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-209БВ-24

Студент: Галич А.П.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 18.12.25

Москва, 2025

Постановка задачи

Вариант 22.

Создать собственный клиент быстрых сообщений (возможно, и сервер – зависит от выбранной архитектуры), который бы работали в рамках сети. Также есть возможность создания собственного клиента быстрых сообщений для уже существующих систем обмена сообщений

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

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

Общий метод и алгоритм решения

Использованные системные вызовы:

- unlink() – удаление именованного канала
- mkfifo() – создание именованного канала
- signal() – установка обработчика сигналов
- open() - открытие файла/канала с флагами доступа
- close() - закрытие файлового дескриптора
- read() - чтение данных из файлового дескриптора
- write() - запись данных в файловый дескриптор

Общий метод

Метод решения основан на реализации клиент-серверной архитектуры с использованием именованных каналов (FIFO) для межпроцессного взаимодействия. Сервер выступает в роли центрального узла, который обрабатывает все подключения клиентов, управляет пользователями и группами, а также обеспечивает маршрутизацию сообщений. Каждый клиент создает свой собственный именованный канал для получения входящих сообщений, в то время как сервер использует общий канал для приема команд от всех клиентов.

Алгоритм работы начинается с инициализации сервера, который создает свой канал и переходит в режим ожидания команд. Клиент при запуске запрашивает логин пользователя, создает уникальный канал и отправляет серверу команду регистрации. Сервер проверяет уникальность логина и добавляет клиента в список активных пользователей. После успешного подключения клиент запускает отдельный поток для приема сообщений из своего канала, что позволяет обрабатывать входящие сообщения асинхронно.

Основной цикл работы клиента заключается в обработке пользовательских команд, которые преобразуются в соответствующие запросы к серверу. Сервер в непрерывном цикле считывает команды из своего канала, анализирует их тип и выполняет соответствующие действия: регистрацию новых пользователей, отключение клиентов, отправку личных сообщений, создание и управление группами, а также рассылку групповых сообщений. Для личных сообщений сервер находит получателя по логину и пересылает сообщение через его канал. Для групповых сообщений сервер определяет всех участников группы, кроме отправителя, и отправляет сообщение каждому из них.

Система обеспечивает обработку ошибок, включая проверку онлайн-статуса пользователей, существования групп и доступности клиентских каналов. При завершении работы клиент отправляет команду выхода, сервер удаляет его из списка активных пользователей, после чего клиент закрывает свой канал и завершает работу. Сервер корректно завершает работу при получении сигнала прерывания, закрывая все каналы и освобождая ресурсы. Весь обмен данными осуществляется через именованные каналы, что обеспечивает надежную коммуникацию между независимыми процессами в рамках одной системы.

Код программы

server.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <signal.h>
#include <errno.h>

#define MAX_CLIENTS 50
#define MAX_GROUPS 20
#define MAX_GROUP_MEMBERS 20
```

```c
#define MAX_LOGIN_LEN 32
#define MAX_MSG_LEN 1024
#define SERVER_PIPE "/tmp/msg_server_pipe"

typedef struct {
    char login[MAX_LOGIN_LEN];
    int active;
    char pipe_name[64];
} Client;

typedef struct {
    char name[MAX_LOGIN_LEN];
    char members[MAX_GROUP_MEMBERS][MAX_LOGIN_LEN];
    int member_count;
    int active;
} Group;

Client clients[MAX_CLIENTS];
Group groups[MAX_GROUPS];
int running = 1;

void handle_sigint(int sig) {
    running = 0;
}

int find_client(const char *login) {
    for (int i = 0; i < MAX_CLIENTS; i++) {
        if (clients[i].active && strcmp(clients[i].login, login) == 0) {
            return i;
        }
    }
    return -1;
}

int find_group(const char *name) {
    for (int i = 0; i < MAX_GROUPS; i++) {
        if (groups[i].active && strcmp(groups[i].name, name) == 0) {
            return i;
        }
    }
    return -1;
}

void send_to_client(const char *pipe_name, const char *message) {
    int fd = open(pipe_name, O_WRONLY | O_NONBLOCK);
    if (fd != -1) {
        write(fd, message, strlen(message) + 1);
        close(fd);
    }
}

void handle_login(const char *login) {
    if (find_client(login) != -1) {
        printf("Клиент %s уже подключен\n", login);
```

```
        return;
    }

    for (int i = 0; i < MAX_CLIENTS; i++) {
        if (!clients[i].active) {  // если клиент не активен
            strcpy(clients[i].login, login);
            snprintf(clients[i].pipe_name, 64, "/tmp/msg_client_%s", login); // создаем
имя канал клиента
            clients[i].active = 1;
            printf("Клиент %s подключился\n", login);

            char response[MAX_MSG_LEN];
            snprintf(response, MAX_MSG_LEN, "SERVER: Успешное подключение как %s",
login);
            send_to_client(clients[i].pipe_name, response);
            return;
        }
    }
    printf("Достигнут лимит клиентов\n");
}

void handle_logout(const char *login) {
    int idx = find_client(login);
    if (idx != -1) {
        clients[idx].active = 0;
        printf("Клиент %s отключился\n", login);
    }
}

void handle_message(const char *from, const char *to, const char *msg) { // отправка соо
от кому
    int idx = find_client(to);
    if (idx == -1) {
        printf("Получатель %s не найден\n", to);
        int sender_idx = find_client(from);
        if (sender_idx != -1) {
            char err[MAX_MSG_LEN];
            snprintf(err, MAX_MSG_LEN, "ERROR: Пользователь %s не в сети", to);
            send_to_client(clients[sender_idx].pipe_name, err);
        }
        return;
    }

    char full_msg[MAX_MSG_LEN];
    snprintf(full_msg, MAX_MSG_LEN, "[%s]: %s", from, msg);
    send_to_client(clients[idx].pipe_name, full_msg);
    printf("Сообщение от %s к %s: %s\n", from, to, msg);
}

void handle_create_group(const char *creator, const char *group_name) {
    if (find_group(group_name) != -1) {
        int idx = find_client(creator);
        if (idx != -1) {
            char err[MAX_MSG_LEN];
```

```
            snprintf(err, MAX_MSG_LEN, "ERROR: Группа %s уже существует", group_name);
            send_to_client(clients[idx].pipe_name, err);
        }
        return;
    }

    for (int i = 0; i < MAX_GROUPS; i++) {
        if (!groups[i].active) {
            strcpy(groups[i].name, group_name);
            strcpy(groups[i].members[0], creator);
            groups[i].member_count = 1;
            groups[i].active = 1;
            printf("Группа %s создана пользователем %s\n", group_name, creator);

            int idx = find_client(creator);
            if (idx != -1) {
                char response[MAX_MSG_LEN];
                snprintf(response, MAX_MSG_LEN, "SERVER: Группа %s создана", group_name);
                send_to_client(clients[idx].pipe_name, response);
            }
            return;
        }
    }
}

void handle_join_group(const char *login, const char *group_name) {
    int g_idx = find_group(group_name);
    if (g_idx == -1) {
        int idx = find_client(login);
        if (idx != -1) {
            char err[MAX_MSG_LEN];
            snprintf(err, MAX_MSG_LEN, "ERROR: Группа %s не найдена", group_name);
            send_to_client(clients[idx].pipe_name, err);
        }
        return;
    }

    for (int i = 0; i < groups[g_idx].member_count; i++) {
        if (strcmp(groups[g_idx].members[i], login) == 0) {
            int idx = find_client(login);
            if (idx != -1) {
                char err[MAX_MSG_LEN];
                snprintf(err, MAX_MSG_LEN, "ERROR: Вы уже в группе %s", group_name);
                send_to_client(clients[idx].pipe_name, err);
            }
            return;
        }
    }

    if (groups[g_idx].member_count < MAX_GROUP_MEMBERS) {
        strcpy(groups[g_idx].members[groups[g_idx].member_count], login);
        groups[g_idx].member_count++;
        printf("%s присоединился к группе %s\n", login, group_name);
```

```
        int idx = find_client(login);
        if (idx != -1) {
            char response[MAX_MSG_LEN];
            snprintf(response, MAX_MSG_LEN, "SERVER: Вы присоединились к группе %s",
group_name);
            send_to_client(clients[idx].pipe_name, response);
        }
    }
}

void handle_group_message(const char *from, const char *group_name, const char *msg) {
    int g_idx = find_group(group_name);
    if (g_idx == -1) {
        int idx = find_client(from);
        if (idx != -1) {
            char err[MAX_MSG_LEN];
            snprintf(err, MAX_MSG_LEN, "ERROR: Группа %s не найдена", group_name);
            send_to_client(clients[idx].pipe_name, err);
        }
        return;
    }

    char full_msg[MAX_MSG_LEN];
    snprintf(full_msg, MAX_MSG_LEN, "[%s@%s]: %s", from, group_name, msg);

    for (int i = 0; i < groups[g_idx].member_count; i++) {
        if (strcmp(groups[g_idx].members[i], from) != 0) {
            int c_idx = find_client(groups[g_idx].members[i]);
            if (c_idx != -1) {
                send_to_client(clients[c_idx].pipe_name, full_msg);
            }
        }
    }
    printf("Сообщение от %s в группу %s: %s\n", from, group_name, msg);
}

int main() {
    signal(SIGINT, handle_sigint);

    memset(clients, 0, sizeof(clients));
    memset(groups, 0, sizeof(groups));

    unlink(SERVER_PIPE);
    if (mkfifo(SERVER_PIPE, 0666) == -1) {
        perror("mkfifo");
        exit(1);
    }

    printf("Сервер запущен. Ожидание клиентов...\n");

    int server_fd = open(SERVER_PIPE, O_RDONLY | O_NONBLOCK);
    if (server_fd == -1) {
        perror("open");
        exit(1);
```

```
    }

    char buffer[MAX_MSG_LEN * 2];
    while (running) {
        int n = read(server_fd, buffer, sizeof(buffer) - 1);
        if (n > 0) {
            buffer[n] = '\0';

            char arg1[MAX_LOGIN_LEN], arg2[MAX_LOGIN_LEN];
            char msg[MAX_MSG_LEN];

            if (sscanf(buffer, "LOGIN %s", arg1) == 1) {
                handle_login(arg1);
            }
            else if (sscanf(buffer, "LOGOUT %s", arg1) == 1) {
                handle_logout(arg1);
            }
            else if (sscanf(buffer, "MSG %s %s %[^\n]", arg1, arg2, msg) == 3) {
                handle_message(arg1, arg2, msg);
            }
            else if (sscanf(buffer, "CREATE_GROUP %s %s", arg1, arg2) == 2) {
                handle_create_group(arg1, arg2);
            }
            else if (sscanf(buffer, "JOIN_GROUP %s %s", arg1, arg2) == 2) {
                handle_join_group(arg1, arg2);
            }
            else if (sscanf(buffer, "GROUP_MSG %s %s %[^\n]", arg1, arg2, msg) == 3) {
                handle_group_message(arg1, arg2, msg);
            }
        }
        usleep(10000);
    }

    close(server_fd);
    unlink(SERVER_PIPE);
    printf("\nСервер остановлен\n");

    return 0;
}
```

client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
```

```c
#include <sys/types.h>
#include <pthread.h>
#include <signal.h>

#define MAX_COMMAND_LEN 2048
#define MAX_LOGIN_LEN 32
#define MAX_MSG_LEN 1024
#define SERVER_PIPE "/tmp/msg_server_pipe"

char my_login[MAX_LOGIN_LEN];
char my_pipe[64];
int running = 1;

void *receive_thread(void *arg) {
    int fd = open(my_pipe, O_RDONLY);
    if (fd == -1) {
        perror("open client pipe");
        return NULL;
    }

    char buffer[MAX_MSG_LEN];
    while (running) {
        int n = read(fd, buffer, sizeof(buffer) - 1);
        if (n > 0) {
            buffer[n] = '\0';
            printf("\n%s\n> ", buffer);
            fflush(stdout);
        }
    }

    close(fd);
    return NULL;
}

void send_to_server(const char *message) {
    int fd = open(SERVER_PIPE, O_WRONLY);
    if (fd == -1) {
        printf("Ошибка: не удается подключиться к серверу\n");
        return;
    }
    write(fd, message, strlen(message) + 1);
    close(fd);
}

void print_help() {
    printf("\nДоступные команды:\n");
    printf("  /msg <логин> <сообщение>     - отправить личное сообщение\n");
    printf("  /create <имя_группы>         - создать группу\n");
    printf("  /join <имя_группы>           - присоединиться к группе\n");
    printf("  /gmsg <имя_группы> <сообщ>   - отправить сообщение в группу\n");
    printf("  /help                        - показать эту справку\n");
    printf("  /quit                        - выйти\n\n");
}
```

```c
void handle_sigint(int sig) {
    running = 0;
}

int main() {
    signal(SIGINT, handle_sigint);

    printf("=== Клиент мгновенных сообщений ===\n");
    printf("Введите ваш логин: ");
    fgets(my_login, MAX_LOGIN_LEN, stdin);
    my_login[strcspn(my_login, "\n")] = 0;

    if (strlen(my_login) == 0) {
        printf("Некорректный логин\n");
        return 1;
    }

    snprintf(my_pipe, 64, "/tmp/msg_client_%s", my_login);
    unlink(my_pipe);

    if (mkfifo(my_pipe, 0666) == -1) {
        perror("mkfifo");
        return 1;
    }

    char login_msg[MAX_MSG_LEN];
    snprintf(login_msg, MAX_MSG_LEN, "LOGIN %s", my_login);
    send_to_server(login_msg);

    pthread_t recv_thread;
    pthread_create(&recv_thread, NULL, receive_thread, NULL);

    printf("\nВы подключены как '%s'\n", my_login);
    print_help();

    char input[MAX_MSG_LEN];
    char command[2048];

    while (running) {
        printf("> ");
        fflush(stdout);

        if (fgets(input, MAX_MSG_LEN, stdin) == NULL) {
            break;
        }

        input[strcspn(input, "\n")] = 0;

        if (strlen(input) == 0) {
            continue;
        }

        if (strcmp(input, "/quit") == 0) {
            break;
```

```
        }
        else if (strcmp(input, "/help") == 0) {
            print_help();
        }
        else if (strncmp(input, "/msg ", 5) == 0) {
            char to[MAX_LOGIN_LEN];
            char msg[MAX_MSG_LEN];

            if (sscanf(input + 5, "%s %[^\n]", to, msg) == 2) {
                snprintf(command, sizeof(command), "MSG %s %s %s", my_login, to, msg);
                send_to_server(command);
                printf("Сообщение отправлено пользователю %s\n", to);
            } else {
                printf("Использование: /msg <логин> <сообщение>\n");
            }
        }
        else if (strncmp(input, "/create ", 8) == 0) {
            char group_name[MAX_LOGIN_LEN];

            if (sscanf(input + 8, "%s", group_name) == 1) {
                snprintf(command, MAX_MSG_LEN, "CREATE_GROUP %s %s", my_login,
group_name);
                send_to_server(command);
            } else {
                printf("Использование: /create <имя_группы>\n");
            }
        }
        else if (strncmp(input, "/join ", 6) == 0) {
            char group_name[MAX_LOGIN_LEN];

            if (sscanf(input + 6, "%s", group_name) == 1) {
                snprintf(command, MAX_MSG_LEN, "JOIN_GROUP %s %s", my_login, group_name);
                send_to_server(command);
            } else {
                printf("Использование: /join <имя_группы>\n");
            }
        }
        else if (strncmp(input, "/gmsg ", 6) == 0) {
            char group_name[MAX_LOGIN_LEN];
            char msg[MAX_MSG_LEN];

            if (sscanf(input + 6, "%s %[^\n]", group_name, msg) == 2) {
                snprintf(command, sizeof(command), "GROUP_MSG %s %s %s", my_login,
group_name, msg);
                send_to_server(command);
                printf("Сообщение отправлено в группу %s\n", group_name);
            } else {
                printf("Использование: /gmsg <имя_группы> <сообщение>\n");
            }
        }
        else {
            printf("Неизвестная команда. Введите /help для справки\n");
        }
    }
```

```
    running = 0;

    char logout_msg[MAX_MSG_LEN];
    snprintf(logout_msg, MAX_MSG_LEN, "LOGOUT %s", my_login);
    send_to_server(logout_msg);

    pthread_join(recv_thread, NULL);

    unlink(my_pipe);
    printf("\nОтключение от сервера...\n");

    return 0;
}
```

Протокол работы программы

./server

kishaki@416:~/kp$ ./server

Сервер запущен. Ожидание клиентов...

Клиент bob подключился

Клиент bib подключился

Сообщение от bob к bib: privet

Сообщение от bib к bob: privet bob

Группа druzi создана пользователем bib

bob присоединился к группе druzi

Сообщение от bib в группу druzi: hello group

Клиент bib отключился

Клиент bob отключился

^C

Сервер остановлен

Первый ./client

kishaki@416:~/kp$ ./client

=== Клиент мгновенных сообщений ===

Введите ваш логин: bob

Вы подключены как 'bob'

Доступные команды:
  /msg <логин> <сообщение>   - отправить личное сообщение
  /create <имя_группы>   - создать группу
  /join <имя_группы>   - присоединиться к группе
  /gmsg <имя_группы> <сообщ>   - отправить сообщение в группу
  /help   - показать эту справку
  /quit   - выйти

>
SERVER: Успешное подключение как bob
> /msg bib privet
Сообщение отправлено пользователю bib
>
[bib]: privet bob
> /join druzi
>
SERVER: Вы присоединились к группе druzi
>
[bib@druzi]: hello group
> /quit

Отключение от сервера...

Второй ./client

kishaki@416:~/kp$ ./client
=== Клиент мгновенных сообщений ===

Введите ваш логин: bib

Вы подключены как 'bib'

Доступные команды:

/msg <логин> <сообщение> - отправить личное сообщение

/create <имя_группы> - создать группу

/join <имя_группы> - присоединиться к группе

/gmsg <имя_группы> <сообщ> - отправить сообщение в группу

/help - показать эту справку

/quit - выйти

>

SERVER: Успешное подключение как bib

>

[bob]: privet

> /msg bob privet bob

Сообщение отправлено пользователю bob

> /create druzi

>

SERVER: Группа druzi создана

> /gmsg druzi hello group

Сообщение отправлено в группу druzi

> /quit

Strace

server.c

```
$ strace -e '!clock_nanosleep,read' ./server
execve("./server", ["./server"], 0x7ffc11baca80 /* 29 vars */) = 0
brk(NULL)                               = 0x5b7d0f4bf000
```

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7b57cb966000

access("/etc/ld.so.preload", R_OK)   = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=37043, ...}) = 0

mmap(NULL, 37043, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7b57cb95c000

close(3)   = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7b57cb600000

mmap(0x7b57cb628000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7b57cb628000

mmap(0x7b57cb7b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7b57cb7b0000

mmap(0x7b57cb7ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7b57cb7ff000

mmap(0x7b57cb805000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7b57cb805000

close(3)   = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7b57cb959000

arch_prctl(ARCH_SET_FS, 0x7b57cb959740) = 0

set_tid_address(0x7b57cb959a10)   = 15262

set_robust_list(0x7b57cb959a20, 24)   = 0

rseq(0x7b57cb95a060, 0x20, 0, 0x53053053) = 0

mprotect(0x7b57cb7ff000, 16384, PROT_READ) = 0

mprotect(0x5b7d033e4000, 4096, PROT_READ) = 0

mprotect(0x7b57cb99e000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

munmap(0x7b57cb95c000, 37043)   = 0

rt_sigaction(SIGINT, {sa_handler=0x5b7d033e1389, sa_mask=[INT], sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x7b57cb645330}, {sa_handler=SIG_DFL, sa_mask=[], sa_flags=0}, 8) = 0

unlink("/tmp/msg_server_pipe")   = 0

mknodat(AT_FDCWD, "/tmp/msg_server_pipe", S_IFIFO|0666) = 0

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0

getrandom("\xa9\x49\xfe\xe0\x64\x79\xa1\x2e", 8, GRND_NONBLOCK) = 8

brk(NULL)   = 0x5b7d0f4bf000

brk(0x5b7d0f4e0000)   = 0x5b7d0f4e0000

write(1, "\320\241\320\265\321\200\320\262\320\265\321\200
\320\267\320\260\320\277\321\203\321\211\320\265\320\275. \320\236\320"..., 66Сервер запущен.
Ожидание клиентов...

) = 66

openat(AT_FDCWD, "/tmp/msg_server_pipe", O_RDONLY|O_NONBLOCK) = 3

write(1, "\320\232\320\273\320\270\320\265\320\275\321\202 bob
\320\277\320\276\320\264\320\272\320\273\321\216\321\207\320"..., 40Клиент bob подключился

) = 40

openat(AT_FDCWD, "/tmp/msg_client_bob", O_WRONLY|O_NONBLOCK) = 4

write(4, "SERVER: \320\243\321\201\320\277\320\265\321\210\320\275\320\276\320\265
\320\277\320\276\320\264\320"..., 59) = 59

close(4)   = 0

write(1, "\320\232\320\273\320\270\320\265\320\275\321\202 bib
\320\277\320\276\320\264\320\272\320\273\321\216\321\207\320"..., 40Клиент bib подключился

) = 40

openat(AT_FDCWD, "/tmp/msg_client_bib", O_WRONLY|O_NONBLOCK) = 4

write(4, "SERVER: \320\243\321\201\320\277\320\265\321\210\320\275\320\276\320\265
\320\277\320\276\320\264\320"..., 59) = 59

close(4)   = 0

openat(AT_FDCWD, "/tmp/msg_client_bob", O_WRONLY|O_NONBLOCK) = 4

write(4, "[bib]: privet\0", 14)   = 14

close(4)   = 0

write(1, "\320\241\320\276\320\276\320\261\321\211\320\265\320\275\320\270\320\265
\320\276\321\202 bib \320\272 b"..., 43Сообщение от bib к bob: privet

) = 43

write(1, "\320\223\321\200\321\203\320\277\320\277\320\260 druzi
\321\201\320\276\320\267\320\264\320\260\320\275\320"..., 65Группа druzi создана пользователем
bob

) = 65

openat(AT_FDCWD, "/tmp/msg_client_bob", O_WRONLY|O_NONBLOCK) = 4

write(4, "SERVER: \320\223\321\200\321\203\320\277\320\277\320\260 druzi \321\201\320\276\320"...,
42) = 42

close(4)   = 0

write(1, "bib
\320\277\321\200\320\270\321\201\320\276\320\265\320\264\320\270\320\275\320\270\320\273\321\20
1\321\217 \320"..., 53bib присоединился к группе druzi

) = 53

openat(AT_FDCWD, "/tmp/msg_client_bib", O_WRONLY|O_NONBLOCK) = 4

write(4, "SERVER: \320\222\321\213
\320\277\321\200\320\270\321\201\320\276\320\265\320\264\320\270\320\275\320"..., 64) = 64

close(4)   = 0

openat(AT_FDCWD, "/tmp/msg_client_bob", O_WRONLY|O_NONBLOCK) = 4

write(4, "[bib@druzi]: prievr\0", 20)   = 20

close(4)   = 0

write(1, "\320\241\320\276\320\276\320\261\321\211\320\265\320\275\320\270\320\265
\320\276\321\202 bib \320\262 \320"..., 58Сообщение от bib в группу druzi: prievr

) = 58

write(1, "\320\232\320\273\320\270\320\265\320\275\321\202 bib
\320\276\321\202\320\272\320\273\321\216\321\207\320\270\320"..., 38Клиент bib отключился

) = 38

write(1, "\320\232\320\273\320\270\320\265\320\275\321\202 bob
\320\276\321\202\320\272\320\273\321\216\321\207\320\270\320"..., 38Клиент bob отключился

) = 38

^Cstrace: Process 15262 detached

Сервер остановлен

Client1

strace -e trace=all ./client

execve("./client", ["./client"], 0x7ffd0dd31f10 /* 29 vars */) = 0

brk(NULL)   = 0x5ef491a57000

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x761b62d2b000

access("/etc/ld.so.preload", R_OK)   = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=37043, ...}) = 0

mmap(NULL, 37043, PROT_READ, MAP_PRIVATE, 3, 0) = 0x761b62d21000

close(3)   = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x761b62a00000

mmap(0x761b62a28000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x761b62a28000

mmap(0x761b62bb0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1b0000) = 0x761b62bb0000

mmap(0x761b62bff000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x761b62bff000

mmap(0x761b62c05000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x761b62c05000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x761b62d1e000

arch_prctl(ARCH_SET_FS, 0x761b62d1e740) = 0

set_tid_address(0x761b62d1ea10) = 15320

set_robust_list(0x761b62d1ea20, 24) = 0

rseq(0x761b62d1f060, 0x20, 0, 0x53053053) = 0

mprotect(0x761b62bff000, 16384, PROT_READ) = 0

mprotect(0x5ef48542d000, 4096, PROT_READ) = 0

mprotect(0x761b62d63000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

munmap(0x761b62d21000, 37043) = 0

rt_sigaction(SIGINT, {sa_handler=0x5ef48542b5b1, sa_mask=[INT],
sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x761b62a45330}, {sa_handler=SIG_DFL,
sa_mask=[], sa_flags=0}, 8) = 0

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x3), ...}) = 0

getrandom("\x07\xf7\x79\x8d\x28\xeb\x38\x1d", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x5ef491a57000

brk(0x5ef491a78000) = 0x5ef491a78000

write(1, "=== \320\232\320\273\320\270\320\265\320\275\321\202
\320\274\320\263\320\275\320\276\320\262\320\265\320\275\320"..., 61=== Клиент мгновенных
сообщений ===

) = 61

fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x3), ...}) = 0

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\262\320\260\321\210
\320\273\320\276\320\263\320\270\320\275"..., 34Введите ваш логин:) = 34

read(0, bob

"bob\n", 1024) = 4

unlink("/tmp/msg_client_bob") = 0

mknodat(AT_FDCWD, "/tmp/msg_client_bob", S_IFIFO|0666) = 0

openat(AT_FDCWD, "/tmp/msg_server_pipe", O_WRONLY) = 3

write(3, "LOGIN bob\0", 10) = 10

close(3) = 0

rt_sigaction(SIGRT_1, {sa_handler=0x761b62a99530, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO,
sa_restorer=0x761b62a45330}, NULL, 8) = 0

rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x761b621ff000

mprotect(0x761b62200000, 8388608, PROT_READ|PROT_WRITE) = 0

rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x761b629ff990, parent_tid=0x761b629ff990, exit_signal=0, stack=0x761b621ff000, stack_size=0x7fff80, tls=0x761b629ff6c0} => {parent_tid=[15343]}, 88) = 15343

rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

write(1, "\n\320\222\321\213
\320\277\320\276\320\264\320\272\320\273\321\216\321\207\320\265\320\275\321\213
\320\272\320\260\320"..., 40

Вы подключены как 'bob'

) = 40

write(1, "\n", 1

)   = 1

write(1, "\320\224\320\276\321\201\321\202\321\203\320\277\320\275\321\213\320\265
\320\272\320\276\320\274\320\260\320\275\320\264\321"..., 35Доступные команды:

) = 35

write(1, " /msg <\320\273\320\276\320\263\320\270\320\275>
<\321\201\320\276\320\276\320\261\321\211\320"..., 98 /msg <логин> <сообщение>   - отправить личное сообщение

) = 98

write(1, " /create <\320\270\320\274\321\217_\320\263\321\200\321\203\320\277\320\277\321\213>
"..., 70 /create <имя_группы>   - создать группу

) = 70

write(1, " /join <\320\270\320\274\321\217_\320\263\321\200\321\203\320\277\320\277\321\213>   "..., 87 /join <имя_группы>   - присоединиться к группе

) = 87

write(1, " /gmsg <\320\270\320\274\321\217_\320\263\321\200\321\203\320\277\320\277\321\213>
<\321"..., 101 /gmsg <имя_группы> <сообщ>   - отправить сообщение в группу

) = 101

write(1, " /help   -"..., 72 /help   - показать эту справку

) = 72

write(1, " /quit   -"..., 44 /quit   - выйти

) = 44

write(1, "\n", 1

)   = 1

futex(0x761b62c05710, FUTEX_WAKE_PRIVATE, 1

SERVER: Успешное подключение как bob

) = 1

> write(1, "> ", 2>) = 2

read(0,

[bib]: privet

> /create druzi

"/create druzi\n", 1024) = 14

openat(AT_FDCWD, "/tmp/msg_server_pipe", O_WRONLY) = 4

write(4, "CREATE_GROUP bob druzi\0", 23) = 23

close(4) = 0

write(1, "> ", 2>) = 2

read(0,

SERVER: Группа druzi создана

>

[bib@druzi]: prievr

> /quit

"/quit\n", 1024) = 6

openat(AT_FDCWD, "/tmp/msg_server_pipe", O_WRONLY) = 3

write(3, "LOGOUT bob\0", 11) = 11

close(3) = 0

unlink("/tmp/msg_client_bob") = 0

write(1, "\n", 1

) = 1

write(1, "\320\236\321\202\320\272\320\273\321\216\321\207\320\265\320\275\320\270\320\265 \320\276\321\202 \321\201\320\265\321\200"..., 44Отключение от сервера...

) = 44

exit_group(0) = ?

+++ exited with 0 +++

Client2

strace -e trace=all ./client

execve("./client", ["./client"], 0x7ffce6c06430 /* 29 vars */) = 0

brk(NULL) = 0x555876e0a000

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x760a02769000

access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=37043, ...}) = 0

mmap(NULL, 37043, PROT_READ, MAP_PRIVATE, 3, 0) = 0x760a0275f000

close(3)   = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x760a02400000

mmap(0x760a02428000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x760a02428000

mmap(0x760a025b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1b0000) = 0x760a025b0000

mmap(0x760a025ff000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x760a025ff000

mmap(0x760a02605000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x760a02605000

close(3)   = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x760a0275c000

arch_prctl(ARCH_SET_FS, 0x760a0275c740) = 0

set_tid_address(0x760a0275ca10)   = 15330

set_robust_list(0x760a0275ca20, 24)   = 0

rseq(0x760a0275d060, 0x20, 0, 0x53053053) = 0

mprotect(0x760a025ff000, 16384, PROT_READ) = 0

mprotect(0x55584d75c000, 4096, PROT_READ) = 0

mprotect(0x760a027a1000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

munmap(0x760a0275f000, 37043)   = 0

rt_sigaction(SIGINT, {sa_handler=0x55584d75a5b1, sa_mask=[INT],
sa_flags=SA_RESTORER|SA_RESTART, sa_restorer=0x760a02445330}, {sa_handler=SIG_DFL,
sa_mask=[], sa_flags=0}, 8) = 0

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x2), ...}) = 0

getrandom("\x0d\xde\xd8\x58\xf3\xb7\xc7\x34", 8, GRND_NONBLOCK) = 8

brk(NULL)   = 0x555876e0a000

brk(0x555876e2b000)   = 0x555876e2b000

write(1, "=== \320\232\320\273\320\270\320\265\320\275\321\202
\320\274\320\263\320\275\320\276\320\262\320\265\320\275\320"..., 61=== Клиент мгновенных
сообщений ===

) = 61

fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x2), ...}) = 0

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\262\320\260\321\210 \320\273\320\276\320\263\320\270\320\275"..., 34Введите ваш логин:) = 34

read(0, bib

"bib\n", 1024) = 4

unlink("/tmp/msg_client_bib") = 0

mknodat(AT_FDCWD, "/tmp/msg_client_bib", S_IFIFO|0666) = 0

openat(AT_FDCWD, "/tmp/msg_server_pipe", O_WRONLY) = 3

write(3, "LOGIN bib\0", 10) = 10

close(3) = 0

rt_sigaction(SIGRT_1, {sa_handler=0x760a02499530, sa_mask=[], sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x760a02445330}, NULL, 8) = 0

rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x760a01bff000

mprotect(0x760a01c00000, 8388608, PROT_READ|PROT_WRITE) = 0

rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x760a023ff990, parent_tid=0x760a023ff990, exit_signal=0, stack=0x760a01bff000, stack_size=0x7fff80, tls=0x760a023ff6c0} => {parent_tid=[15356]}, 88) = 15356

rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

write(1, "\n\320\222\321\213 \320\277\320\276\320\264\320\272\320\273\321\216\321\207\320\265\320\275\321\213 \320\272\320\260\320"..., 40

Вы подключены как 'bib'

) = 40

write(1, "\n", 1

) = 1

write(1, "\320\224\320\276\321\201\321\202\321\203\320\277\320\275\321\213\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321"..., 35Доступные команды:

) = 35

write(1, " /msg <\320\273\320\276\320\263\320\270\320\275> <\321\201\320\276\320\276\320\261\321\211\320"..., 98 /msg <логин> <сообщение> - отправить личное сообщение

) = 98

write(1, " /create <\320\270\320\274\321\217_\320\263\321\200\321\203\320\277\320\277\321\213> "..., 70 /create <имя_группы> - создать группу

) = 70

write(1, " /join <\320\270\320\274\321\217_\320\263\321\200\321\203\320\277\320\277\321\213> "..., 87 /join <имя_группы> - присоединиться к группе

) = 87

write(1, " /gmsg <\320\270\320\274\321\217_\320\263\321\200\321\203\320\277\320\277\321\213>
<\321"..., 101 /gmsg <имя_группы> <сообщ>   - отправить сообщение в группу

) = 101

write(1, " /help   -"..., 72 /help   - показать эту справку

) = 72

write(1, " /quit   -"..., 44 /quit   - выйти

) = 44

write(1, "\n", 1

)   = 1

write(1, "> ", 2>)   = 2

read(0,

SERVER: Успешное подключение как bib

> /msg bob privet

"/msg bob privet\n", 1024)   = 16

openat(AT_FDCWD, "/tmp/msg_server_pipe", O_WRONLY) = 4

write(4, "MSG bib bob privet\0", 19)   = 19

close(4)   = 0

write(1, "\320\241\320\276\320\276\320\261\321\211\320\265\320\275\320\270\320\265
\320\276\321\202\320\277\321\200\320\260\320\262\320"..., 69Сообщение отправлено пользователю
bob

) = 69

write(1, "> ", 2>)   = 2

read(0, /join druzi

"/join druzi\n", 1024)   = 12

openat(AT_FDCWD, "/tmp/msg_server_pipe", O_WRONLY) = 4

write(4, "JOIN_GROUP bib druzi\0", 21) = 21

close(4)   = 0

write(1, "> ", 2>)   = 2

read(0,

SERVER: Вы присоединились к группе druzi

> /gmsg druzi prievr

"/gmsg druzi prievr\n", 1024)   = 19

openat(AT_FDCWD, "/tmp/msg_server_pipe", O_WRONLY) = 4

write(4, "GROUP_MSG bib druzi prievr\0", 27) = 27

close(4)   = 0

write(1, "\320\241\320\276\320\276\320\261\321\211\320\265\320\275\320\270\320\265
\320\276\321\202\320\277\321\200\320\260\320\262\320"..., 62Сообщение отправлено в группу druzi

) = 62

write(1, "> ", 2>) = 2

read(0, /quit

"/quit\n", 1024) = 6

openat(AT_FDCWD, "/tmp/msg_server_pipe", O_WRONLY) = 3

write(3, "LOGOUT bib\0", 11) = 11

close(3) = 0

unlink("/tmp/msg_client_bib") = 0

write(1, "\n", 1

) = 1

write(1, "\320\236\321\202\320\272\320\273\321\216\321\207\320\265\320\275\320\270\320\265
\320\276\321\202 \321\201\320\265\321\200"..., 44Отключение от сервера...

) = 44

exit_group(0) = ?

+++ exited with 0 +++

Вывод

В ходе курсового проекта была реализована клиент-серверная система обмена сообщениями на основе именованных каналов (FIFO). Система обеспечивает как личную, так и групповую переписку между пользователями. Именованные каналы предоставляют простой и эффективный способ организации межпроцессного взаимодействия между независимыми приложениями в рамках одной системы. Ключевое преимущество FIFO в данном контексте — возможность каждого клиента создать собственный именованный канал (например, msg_client_bob), к которому сервер может подключаться для отправки сообщений. Реализация демонстрирует эффективную работу с механизмами межпроцессного взаимодействия, включая неблокирующий ввод-вывод для обеспечения отказоустойчивости сервера и многопоточность на стороне клиента для асинхронного приема сообщений