Пометка к заданию №13
Задание 1:
Реализовано 2 варианта реализации взаимодействия клиента с сервером через очереди сообщений:

- POSIX
- System V

POSIX:
Описание:
POSIX использует mq_open, mq_send, mq_receive и mq_unlink для работы с очередями сообщений. Сервер создает очередь /posix_msg_queue, отправляет сообщение "Hi!", ждет ответа от клиента ("Hello!") и затем удаляет очередь.

Работа программ:
POSIX-server:

1. Создаём очередь сообщений mq_open(QUEUE_NAME, O_CREAT | O_RDWR, 0644, &attr);
2. Отправляем сообщение клиенту mq_send(mq, "Hi!", ...);
3. Ждём ответ от клиента mq_receive(mq, buffer, MAX_SIZE, NULL);
4. Закрываем и удаляем очередь mq_unlink(QUEUE_NAME);

Ключевые функции:

- mq_open() – создание очереди
- mq_send() – отправка сообщения
- mq_receive() – получение сообщения
- mq_unlink() – удаление очереди

Запуск сервера:

```
mashina@mashina:~$ cd /home/mashina/zd-13-1
mashina@mashina:~/zd-13-1$ ./zd-13-1-posix-server
Сервер: ожидаю подключение клиента...
```

POSIX-client:

1. Подключаемся к очереди сервера mq_open(QUEUE_NAME, O_RDWR);
2. Получаем сообщение от сервера mq_receive(mq, buffer, MAX_SIZE, NULL);
3. Отправляем ответ ("Hello!") серверу mq_send(mq, "Hello!", ...);
4. Закрываем очередь mq_close(mq);

Ключевые функции:

- mq_open() – открыть очередь
- mq_receive() – получить сообщение
- mq_send() – отправить ответ

Запуск клиента:

```
mashina@mashina:~/zd-13-1$ ./zd-13-1-posix-client
Клиент: подключился к очереди сообщений.
Клиент: получено сообщение 'Client connected'
Клиент: отправил сообщение 'Hi!'
```

Что отображается на сервере:

```
Сервер: клиент подключился, отправляю сообщение...
Сервер: отправлено сообщение 'Hi!'
Сервер: получено сообщение от клиента 'Hi!'
mashina@mashina:~/zd-13-1$
```

System V:

Описание:

System V использует msgget, msgsnd, msgrcv и msgctl для работы с очередями. Очередь создается с ключом 1234, сервер отправляет "Hi!", ожидает "Hello!" от клиента, а затем удаляет очередь.

Работа программ:

System V-server:

1. Создаём очередь msgget(QUEUE_KEY, IPC_CREAT | 0666);
2. Ждём сообщение от клиента msgrcv(msgid, &msg, sizeof(msg.text), 1, 0);
3. Отправляем сообщение ("Hi!") клиенту msgsnd(msgid, &msg, sizeof(msg.text), 0);
4. Ждём ответ от клиента msgrcv(msgid, &msg, sizeof(msg.text), 3, 0);
5. Удаляем очередь msgctl(msgid, IPC_RMID, NULL);

Ключевые функции:

- msgget() – создание очереди
- msgrcv() – получение сообщения
- msgsnd() – отправка сообщения
- msgctl() – удаление очереди

System V-client:

1. Подключаемся к очереди msgget(QUEUE_KEY, 0666);
2. Отправляем серверу уведомление о подключении msgsnd(msgid, &msg, sizeof(msg.text), 0);

3. Получаем сообщение ("Hi!") от сервера msgrcv(msgid, &msg, sizeof(msg.text), 2, 0);
4. Отправляем ответ ("Hello!") серверу msgsnd(msgid, &msg, sizeof(msg.text), 0);

Ключевые функции:

- msgget() – открыть очередь
- msgsnd() – отправить сообщение
- msgrcv() – получить сообщение

Результат аналогичен: