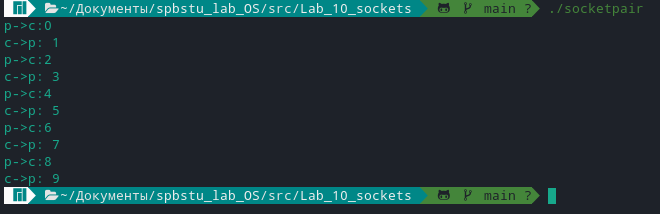
# **Лабораторная работа №10 «СОЗДАНИЕ СОЕДИНЕНИЙ НА СОКЕТАХ»**

**Цель работы**

Освоение набора системных вызовов для создания сокетных соединений различных типов для обмена данными по сети.

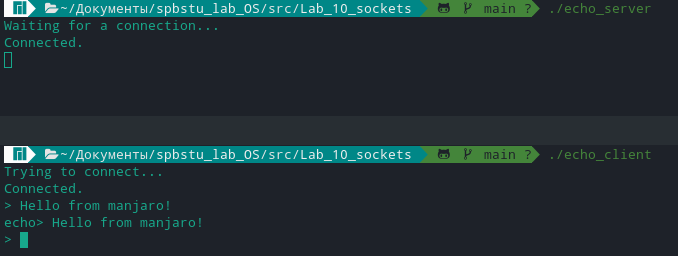
**Последовательность выполнения работы:**

1. Войдите в систему и скопируйте с разделяемого ресурса в свой HOME-каталог набор исходных файлов для десятого занятия.
2. Скомпилируйте и выполните программу socketpair.cpp, иллюстрирующую создание простейшего вида сокета и обмен данными двух родственных процессов. Проанализируйте вывод на консоль. Существует ли зависимость обмена от различных соотношений величин временных задержек (в вызовах sleep()) в процессе-родителе и в процессе-потомке?



Родитель и потомок передают друг другу поочерёдно числа от 0 до 9, передача происходит через сокет. Зависимость от задержек есть только в скорости вывода, но порядок чисел изменён не будет.

1. Скомпилируйте программы echo\_server.cpp и echo\_client.cpp , задавая им при компиляции разные имена (размещаем файлы в одном каталоге). Запустите программы сервера и клиента на разных терминалах. Введите символьную информацию в окне клиента и проанализируйте вывод. Какой разновидности принадлежат сокеты, используемые в данном примере клиент-серверного взаимодействия? С чем связано создание специального файла в текущем каталоге во время исполнения программ?

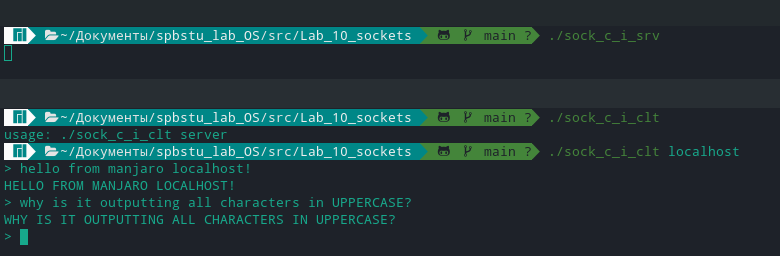


Видим, что результатом работы является вывод сообщения клиента клиенту только с добавлением “echo>” перед сообщением.

Тип сокета SOCK\_STREAM (двусторонние надежные последовательные потоки байтов) и адресное пространство AF\_UNIX (сокет рассчитан локальное соединение).

Создание специального файла связанно с созданием сокета в echo\_server.cpp. Сокет привязывается к UNIX адресу и после выполнения функции bind создается специальный файл по заданному пути.

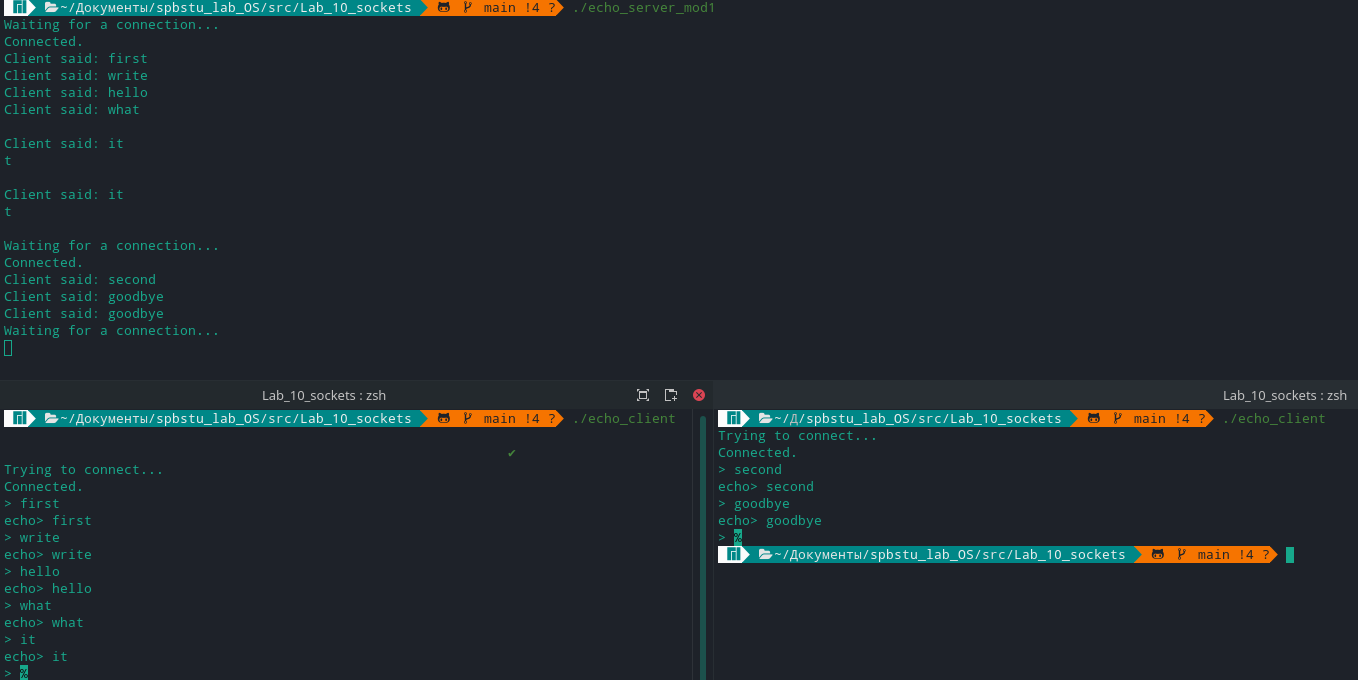
1. Скомпилируйте c разными именами программы sock\_c\_i\_srv.cpp и sock\_c\_i\_clt.cpp (в них используется общий include файл local\_c\_i.h). Запустите программы сервера и клиента на разных терминалах. При запуске клиента указывайте в качестве параметра командной строки имя хоста localhost. Введите символьную информацию в окне клиента и поясните вывод. Какой разновидности принадлежат сокеты, используемые в данном примере клиент-серверного взаимодействия?



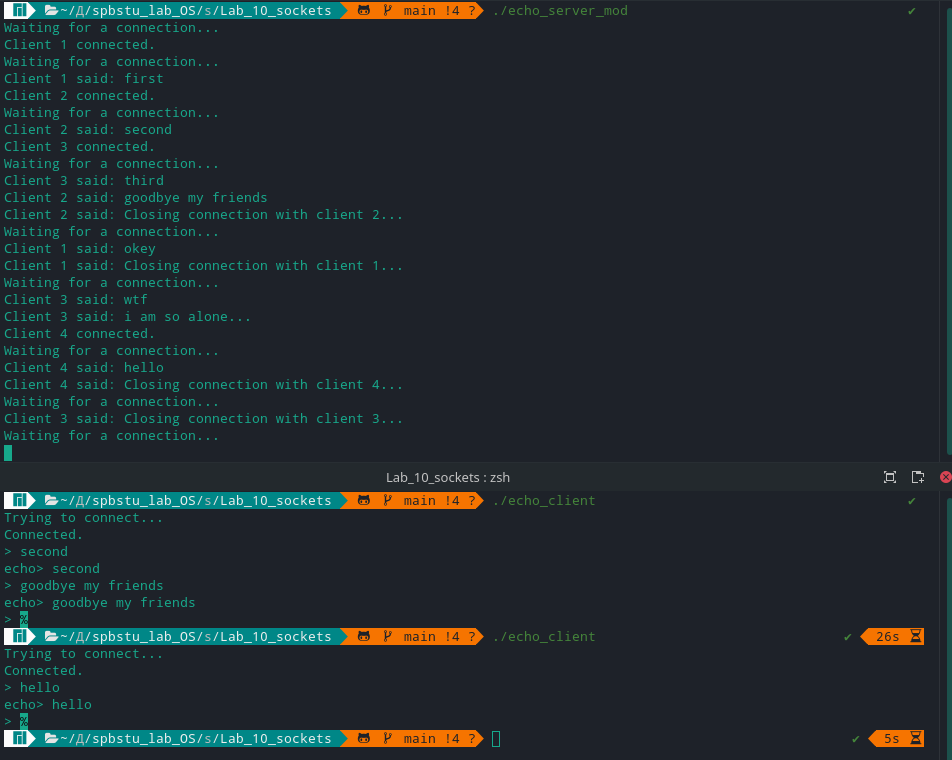
Тип SOCK\_STREAM (двусторонние надежные последовательные потоки байтов) и адресное пространство AF\_INET (то есть сокет рассчитан на соединение по сети).

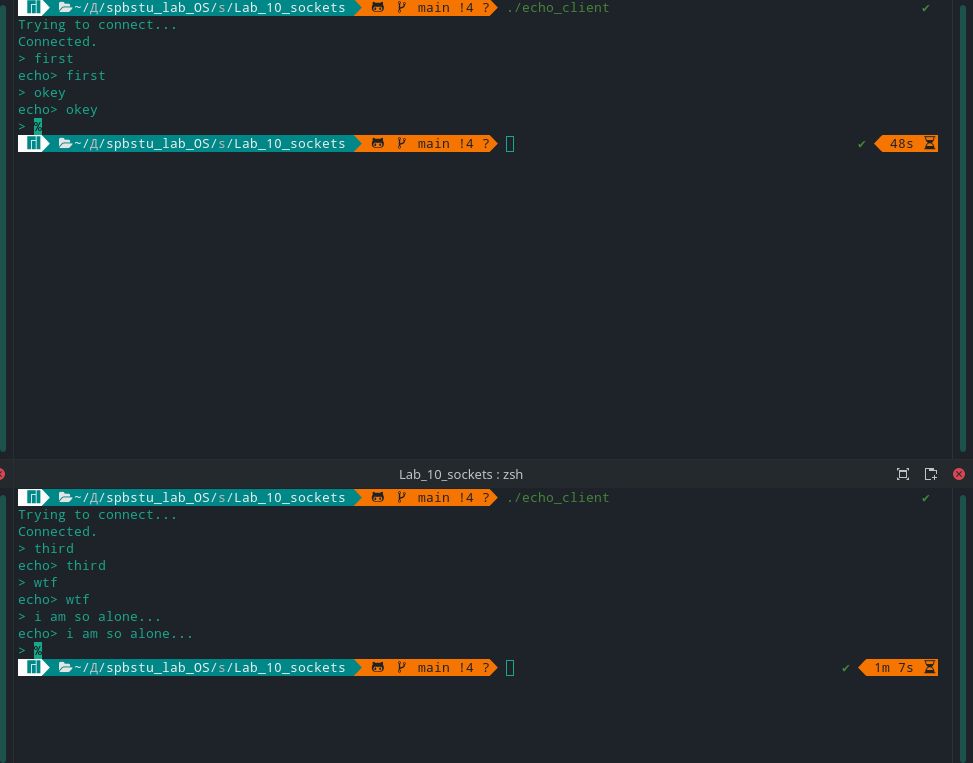
1. Модифицируйте программу echo\_server.cpp так, чтобы при ответе на запросы клиента что-либо выводилось в окне сервера. Испытайте работу эхосервера при одновременной работе с несколькими клиентами. Для вывода сообщения сервера просто попросим делать printf перед отправкой сообщения.

При одновременном подключении нескольких клиентов общается с сервером только первый. Остальные ждут. Как только первый отключается, подключается следующий.



Программа была доработана для работы с несколькими клиентами одновременно (происходит вызов fork и каждый клиент работает с новым дочерним процессом).





**Вывод**

В ходе лабораторной работы освоен набор системных вызовов для создания сокетных соединений различных типов для обмена данными по сети.