиентом должна быть реализована при помощи pipe'ов

Введение

В данной курсовой работе рассматривается разработка клиент-серверной системы для обмена сообщениями с использованием именованных каналов (FIFO), процессов и потоков. Программа позволяет пользователям обмениваться личными и групповыми сообщениями в реальном времени. Основная цель работы — изучить механизмы межпроцессного взаимодействия (IPC) и многопоточности в операционных системах на примере реализации системы чата.

Технологии, используемые при решении:

Именованные каналы (FIFO)

Именованные каналы (FIFO) — это механизм межпроцессного взаимодействия (IPC), который позволяет процессам обмениваться данными через файловую систему. В отличие от неименованных каналов (pipe), именованные каналы существуют как файлы и могут использоваться независимо от родственных процессов.

Процессы

Процесс — это экземпляр выполняющейся программы. Каждый процесс имеет собственное адресное пространство и ресурсы.

В данной программе клиенты и сервер — это отдельные процессы, которые взаимодействуют через именованные каналы.

Потоки

Поток — это легковесный процесс, который разделяет адресное пространство с другими потоками в рамках одного процесса.

В данной программе каждый клиент создает отдельный поток для чтения входящих сообщений, что позволяет обрабатывать сообщения асинхронно.

Алгоритм работы программы:

1. Запуск сервера

1. Сервер создает именованный канал (FIFO) `/tmp/chat_server` для приема команд от клиентов.
2. Сервер открывает FIFO для чтения и переходит в режим ожидания команд.

2. Подключение клиента

1. Клиент запускается и запрашивает у пользователя логин.
2. Клиент подключается к серверу, отправляя команду `CONNECT <логин>` через FIFO сервера.
3. Сервер:
 - Регистрирует клиента.
 - Создает именованный канал для клиента (`/tmp/chat_client_<логин>`).
 - Сохраняет информацию о клиенте (логин, имя FIFO, файловый дескриптор).
4. Клиент открывает свой FIFO для чтения и запускает отдельный поток для приема сообщений.

3. Работа клиента

1. Клиент предоставляет пользователю меню команд:
 - `create_group <имя_группы>`: Создает новую группу.
 - `join_group <имя_группы>`: Присоединяется к существующей группе.

- send <логин_получателя> <сообщение>: Отправляет личное сообщение.
 - send_group <имя_группы> <сообщение>: Отправляет сообщение в группу.
 - exit: Отключается от сервера.
2. Клиент отправляет команды серверу через FIFO сервера.
-

4. Обработка команд сервером

1. Сервер читает команды из своего FIFO.
 2. В зависимости от команды:
 - CONNECT:
 - Регистрирует клиента.
 - Создает FIFO для клиента.
 - SEND:
 - Находит получателя по логину.
 - Отправляет сообщение через FIFO получателя.
 - CREATE_GROUP:
 - Создает новую группу и добавляет ее в список групп.
 - JOIN_GROUP:
 - Добавляет клиента в список участников группы.
 - SEND_GROUP:
 - Находит группу.
 - Отправляет сообщение всем участникам группы, кроме отправителя.
 - DISCONNECT:
 - Удаляет клиента из списка.
 - Закрывает и удаляет FIFO клиента.
-

5. Получение сообщений клиентом

1. В отдельном потоке клиент читает данные из своего FIFO.
 2. Если получено сообщение, оно выводится на экран.
-

6. Завершение работы

1. Клиент отправляет команду DISCONNECT и завершает работу.
2. Сервер:
 - Удаляет клиента из списка.
 - Закрывает и удаляет FIFO клиента.
3. При завершении сервера (например, по Ctrl+C):
 - Закрывает и удаляет FIFO сервера.
 - Освобождает все ресурсы.

Заключение

В ходе выполнения курсовой работы была разработана клиент-серверная система для обмена сообщениями с использованием именованных каналов (FIFO), процессов и потоков. Программа демонстрирует эффективное использование механизмов межпроцессного взаимодействия и многопоточности для реализации системы чата.

Основные результаты:

- Изучены и применены на практике именованные каналы (FIFO).
- Реализована многопоточная обработка входящих сообщений.
- Обеспечена синхронизация доступа к общим данным с помощью мьютексов.